

On how metadata enables enriched file-based production workflows

Dieter Van Rijsselbergen
Ghent University – IBBT

Maarten Verwaest
VRT–medialab

Erik Mannens
Ghent University – IBBT

Rik Van de Walle
Ghent University – IBBT

As file-based production technology gains industry understanding and commercial products are becoming common-place, many broadcasting and production facilities are commencing re-engineering processes towards file-based production workflows.

Sufficient attention should however also be spent on the development and incorporation of standardized metadata in order to reach the full potential of such file-based production environments. In addition to its initial meaning, metadata and underlying data models can represent much more than just some meta-information about audiovisual media assets. In fact, properly modeled metadata can provide the structure that holds various media assets together and that guides creative people through production workflows and complex media production tasks. Metadata should hence become a first-class citizen in tomorrow's file-based production facilities

The aim of our paper is to show how standardized metadata standards and data models, complemented by custom metadata developments, can be employed practically in a file-based media production environment in order to construct a coherently integrated production platform. We discuss the types of metadata that are exchanged between different parts of the system, which enables the implementation of an entire production workflow and provides seamless integration between different components.

1. Introduction: Requirements for a file-based media production facility

While initially taking the media, entertainment and broadcasting industry by storm, file-based production technology is now becoming common place [1]. Many facilities have begun or are about to start significant transitions and re-engineering processes to convert legacy and proprietary tape-based systems to state-of-the-art media production solutions. The goal of such a transition is to make the implementation of complex media production workflows – that typically involve many different production components and actors – easier, more efficient, and in the end cheaper. The efficient implementation of such a transition becomes even more acute when the needs for future support for rapidly evolving new distribution media, such as the internet, is considered.

The multitude of processes involved in a typical media production workflow complicates the construction of an efficient file-based media production facility. Each of these processes likely uses different hardware or software and involves a great amount of workflow information that floats around between systems and users, in both formalized and ad hoc manifestations.

In order to accomplish the goal of successfully implementing a file-based production facility, we can distinguish two requirements that should be satisfied.

- 1) All components – both hardware and software – used by production processes that contribute to the media production workflow should be made easy to integrate into the overall file-based media production infrastructure.

- 2) The workflow information, represented by metadata, associated with the media production process should be a first-class citizen in a file-based facility and should be modeled and exchanged as extensively and accurately as possible.

The best way to integrate a large set of heterogeneous components is by abstracting the specifics of a component's interface into a façade interface. Expensive and exhaustive point-to-point integrations can be avoided by abstracting away the specifics of different implementations. This way, abstracted interfaces can provide services that hide underlying implementation details, including brands, proprietary exchange formats and expose more generic capabilities. Clearly, this moves us into the area of service oriented architectures (SOA) which provide an essential foundation for efficient and cost-effective integrations for the realization of the complex workflows associated with media production.

In this paper, we will not particularly focus on how SOA media facilities are constructed, which is likely to be covered in companion conference papers and has been discussed in the work of, among others, Footen and Faust [2]. However, we will focus our attention on the second requirement, the implementation of the metadata aspect of media production.

The use of information technology in the media enterprise for handling audiovisual essence presents a unique opportunity to re-engineer the way media is produced and conceived. Unlike in the past, both audiovisual essence and production metadata can now live within the same information technology space. A media production is no longer a collection of tapes with identification notes – metadata! – stuck to them, and should no longer be accompanied by heaps of informal production information written down in binders or scattered among production staff desks. The unification of separate essence and workflow information flows into a single storage and exchange medium allows us to arbitrarily interweave audiovisual material with its associated metadata.

This metadata and its underlying data models can be much more than some generic set of information that describes an individual media asset. Properly modeled metadata can provide the structure that holds various media assets together and that guides creative people and their tools through production workflows and complex media production tasks.

In fact, this practice has already been performed in various other product manufacturing industries where virtual models of products are first constructed by means of Computer Aided Design (CAD) tools and then assembled from components that are crafted by means of Computer Aided Manufacturing (CAM) [3]. While the media industry is certainly unique with respect to the creative aspects of its production methods, we should nevertheless thoroughly analyze the production process and attempt to model all information that describes the workflow and the product in all stages of completion.

When production and workflow metadata is collected and unified into a single data model and stored in a central location, this information is available for all applications and users that require it. This alleviates any redundant and tedious labor in workflow information shuffling, translation and duplication and avoids information loss from which earlier ad hoc paper-based workflows often suffer. And just as the services from Requirement 1 abstract sources of functionality, does a unified set of metadata abstract away different sources of production workflow information. Preferably, such a set of metadata uses as much standardized and open specifications as possible.

Additionally, by moving structured production metadata into the production system, the system itself can cooperate in handling, inferring and processing metadata and even use the virtual model to automate production processes. In particular, the system can handle a number of tasks such as timecode and filename management during acquisition.

The production metadata obtained through Requirement 2 forms a layer that can be placed over a bare service-oriented integrated facility from Requirement 1, in such a way that the metadata becomes part of the service contracts. Each piece of metadata is a part of the unified production information model. This allows services to easily obtain information about the entire production.

The aim of this paper is to demonstrate how metadata standards, complemented by custom metadata developments, can be employed practically in a file-based media production environment in order to construct a coherently integrated production platform. We illustrate our point using the case of the Scoop and Kameleon proof-of-concept drama production system, developed in the IBBT PISA research project by a collaboration of Flemish broadcaster VRT and academic research groups and industry partners¹. This system realizes a contemporary drama production workflow, but also implements a number of more advanced production processes and associated applications. The system provides basic script writing, file-based acquisition and editing functionality, but also adds 3-D Previsualization and Semantic Mastering applications for a scalable realization of production for a variety of delivery platforms. This system is an extended version of the one discussed in [4]. While that work focused mainly on the integration issues of a custom and specific production system with generic Media Asset Management Systems (MAMS) and a craft editing environment, we will focus on the metadata that is exchanged between the different system components. Our goal is to provide an idea of what kind of metadata, beyond classical examples as e.g., the identification fields of Dublin Core can be employed in a fully file-based and integrated production platform. We also wish to illustrate that various metadata specifications, both existing and custom developed ones, from both the industry and academia, can be seamlessly integrated to form a large virtual media production model that enables the realization of efficient drama production workflows.

2. The PISA Drama Production System

In order to provide some perspective, we will first discuss the functionality of the drama production system and will then lay out the different metadata schemes and documents that are exchanged in order to make the system operate. Figure 1 shows the different aspects of the drama production model that the PISA system provides, in terms of production processes. Each of the processes displayed is shortly discussed below to provide an idea of the kinds of production workflow information the PISA system will have to deal with.

