Archival Structures, Workflows and Distributed Systems

Jürgen Enge; Zurich University of the Arts; Zurich, Switzerland
University of Arts and Design Karlsruhe; Karlsruhe; Germany

Abstract

This contribution focuses on core structures and performing aspects of digital media archives. It takes up a practical point of view and focuses on open source solutions. By means of recently developed database systems four subjects will be faced: the presentation of different metadata models and their practical implementation, ingesting techniques for metadata and media files (workflow), storage systems and last but not least the appearance of so called meta archives. They give considerations to the consolidation of a distributed networked search among heterogeneous metadata systems.

Introduction

One of the main challenges during the preparation phase of a digital archive is the selection and design of an appropriate metadata model. This process has heavy impact on the success of the digital archive. The metadata model is a projection of the real world object or its descriptive data in digital format. This mapping decides the complexity of data insertion and should pragmatic and not complex.

The following text will presents four different metadata schemes. Each model focuses on core elements which induce different structures and entities. The IFLA model, the IDEAMA archive and DILPS system present varying *work centered* schemes. The IMEB Archive on the other hand has a composer centric structure.

The next topic to be discussed are different ways of ingest of digital media objects. The ingest are the procedures, which implement the workflow of digital media object transferred into the archive system. This workflow is related to both, the source content types (audio/video/image/text/..) and the future use of the media objects (library use, internet use).

The next paragraph gives some ideas about sustainable storage solutions and focuses on balance of costs and quality of storage systems.

Finally, a short survey on meta archives describes recent projects on distributed or centralistic systems.

State of the art

A well-known metadata model has been invented by the 'Dublin Core Metadata Initiative' (DCMI) [1]. The definition of the DCMI should be used as a basis for the metadata model design due to the fact that many archive systems use models related to DCMI as core metadata set for data exchange. Based on the fact that DCMI metadata terms are very limited, the model in most cases needs to be expanded to fulfill the needs of the archive structure.

The **MA**chine-**R**eadable **C**ataloging (MARC) record is a digital standard format which is used for storing metadata in libraries [2]. Due to the fact that it is widely (international) used in libraries, the attributes used in MARC records should be recognized prior to developing the metadata scheme.

'The International Federation of Library Associations and Institutions' (IFLA) is another resource to find hints and solutions for metadata models [3]. Especially the report 'Functional Requirements for Bibliographic Records' is very helpful for building up new metadata models [4]. A list of metadata resources can be found at the metadata resource list [5].

'The Preservation Metadata Maintenance Activity' (PREMIS) project gives a very good impression on the complexity of an archive system with the 'PREMIS Data Dictionary for Preservation Metadata' [6]. Besides the metadata descriptions, there are guidelines for creation, management and use with an orientation towards workflows.

Metadata Structure Examples

When developing a metadata scheme, several levels of data have to be defined. The attributes describing the objects are very important, but the structure within the scheme has to be defined very carefully. The mapping between the 'real' archival objects and the metadata scheme has to be as complete as possible with regard to the complexity of the whole model.

One model which meet these concerns is the IFLA model. The proposed model consists of four core levels in a hierarchical relationship. Work, expression, manifestation and item are the core entities of the IFLA model.

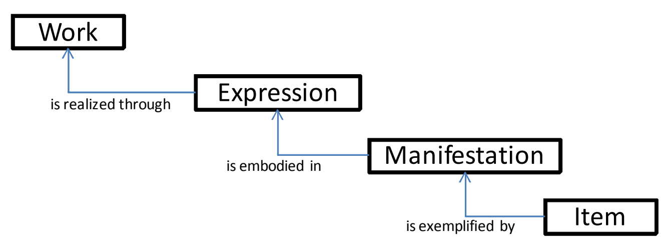

Figure 1. *Main entities and primary relationships of IFLA model*

The IFLA model is a vivid example for a work centric metadata scheme. Other collections may raise different requirements, for example person centric or event centric. This does not mean, that it is impossible to model such a scheme based on the IFLA model, but it could become inefficient.

In conjunction with references to other entities like persons, events, places etc., this model can become very complex and hard to implement in an archive system.