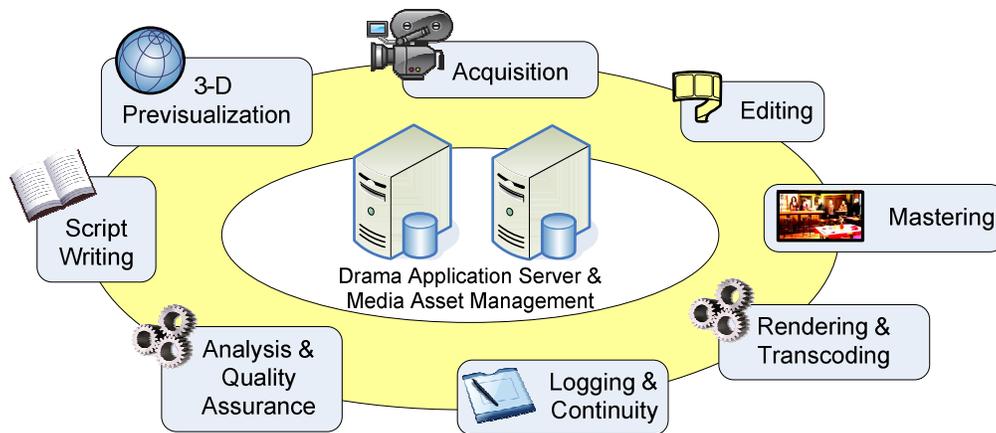


Figure 1: Processes involved in file-based drama production.

Script Writing and 3-D Pre-visualization

The drama production process commences with the creation of a number of programmes and episodes. Episodes are then divided into a number of scenes by the production and script writing staff, to serve as the basic unit of further production. The content of each scene can be defined using a web-based screenplay editor, or it can be set up using the Scoop 3-D Previsualization modeler. This latter application provides

¹ <http://www.ibbt.be/project/pisa> and http://www.vrtmedialab.be/index.php/english/project/project_p_pisa

creative staff with a computer-generated 3-D environment in which characters are placed, dialogue is written and virtual cameras are parameterized and animated. The director will use this 3-D environment to set up shots and actors and coordinate cast performances accordingly. Essentially, 3-D Previsualization provides the production crew with animated storyboards that help visualize how scenes will be shot. A screenshot of the application is shown in Figure 2. More information about the 3-D Previsualization application, which also features a multi-touch interface, can be found in [5].

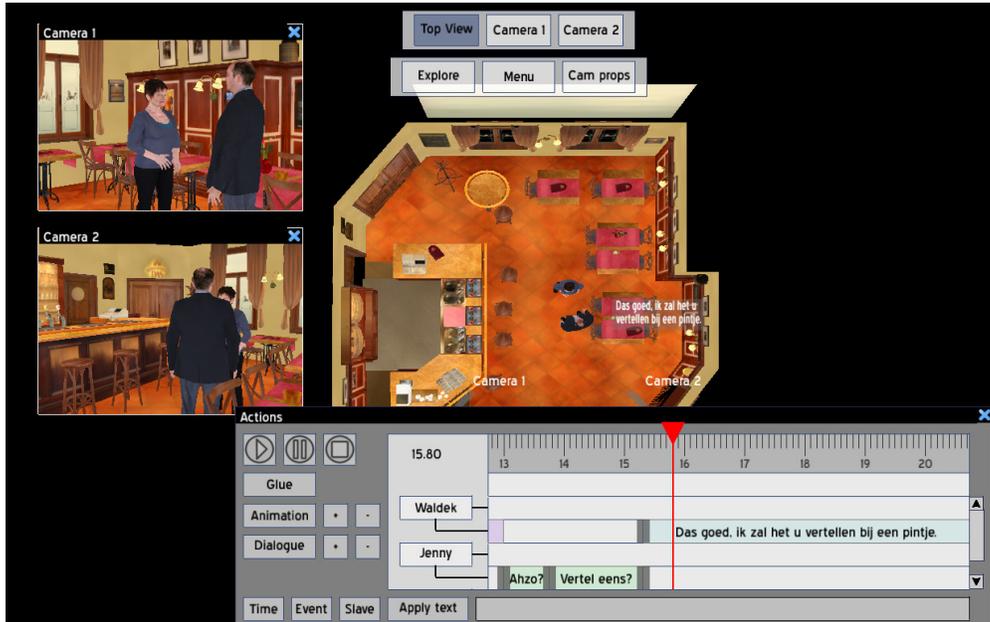


Figure 2: Screenshot of the 3-D Previsualization Editor.

Acquisition & Logging

The material acquisition process follows the completion of the regular shooting scripting or 3-D previsualization stages and is integrated entirely into the drama production system. The shot definitions planned previously by the director are reused here as metadata placeholders for acquired material.

A simple computer program controls the acquisition process and can be used to control off-the-shelf file-based cameras, decks or ingest servers. A screenshot of this application is shown in Figure 3a and Figure 3c. Upon acquisition, the symbolic camera and microphone definitions are associated with physical recording devices such that the necessary media tracking relationships can be put in place. Timecodes are read back from recording units and are automatically stored together with logging annotations that are made during or after recording. All communications between the acquisition devices and the drama production system occurs automatically without manual interference, for example for timecode annotations, resolving file names and ingesting.

All recorded takes can be annotated with log messages and approval indicators. Considering that the system can also operate in a multi-camera configuration, log messages can be associated per recorded camera view and/or for the performance as a whole. For such multi-camera setups, the PISA software also supports live-mixing on during recording (as depicted in Figure 3c) where the edit decisions are captured in order to be imported into a craft editor for further refinement.

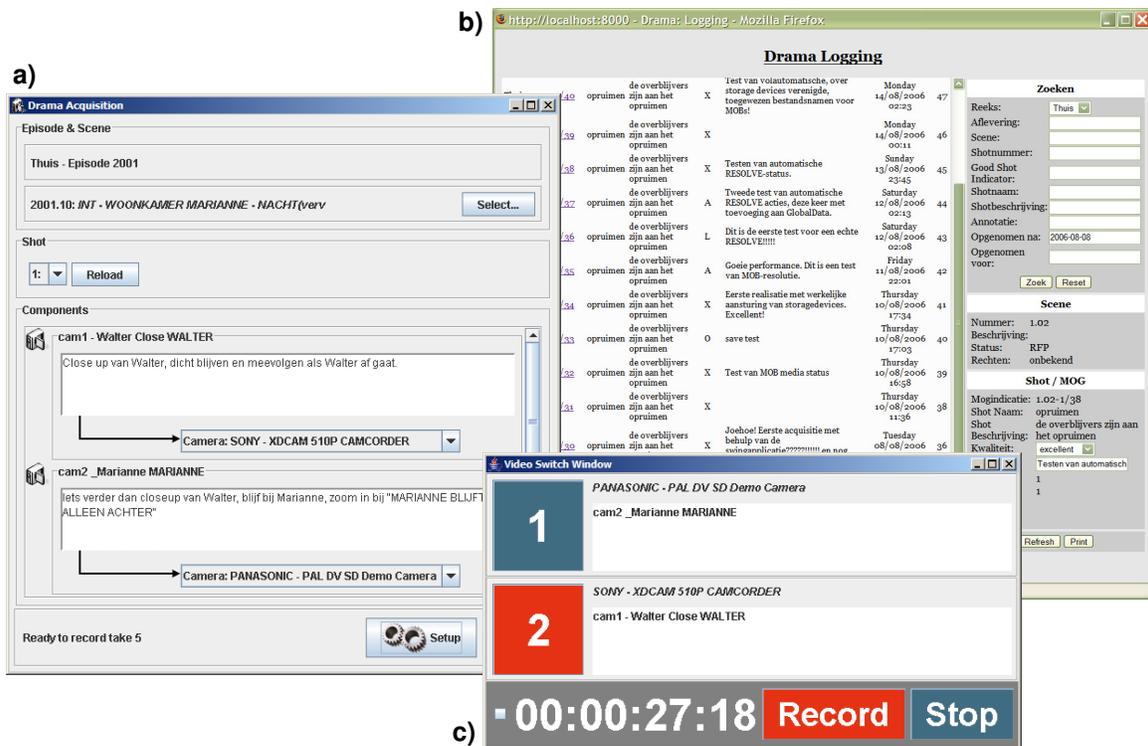


Figure 3: Screenshots of Acquisition and Logging.

Analysis and Quality Assurance

During the analysis phase of the production process any number of analysis techniques can be applied to media assets that have been ingested. Typical applications of analysis and feature extraction tools include shot-cut detection and representative keyframe selection. Another application is the detection of moving Regions of Interest that represent on-screen characters and props. The results of this analysis can be used to enable advanced applications and automated machine reasoning about the drama product being produced.

The analysis could be automatic or be performed by human operators, or a combination of both. Due to the limited running time of the PISA project, the required analysis that maps semantic story elements to regions on the screen, is currently done manually but will be automated in the future, using extensive intelligent detection algorithms. It is worth noting that we expect the accuracy and computation requirements of current state-of-the-art detection techniques to be aided significantly by the rich semantic context provided by the extensive drama metadata model. For example, the knowledge that a scene, according to the storyboard, is likely to feature only two characters located roughly at either side of the frame should ease the detection process of these two characters significantly.

As a part of the Quality Assurance process, all logging information that was performed during acquisition can be revised, and updated/extended afterwards using the drama application's web interface application, cf. Figure 3b. An early selection can be made of material that is likely to be used for editing, while discarding all redundant or useless takes.

Editing

Editing is a very important drama production process that deserved a significant amount of attention in the PISA drama production system. Stories and moods are shaped largely through careful editing. In order to provide appropriate editing support in our proof-of-concept system, we chose not to implement a non-linear editing application ourselves, but to integrate our system with an existing and full-featured craft editing environment. Compositions (some from live-editing at time of acquisition) can be exported to the craft editor and finished compositions can be imported again to become part of the unified metadata set that the production system manages. Thanks to the integration of our system with a generic Media Asset Management System, the endeavor of connecting with the craft editor was greatly simplified, as will be discussed later in Section 4.

Semantic Mastering

Viewers are gaining access to a multitude of platforms for the consumption of broadcasted audiovisual media; from movie theaters and high definition-enabled or standard definition television screens at home, to smart cellular phones on the road, and online using the web. These consumption devices vary widely in capabilities and specifications. Device display resolution, for example, can differ over an order of magnitude. Content producers and providers must attempt to provide this entire range of different devices with access to properly customized audiovisual products. This implies that content must be altered to meet the limitations of the audience's terminal and network. However, we cannot reasonably expect to provide users using a cell phone with an identical, but simply down-scaled version of source material originally acquired in high definition. Furthermore, considering the increasing percentage of viewers that has access to true high-definition presentation devices, creative content creators are likely to start focusing more on wide cinematic framing without being constrained to the limited real estate currently offered by common narrow television screens. This is why the PISA system introduces a Semantic Mastering process [6].

Similar to professional audio production, where sound operators prepare and optimize a mixed product for a given output medium, mastering is used to fine-tune an audiovisual drama production for release to a variety of consumption platforms. The drama crew decides how the video repurposing (cropping and scaling of the original imagery in order to fit on a smaller display device) should be performed by expressing mastering instructions in terms of semantic elements and using proper cinematographic terminology where possible. Example repurposing parameters would let Jenny "fill 100% of the frame" and "position her on screen left". Naturally, the results of the analysis processes are used extensively by Semantic Mastering: reframing can only be done correctly if semantic objects have been properly detected and described.

Creative staff uses a mastering client application, Kameleon (displayed in Figure 4), to extend existing scene description documents with mastering parameters.

Transcoding & Rendering

Throughout the drama production processes, a number of media transcoding and rendering processes take place. These processes operate automatically using specifications or media assets provided by other production processes. Transcoding or rewrapping is performed to match the requirement of certain components in the production process, for example, we exchange only standard definition-resolution material with our MAMS and craft editors, which is derived from high definition originally acquired footage. Rewrapping is done from MXF to other container formats for broader support in various PC applications and platforms.

Scenes that have undergone 3-D previsualization can also be turned into media assets by means of 3-D previsualization rendering. Similarly, we employ semantic adaptation to render the semantic mastering specifications into a number of different output formats in order to fit various consumption devices.

The transcoding and rendering processes participate fully in the metadata lifecycle and provide the drama production system with new media assets, as well as metadata that describe these assets.

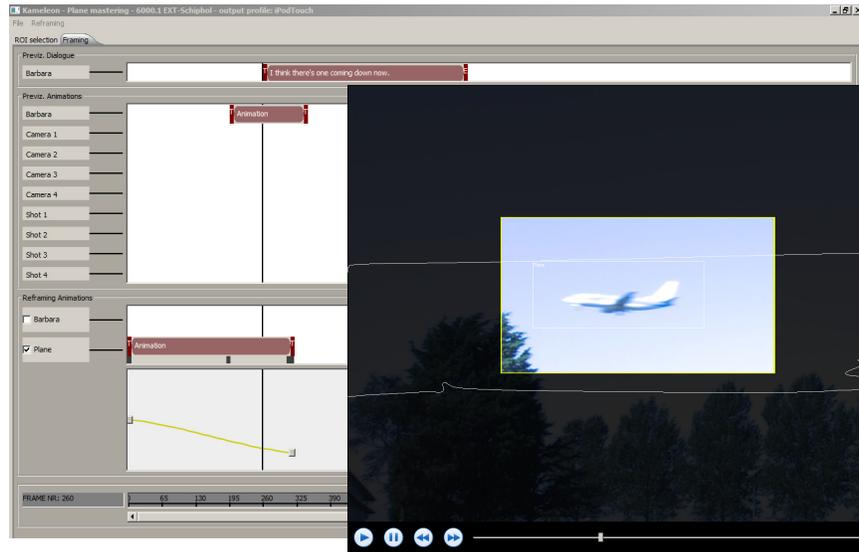


Figure 4: Screenshot of the Semantic Mastering application.

3. Architecture

The architecture of the PISA production system is depicted in Figure 5. In our proof-of-concept setup, all systems were connected using a standard network and communicated using the common TCP/IP protocol.

Audiovisual essence is stored in file wrappers and all essence transfers take place over the interconnecting network, using common protocols such as FTP as the lowest common denominator for most devices and more advanced and more flexible network protocols for more specific applications or component pairings (e.g., SMB for use in clients applications such that networked resources can be accessed directly without a requiring local copies). Thanks to the file-based nature of the acquisition process in the PISA system, we can make an abstraction of the actual acquisition devices used during production. In fact, any device can be used, as long as it provides network-accessible storage and wraps its recorded essence in a proper container (e.g., MXF or MP4).

A central Drama Production Application Server collects all production metadata in a database and runs application logic that controls and coordinates operations performed on this metadata.

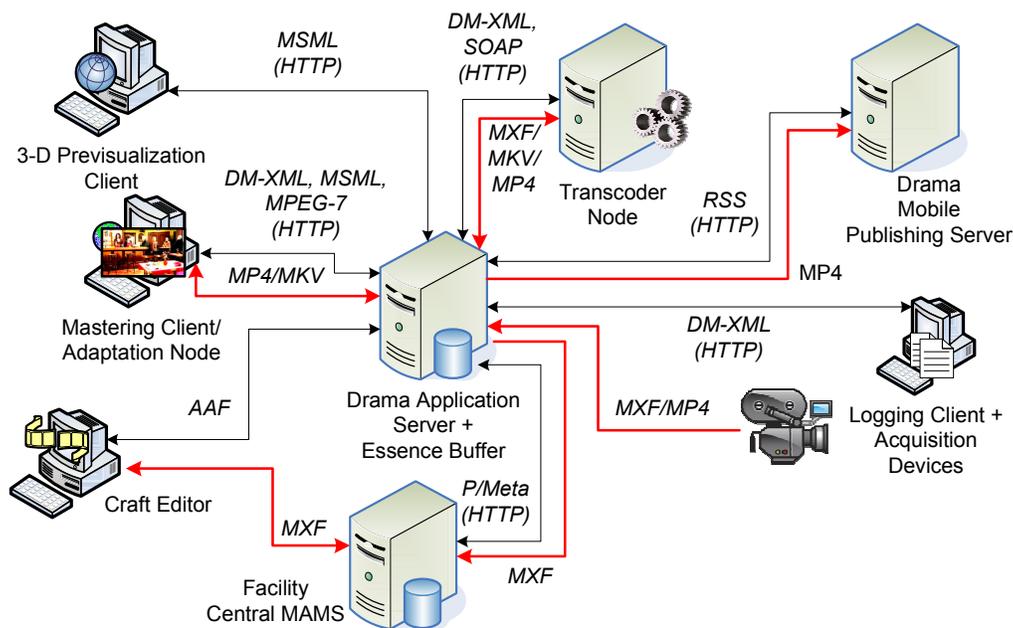


Figure 5: Architecture of the PISA Drama production system.

4. Metadata and workflow information

As was mentioned before, we consider all metadata in the production system to contribute to a single and extensive model of the media production (which includes both parts that are planned for production and parts that have actually been realized). In order to construct the entire model of production and workflow metadata, a number of existing specifications were combined with a custom data model, designed with a number of existing standards and best practices in mind.

Figure 6 paints the overall picture of how the production metadata ecosystem is constructed. The PISA drama data model acts as the binding fabric and defines the structure of the product and its assembly process using three layers of business objects. Intertwined with the PISA drama model is metadata from other specifications that were found well-suited to represent specific aspects of the drama workflow.

The PISA Drama Data Model

When starting the development of our proof of concept drama production system, we found that none of the existing metadata standards was equipped to sufficiently capture the specifics of the drama production process (cf., the initial Requirement 2). Borrowing from existing specification such as P/Meta [7] and the BBC Standard Media Exchange Framework (SMEF) [8], combined with modeling efforts performed by the VRT R&D department, we constructed a data model that can store generic media production elements, but is also able to express specific details about how drama products are assembled [9]. A similar modeling attempt could be repeated for other media formats, such as news, for which specifications such as NewsML [10] could serve as templates.

One central instance of the data model is stored for the drama system and all metadata and information exchanges are related to the objects in this data model store. An overview of the primary elements that

comprise the data model is depicted in Figure 7. The data model consists of three layers of functionality, each representing a specific phase of the production process of a single media asset.

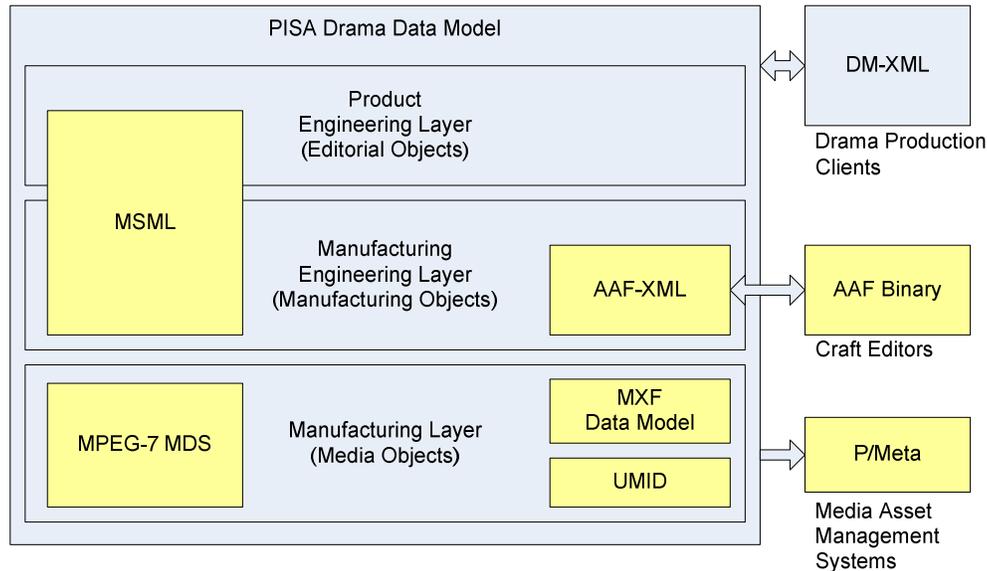


Figure 6: The PISA metadata ecosystem.

The Product Engineering processes determines product specifications while maintaining abstraction from the actual media asset manufacturing, essentially by defining the narrative aspects of the product. Two specific drama Product Engineering processes can be distinguished: synopsis writing and dialogue script writing. In both processes, producers, story writers and editors, researchers and the director collaborate on the definition of the narrative aspects of the product. A synopsis contains only a summary of a scene or episode, while the dialogue script contains integral character performance descriptions and dialogue. Product Engineering defines a product as a composition of logical units of creative or editorial work (e.g., episodes, sequences and scenes) or Editorial Objects. For drama, scenes are obviously the most prominent Editorial Objects, since they serve as the typical discrete unit of work through which drama is performed, shot and edited.

During Manufacturing Engineering, the Product Engineering specifications are elaborated into more exact manufacturing guidelines as to how actual media assets will be realized. In fact, this process creates blueprints for the individual components that the final product will be assembled of, and specifies how this assembly is to be made. One type of such blueprints are Shots (and their subcomponents, Camera and Microphone perspectives) through which directors can define their vision of how a scene should be realized into audiovisual essence. 3-D previsualization (cf. Section above) and Semantic Mastering are more elaborate Manufacturing Engineering process. Similarly, the craft editing process defines an Edit Decision List specification, the composition, which can be used to produce an assembled version of various different product components.

When the Manufacturing Engineering specifications have been finalized, the actual creation or Manufacturing process of the product, or components thereof, can be initiated. The result of Manufacturing processes are tangible audiovisual media assets. The acquisition of audiovisual material on the set, through filming or video capturing, is the most common Manufacturing process. The rendering of a composition into a ‘flattened’ media asset is also a Manufacturing process. Rendering and transcoding processes also operate in the Manufacturing tier. This includes previsualization rendering, which translates 3-D scene and animation descriptors into sequences of 2-D images.

The two latter processes illustrate that, given sufficiently accurate manufacturing engineering specifications, some manufacturing processes can be automated and that audiovisual essence depicting the scene can be rendered directly from them.

The drama system keeps track of each created audiovisual media asset through a Media Object. Every Media Object is automatically related to one or more Manufacturing Objects: the Media Object is said to be a realization, or rather concept we commonly refer to as the ‘take’, of the Manufacturing Object.

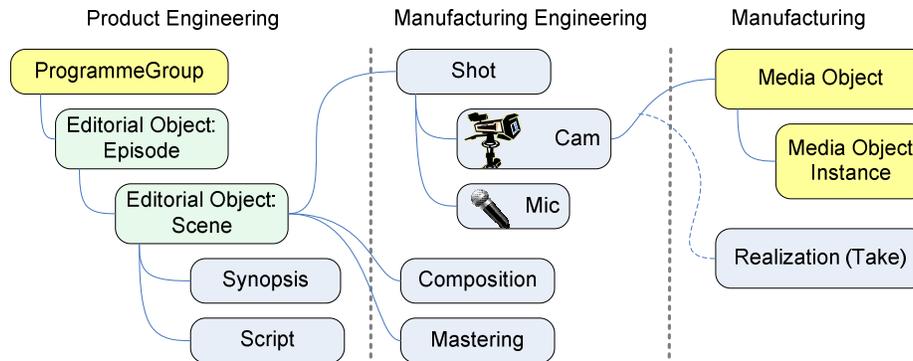


Figure 7: Overview of the PISA Drama Data Model.

The data model combines notions of all production processes mentioned in Section 2. It does, however, only represent a part of the production metadata set that is handled in the PISA drama production system. We have used other specifications where applicable to avoid reinventing existing solutions and to ease information exchange with external systems by using open and standard specifications. As such, for process-specific documents existing or custom specifications have typically been employed.

In the remainder of this section, we discuss the different metadata formats that we have integrated in our system. We start from the lower Manufacturing layer since this is an area that has been best explored, and work our way up to the more abstract Manufacturing and Product Engineering metadata.

In terms of the Manufacturing layer of the data model, where Media Objects live, we have built upon the extensive efforts by SMPTE, MPEG and the EBU in this area by reusing existing specifications and concepts as much as possible. Notable examples are the SMPTE UMID and MXF Data Model and MPEG-7 MDS descriptors, which are discussed next.

Manufacturing metadata: MXF Data Model and UMIDs

SMPTE Unique Material Identifiers (UMID) [11] are used as the primary means of audiovisual material identification in the PISA drama production system. UMIDs are extracted from recorded essence container files during the acquisition process as soon as they are ingested from network-attached file-based storage. UMIDs serve as transparent material references in metadata such that actual file locations are abstracted. This way, optimizations in terms of bandwidth by using material duplicates can be used throughout the production facility without invalidating existing references.

The PISA system describes the essence of Media Object instances using MXF data model terminology [12] [13]. We distinguish different Source Packages that consist of a number of tracks, each of which is described using a proper EssenceDescriptor (e.g., a CDCIEssenceDescriptor for an encoded video signal without temporal predictions). Additionally, Realization objects contain mapping information that links a certain take to a media object. This mapping is defined using the MXF source reference chain mechanism; Tracks containing logical Source Clips are used to indicate where exactly a specific Shot is active within the Media Object’s audiovisual stream.

Manufacturing metadata: MPEG-7 Multimedia Description Schemes (MDS)

In order for applications in the PISA drama production system to reason about the semantics of the content of media assets, and to automate certain production tasks, the semantic elements described in the scenario (e.g., characters and props) must be located in the audiovisual streams that represent a scene. A large number of initiatives have already been undertaken for the description of audiovisual content.

MPEG-7 is an extensive multimedia description standard that describes the characteristics of many aspects of audiovisual streams. Such aspects are typically low-level features of audio and video, including color, texture and shape information, sound category indexing, audio spectrum analysis and the definition of temporally and spatially moving regions. For a long time now, MPEG-7 has mostly remained an academic standard because few actual applications and solutions existed that employ automatically detected low-level features. However, it is gaining more and more traction as newer detection algorithms gain accuracy and interest in such detection algorithms grows for use in applications that are useable on the field and in real-world use scenarios.

We have used the Multimedia Description Schemes (MDS) part of MPEG-7 [14] to link moving regions in recorded images to semantic elements defined in a scene's narrative. Each relevant on-screen object in the video frames is rotoscoped by means of a moving rectangle that is defined through a limited number of key frame coordinate, between which intermediary rectangle locations are interpolated for smooth motions. The rectangle coordinates are determined by automatic analysis or manual annotation processes (cf., Section 2). Again, in terms of metadata we can make an abstraction of the exact type of analysis process was used and it suffices to describe the result of such this process.

Listing 1 shows an abbreviated MPEG-7 Region of Interest descriptor. A SpatioTemporalLocator describes the trajectories of the rectangular MovingRegion that represents a "Plane" object (specified by means of a StructuredAnnotation). A number of different key points are recorded at specific KeyTimePoints, the value of which is the frame when this key point is activated. An InterpolationFunctions element is used for each coordinate of the rectangle (i.e., x and y position, width and height) and one value is inserted per KeyTimePoint. The example includes only a single FigureTrajectory, but multiple could be added in case, for example, when the object goes out of frame and then dips back in. While our initial implementation describes semantic object using a single rectangle, MPEG-7 also supports more complex shapes which can be combined to represent more accurate descriptions of objects.

```

<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2004">
  <Description xsi:type="urn:ContentEntityType"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <MultimediaContent xsi:type="urn:VideoType">
      <Video>
        <SpatioTemporalDecomposition>
          <MovingRegion>
            <TextAnnotation>
              <StructuredAnnotation>
                <Who><Name>Plane</Name></Who>
              </StructuredAnnotation>
            </TextAnnotation>
          <SpatioTemporalLocator>
            <FigureTrajectory type="rectangle">
              <MediaTime>
                <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F">
                  208
                </MediaRelIncrTimePoint>
                <MediaIncrDuration mediaTimeUnit="PT1N25F">86</MediaIncrDuration>
              </MediaTime>
              <Vertex>
                <KeyTimePoint>
                  <MediaRelIncrTimePoint>0</MediaRelIncrTimePoint>
                  <MediaRelIncrTimePoint>20</MediaRelIncrTimePoint>
                </KeyTimePoint>
                <InterpolationFunctions> <!-- X coordinate -->
                  <KeyValue>0.8366388</KeyValue><KeyValue>0.62526095</KeyValue>
                </InterpolationFunctions>
                <InterpolationFunctions> <!-- Y coordinate -->
                  <KeyValue>0.35157698</KeyValue><KeyValue>0.35899815</KeyValue>
                </InterpolationFunctions>
                <InterpolationFunctions> <!-- Width -->
                  <KeyValue>0.2197286</KeyValue><KeyValue>0.19102296</KeyValue>
                </InterpolationFunctions>
                <InterpolationFunctions> <!-- Height -->
                  <KeyValue>0.14285715</KeyValue><KeyValue>0.14285715</KeyValue>
                </InterpolationFunctions>
              </Vertex>
            </FigureTrajectory>
          </SpatioTemporalLocator>
        </MovingRegion>
      </SpatioTemporalDecomposition>
    </Video>
  </MultimediaContent>
</Description>
</Mpeg7>

```

Listing 1: Example MPEG- 7 DMS descriptor.

Manufacturing Engineering metadata: Advanced Authoring Format

The metadata involved in the Manufacturing layer is relatively straightforward.

The specifications that make up the blueprints for the actual creation of media assets are more involved, and should be expressed using documents that represent the true craft of the production process in question. One such type of manufacturing blueprint is the contents of the Composition manufacturing engineering object, which is defined by the Advanced Authoring Format (AAF) Object Specification [15]. The structured AAF Object Specification realizes the requirements for an exchange format that expresses the result of non-linear video editing processes. AAF object structures can describe different tracks with source clips that are cut, trimmed and transitioned between in order to form an audiovisual presentation that tells exactly the cinematographic story that the directors and editor wish to convey to their audience. Essentially, an AAF object model instance represents an advanced type of Edit Decision List (EDL). Two processes in our system deal with AAF compositions. Obviously, we use AAF compositions for the integration with the craft editing environment, but already during acquisition an AAF composition is

'recorded' to describe any live mixing that occurs on the set when multiple cameras are used simultaneously.

All compositions stored by the drama production system are composed of AAF objects and stored in an XML rendition of this format. Considering that the MXF data model that we used for Media Object descriptions is actually a subset of the AAF object model, we were able to easily join the Manufacturing layer of the data model with AAF compositions. As mentioned, we use UMIDs as primary identification means for all material in the system. Hence, compositions also refer to material using UMIDs instead of absolute file locations. The XML format and use of symbolic material references allowed us to seamlessly integrated AAF compositions with the rest of the PISA workflow metadata model.

Product and Manufacturing Engineering metadata: Movie Script Markup Language (MSML)

Similar to how AAF is used as the manufacturing blueprint for editorial compositions, the drama production system needed a way to describe scene setups and script information.

Movie Script Markup Language (MSML) [16] is a screenplay and 3-D previsualization specification developed during the PISA project that is designed to model contemporary screenplays, enhanced with proper timing information and notions of character and camera animation. In developing MSML, we have tried to remedy the lack of a properly structured document format for the definition of screenplays. The semantics of the scenario can be expressed using elements like characters, and dialogue and action descriptions. At its heart, MSML is truly structured and does not rely on formatting styles to imply the meaning of portions of text.

Because of the strong correlation between the semantics of a scene and its manufacturing specifications (e.g., shots are often defined in terms of characters: "shoot a close-up of Joe"), we have used a single document format for both the scene's narrative and the scene's realization in terms of camera and microphone perspectives. Hence, MSML is used to describe scenes in terms of product engineering and manufacturing engineering. MSML documents can represent contemporary screenplays, but can also define a more elaborate 'shooting script' that details what shots a director wishes to realize and that tightly links to the original screenplay elements. The advantage of this approach is that MSML is used for many engineering processes, including script writing, 3-D previsualization, and even Semantic Mastering.

There exists an obvious (and intended) similarity between MSML and the drama data model. Many Manufacturing Engineering objects (incl., shots, cams, mics) are also found in MSML. One could say that we enforce a 'zero-divergence-doctrine' between the drama data model and MSML, similar to the AAF-MXF data model relationship. The aim of this approach is to have a flexible document format, in which various scene setups can be proofed and easily changed, that also serves as a direct template for the creation of data model objects. Once MSML documents have undergone a number of revisions and have been approved by directors and producers, its content is converted into actual data model objects. These data model objects are stored in the central database and are used to link with audiovisual material (cf., the Manufacturing Engineering section of Figure 7). Each data model object stores only a limited amount of specification information (e.g., previsualization animation information is not included) but does refer to its originating MSML document at all time.

```

<scene xmlns="http://drama.pisa.ibbt.be/MSML/xmlschema"
  location="Jeep" locationindicator="INT" time="Day">

  <character name="Joe" dramaID="http://wp3fiction/services/data/char/12"/>
  <character name="Andi" dramaID="http://wp3fiction/services/data/char/13"/>

  <instruction type="cameradirection">Favoring Joe.</instruction>

  <action>
    <characterref>Joe</characterref> drives recklessly.
    <characterref>Andi</characterref> sits next to him,
    an attractive girl in her mid-twenties.
  </action>

  <dialogue character="Andi" annotation="shouting">
    How much longer?
  </dialogue>
  <dialogue character="Joe">Couple o' hours. You okay?</dialogue>

  <action>She smiles wearily.</action>
  <dialogue character="Andi">I'll make it.</dialogue>

  <action>
    Suddenly, the motor <sound>sputters</sound>.
    They look at each other, concerned.
  </action>
  <instruction type="transition">cut to:</instruction>

  <cam uid="fb6fb155-3a86-42f0-9667-420b210550c3" id="cam1" name="Camera1" >
    <description>Close-up Joe</description>
    <direction pan="29.14" tilt="-1.4" roll="0"/>
    <fieldofview value="21"/>
    <focaldistance value="400"/>
  </cam>

  <animation subject="cam1">
    <keys>
      <key time="0"><position x="0" y="0" z="0"/></key>
      <key time="1"><position x="1" y="1" z="1"/></key>
    </keys>
  </animation>
</scene>

```

Listing 2: A Movie Script Markup Language document.

The MSML document shown in Listing 2 describes a scene that takes place in a *Jeep*. Two characters *Joe* and *Andi* participate in this scene. Actions and dialogue are represented by distinct element types. The screenplay also contains a number of instructions to indicate the directors (or screenwriters) desire for camera focus and transitions. A preparation of a camera Point of View (*cam1*) is also included, along with an animation element that animates the position of the camera in the 3-D previsualization environment. As the example shows, MSML merely expresses the structure of the elements in a screenplay and contains no formatting information. A typical screenplay template would, e.g., capitalize sound effect elements, and would place dialogue annotations between brackets. MSML documents can be paired with XML transformation techniques such as XSLT or STX in order to obtain a properly formatted screenplay rendition.

5. Metadata Exchange

In order to operate and to take full advantage of the model of the drama production that lives collected in the central PISA application server, this information needs to be communicated to other applications in the media production facility. In this section, we discuss the various ways in which this metadata is exchanged.

It is worth noting that, apart from binary AAF files, metadata is stored and exchanged in the form of XML documents. Such XML documents can be sufficiently structured using XML schemas in order to be machine-processable, and still remain human-readable, which, especially for debugging and troubleshooting in complex workflows is a very valuable feature. Since metadata is likely to remain relatively small in comparison to the essence it describes, there should generally be no issues in handling the processing workload involved in processing and transforming XML documents. Naturally, very specific or real-time applications could be better served with a leaner and likely binary format, e.g., binary AAF files. In those cases however, we should strive to use open specifications as much as possible.

DM-XML

The main method of metadata exchange consists of a serialized form of the PISA drama data model itself, which, for the sake of convenience, we refer to as DM-XML. This form is exchanged by production components that were developed specifically for the PISA project (incl., 3-D previsualization, acquisition, Kameleon). DM-XML, as the name suggest, is an XML format, and one that is formalized by means of a public XML schema document. As such, all parties that participate in metadata exchanges know exactly what format to read from another source and similarly know how to produce correct messages themselves. Where applicable, DM-XML documents contain fragments of the other metadata specifications we have listed previously. For example, the information concerning a specific video take will contain the MPEG-7 descriptor which describes the object visible in that take.

DM-XML documents are exchanged over the network using HTTP and through REST (Representational State Transfer) [17] services. This architectural style for information exchange uses the intrinsic nature of the HTTP protocol (which in itself already supports verbs for data creation, retrieval and updates) to manipulate information resources. Hence, while classical web services constructed with SOAP require additional message wrapping and translation steps, this style of data exchange is very easy to implement by clients and is also better suited for the exchange of large amounts of metadata or even audiovisual essence.

```
<mam:component baseURI="/services/data/component" id="137"
                xmlns:mam="mammal.drama.pisa.ibbt.be">
  <mam:name>Camera 4</mam:name>
  <mam:UUID>18b6b6f1-151b-4c40-9188-dcab06a82ecf</mam:UUID>
  <mam:created_for baseURI="/services/data/editorial_object_description" id="139">
    <mam:creation_time>2008-09-12T09:39:36.000+02:00</mam:creation_time>
    <mam:status code="APP"/>
    <mam:location_indicator code="EXT"/>
    <mam:location>Schiphol</mam:location>
    <mam:time_code>DAY</mam:time_code>
  </mam:created_for>
  <mam:description>Plane landing</mam:description>
  <mam:output_profile id="3">
    <mam:production baseURI="/services/data/production" id="1">
      <mam:producer>Winnie Enghien</mam:producer>
      <mam:director>George Terryn</mam:director>
    </mam:production>
    <mam:name>ORIGINAL_HD</mam:name>
    <mam:description>Full-frame High Definition</mam:description>
    <mam:resolution_width>1920</mam:resolution_width>
    <mam:resolution_height>1080</mam:resolution_height>
  </mam:output_profile>
  <mam:realisations>
    <mam:realisation baseURI="/services/data/realisation" id="186">
      <mam:indication>6000.1-Shot 4/1 Camera 4</mam:indication>
      <mam:good_shot_indicator code="X"/>
      <mam:annotation>Excellent tire smoke!</mam:annotation>
      <mam:start_time>2008-09-12T10:23:03.000+02:00</mam:start_time>
    </mam:realisation>
  </mam:realisations>
</mam:component>
```

Listing 3: Camera Component exchange information in DM-XML

In Listing 3, the DM-XML exchange information for a Camera component is shown. All attribute information (e.g., name and description) of a camera PoV is included, as well as referenced objects (e.g., the MSML document from which it was created, and all realizations of this camera PoV). Note that the component's UUID is preserved for each component that is created from an MSML document (the *cam* element in Listing 2 also has such a unique component identifier). Thanks to this identifier and the associated *created_for*-object, it is always possible to request the full details about a component, without necessarily storing all this metadata as separate entities in the database. Each referenced object element contains a base URI and ID, which can be used to fetch the full information set about that object using the REST HTTP services. This way, the entire object graph that represents the production metadata can easily be traversed.

P/Meta

The DM-XML metadata exchange format, although a direct materialization of the drama data model and hence also an open and accessible format, is currently not sufficiently generic to be exchangeable between a wide range of media production components from different vendors, especially those built for more generic media production or asset management tasks.

The exchange of production information between products from different vendors is more likely to occur by means of a standard that can be applied industry-wide. EBU P/Meta is such a standard aimed at the exchange of generic programme item and media object information. The PISA drama production systems exports part of the virtual drama model to P/Meta documents in order to communicate with external Media Asset Management Systems that provide – or that serve as a bridge toward – functionality not present in the PISA drama production system, including media play-out and archiving.

While by no means all production and workflow metadata can be expressed with P/Meta, it is sufficiently expressive in order to provide generic media asset management, play-out capabilities and to provide integration with craft editors.

The XML fragment printed in Listing 4 illustrates the P/Meta metadata that is exchanged about a Media Object (MOB) and one of its Media Object Instances (MOI). The MOB is identified using a global drama production system URI, while the identifier of the MOI is the UMID of the MXF file that is being represented. In order to allow transfers of the essence container, the UMID has also been translated into an FTP location where the Media Asset Management System can fetch the MOI's material. A P/Meta MOB also contains annotations about its essence. While none of the drama data model-specific objects are directly represented in P/Meta, we have used the *MOB_SCRIPT_DESCRIPTION* and *MOB_CAPTURED_DESCRIPTION* to transfer realization and camera PoV component information. This information is ingested into the MAMS, and given a proper integration, is also transferred to the craft editor environment where material can be viewed immediately with this continuity and logging information. Similarly, generic P/Meta item objects are used to represent episodes and scenes. Such items can also include a limited amount of production metadata (e.g., a list of characters in a scene Item), but they are employed mainly as a means of structuring imported media assets.

```

<MOB>
  <MOB_IDENTIFICATION>
    <IDENTIFIER>
      <IDENTIFIER_DETAIL>
        <IDENTIFIER_TYPE_CODE>V612</IDENTIFIER_TYPE_CODE>
        <IDENTIFIER_NUMBER>
          http://wp3fiction.pisa-poc.be/services/data/mob/720
        </IDENTIFIER_NUMBER>
      </IDENTIFIER_DETAIL>
    </IDENTIFIER>
    <MOB_TYPE_CODE>V217</MOB_TYPE_CODE>
    <MEDIA_OBJECT_TITLE_HISTORY>
      <ORIGINAL_MOB_TITLE>
        <MOB_TITLE>2001.03-2/1 (1: CU Renzo, mic1)</MOB_TITLE>
      </ORIGINAL_MOB_TITLE>
    </MEDIA_OBJECT_TITLE_HISTORY>
  </MOB_IDENTIFICATION>
  <MOB_ITEMISED_DESCRIPTION>
    <FRAGMENT_DETAIL>
      <MOB_SCRIPT_DESCRIPTION>
        2: halfway: zoom in
      </MOB_SCRIPT_DESCRIPTION>
      <MOB_CAPTURED_DESCRIPTION>
        2: First take annotation for this scene,
      </MOB_CAPTURED_DESCRIPTION>
    </FRAGMENT_DETAIL>
  </MOB_ITEMISED_DESCRIPTION>
</MOB>
<MATERIAL_EXCHANGE_INSTANCE>
  <MATERIAL_EXCHANGE_DETAIL>
    <EXCHANGE_TECHNICAL_DETAILS>
      <FILE_INSTANCE>
        ftp://wp3fiction.fipa-poc.be/buffer/3243_24_C0005.MXF
      </FILE_INSTANCE>
      <MOB_DETAIL>
        <IDENTIFIER>
          <IDENTIFIER_DETAIL>
            <IDENTIFIER_TYPE_CODE>V215</IDENTIFIER_TYPE_CODE>
            <IDENTIFIER_NUMBER>
              060a2b340101010501010d43130000001e791d00744005800800460202210315
            </IDENTIFIER_NUMBER>
          </IDENTIFIER_DETAIL>
        </IDENTIFIER>
        <MOB_TYPE_CODE>V217</MOB_TYPE_CODE>
      </MOB_DETAIL>
    </EXCHANGE_TECHNICAL_DETAILS>
  </MATERIAL_EXCHANGE_DETAIL>
</MATERIAL_EXCHANGE_INSTANCE>

```

Listing 4: Media Object exchange information in P/Meta

Binary AAF

Since AAF-derived documents are already stored within the drama production system, it is trivial to export and re-import these in a binary format for exchange with full-featured craft editor solutions. Again, a REST service interface was developed as an abstraction façade for a craft editor solution such that binary AAF files can be easily transmitted and received.

Having a working MAMS integration through which essence can be exchanged, and a MAMS that itself already provides craft editor integration, allowed us to avoid having to construct a similar essence exchange channel with the craft editor environment. We feel that this is an interesting demonstration of how a SOA can ease integrations. By concatenating two individual integration efforts, one between the MAMS and the craft editor, and the other between the MAMS and the drama production system, we were able to avoid building a third point-to-point integration between the drama production system and the craft editor system

(cf. Figure 5). Now, only the composition AAF file needs to be exchanged, from which the required media assets are distilled and essence file transfers between the MAMS and craft editor are performed without mediation from the drama production system. Thanks to the use of UMIDs in our AAF compositions, the drama production system remains agnostic to whatever the location of the media assets within the MAMS and craft editor may be; the façade service provides the proper translations when importing or exporting AAF files.

The current system implementation only supports AAF files with simple transitions, cuts and multi-camera selective switching between source clips. At the moment, more elaborate effects are insufficiently standardized to be universally usable. Hopefully, this will change in the near future as the work on the AAF Effects Protocol is completed.

6. Observations & Conclusions

In this paper, we have demonstrated by example that extensive metadata modeling efforts and its use throughout the production process to represent workflow information is invaluable for the implementation of efficient and enriched workflows in file-based production facilities. All workflow components, which include hardware, software and the production crew, work together on a single goal: one product that is defined by a single set of numbers. Different components contribute complementary parts of the product's workflow information, but always in such a way that all this information is tightly related and no disparate information sources exist to describe a single conceptual object.

For example, the MSML format allows us to extend narrative screenplays into 3-D previsualized environments, which in turn contains information that we can fully reuse throughout the rest of the production process. As such, strong bindings can be created between the descriptive metadata elements and the audiovisual essence we aim to produce in the end. Additionally, low-level feature analysis can help close the descriptive gap between media assets and semantic drama production elements, in order to help realize computer-assisted production processes such as semantic mastering.

At the same time, an expressive workflow metadata model cannot function properly without a number of service abstractions that provide translations from this clean and ideal model to the gritty and potentially complex details of whatever product is hidden behind the public interface. By defining a generic interface for a number of components (e.g., MAMS and craft editing solutions) that consists of a number of service methods and information messages (i.e., renditions of part of the production workflow metadata), our system, to some extent, also implements a service oriented architecture.

We did however, deviate somewhat from the more commonly used SOA construction scheme. As mentioned, we have used REST services as the main means for communication. In our case, this was perfectly fit for a flexible way of fetching structured information by means of resource locators (URLs) and using simple HTTP verbs. However, we do use common SOAP-based web services for a Remote Procedure Call-based style of communication in which we invoke remote operations, e.g., the initiation of transcoding or rendering operations. At the time, there was no explicit need for messaging middleware (also known as an enterprise service of messaging bus, ESB) or an explicit layer of business process management (BPM) functionality. The central drama production application server provides all production information and coordinates actions by implementing most business logic itself. In order to support a more elaborate and scalable production facility (this implementation is only a proof-of-concept setup) future implementation could be extended to use an ESB and more flexible BPM modules.

We were able to use a number of standardized formats to aid in the specification of various production process documents and the integration of different components. Unfortunately, the expressivity of these formats concerning the intricacies of the drama production process is limited. For example, only a small set of information from the drama production data model can be exchanged in a lossless manner using P/Meta. This prompted the custom development of our drama production data model and MSML specification. In order to preserve the rich metadata fabric constructed during production across various heterogeneous services in the production facility, a lot of work still remains to be done in terms of standardization of media production metadata models, and with it, media production service interfaces. This will reduce the

amount of custom modeling efforts and will thereby significantly improve metadata and service interoperability.

The work of the AMWA in this respect is very interesting and promising, considering the existing AAF format and the ongoing specification efforts concerning its Application Specifications and Media Services Architectures. Similar to how NewsML documents can already elaborately describe news items and associated essence, open and standardized specifications can hopefully be defined for creative and complex production processes such as drama production and multi-platform product repurposing. In this respect, we hope that our metadata developments can serve as inspiration for such standardization endeavors or for other experiments in the implementation of enriched media production workflows.

References

- [1] Al Kovalick, *Video Systems in an IT Environment*. Oxford: Focal Press, 2006.
- [2] John Footen and Joey Faust, *The Service-Oriented Media Enterprise*. Oxford: Focal Press, 2008.
- [3] D. Koshal, Ed., *Manufacturing Engineer's Reference Book*. Oxford: Butterworth-Heinemann Ltd, 1993.
- [4] Matthias De Geyter, Nico Oorts, and Luk Overmeire, "Integration Demands on MAM Systems: A Proof of Concept Solution," *SMPTE Motion Imaging Journal*, pp. 38-46, November 2008.
- [5] Maarten Cardinaels et al., "A Multi-touch 3D Set Modeler for Drama Production," in *Proceedings of the IBC Conference*, 2008, pp. 330-335.
- [6] Dieter Van Rijsselbergen, Barbara Van De Keer, Maarten Verwaest, Erik Mannens, and Rik Van de Walle, "Enabling Universal Media Experiences Through Semantic Adaptation in the Creative Drama Production Workflow," in *Proceedings of the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, 2009.
- [7] European Broadcasting Union (EBU). (2005, June) P/Meta Metadata Exchange Scheme v1.1. Technical Report 3295. [Online]. <http://www.ebu.ch/en/technical/metadata/specifications/>
- [8] British Broadcasting Corporation (BBC). (2005) Standard Media Exchange Framework (SMEF) Data Model, v1.10. [Online]. <http://www.bbc.co.uk/guidelines/smef/>
- [9] Dieter Van Rijsselbergen, Maarten Verwaest, Barbara Van De Keer, and Rik Van de Walle, "Introducing the Data Model for a Centralized Drama Production System," in *Proceedings of the IEEE International Conference on Multimedia & Expo 2007*, 2007, p. 615–618.
- [10] International Press Telecommunications Council (IPTC). (January, 2008) NewsML-G2 specification, v2.0 rev. 2. [Online]. <http://www.iptc.org/>
- [11] Society of Motion Picture Television Engineers (SMPTE), *SMPTE Standard for Television - The Unique Material Identifier (UMID)*, SMPTE 330M-2004., 2004.
- [12] Society of Motion Picture Television Engineers (SMPTE), *SMPTE Standard for Television — Material Exchange Format (MXF) — File Format Specification*. SMPTE 377M-2004., 2004.
- [13] Society of Motion Picture Television Engineers (SMPTE), *SMPTE Engineering Guideline for Television — Material Exchange Format (MXF)*, EG 41-2004., 2004.
- [14] B. S. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG-7: Multimedia Content Description Interface*.: John Wiley & Sons Ltd, 2002.
- [15] Advanced Media Workflow Association (AMWA). (2005, April) Advanced Authoring Format Object Specification v1.1. [Online]. <http://www.amwa.tv>
- [16] Dieter Van Rijsselbergen, Barbara Van De Keer, Maarten Verwaest, Erik Mannens, and Rik Van de Walle, "Movie Script Markup Language," in *Proceedings of ACM Document Engineering 2009*, 2009.
- [17] Roy T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Irvine, CA., Ph.D. Thesis 2000.