As a result, the metadata schema depends on the core element of the archive. The next examples show work, artist and item centric metadata schemes.

Work centric archive: IDEAMA

The IDEAMA Audio Archive is a work centric archive. It consists of several audio pieces partitioned by sets and tracks. Every set is related to one or many composers with attached multilingual biographies (Figure 2). The tracks are referencing audio files of different formats and qualities.

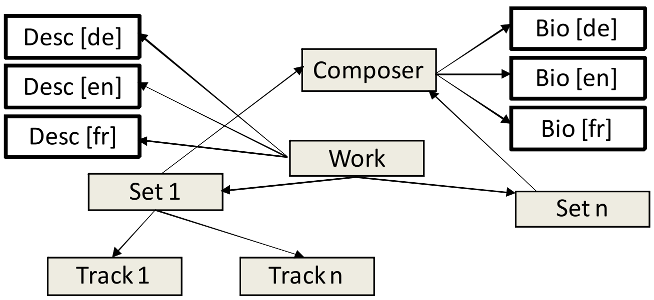

Figure 2. *Structure of IDEAMA Metadata Model*

By reasons of the fact, that the IDEAMA Audio Collection is closed, the metadatabase is implemented according to the structure of the original archive [7].

Composer Centric Archive: IMEB

The 'Institut International de Musique Electroacoustique de Bourges' (IMEB) hosts a festival on contemporary electro acoustic music. What makes this archive interesting for us is the composer centered structure. This means, works are related to composers and can consist of different tracks.

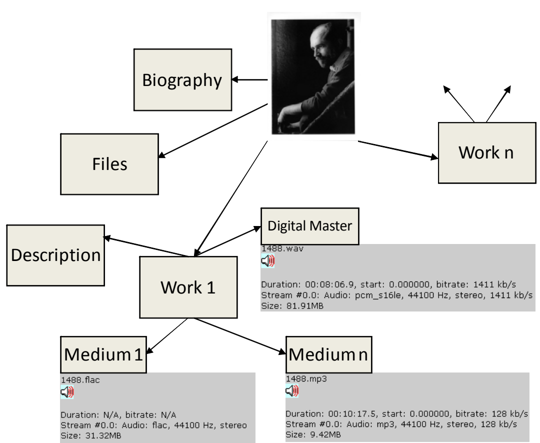

Figure 3. *Composer centric metadata schema*

Within this scheme, the *files* represent a non predictable set of additional materials. They are port of the composer transmittal. The basic metadata set of this archive derives from a line oriented database with many redundancies, which are removed when uploading into the archive system [8].

Item Centric: DILPS

The Distributed Image Library Processing System (DILPS) was originally intended as a plain slide archive system [9]. That is why, there is a focus on fast ingest and large scale. Due to the fact, that DILPS is not focused on a specific collection, there is need for more flexibility.

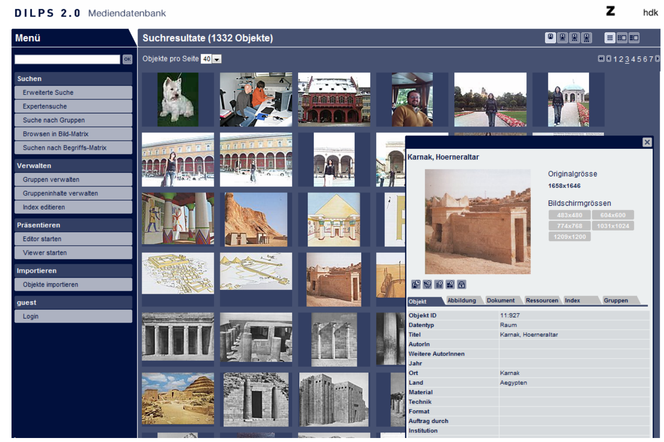

Figure 4: *Screenshot DILPS*

This means, that the metadata model of DILPS has some borrowing from the IFLA papers. The main structure consists of items and resources, whereas the *item* relates to the *work* and the *resource* relates to the so called *manifestation*. The type of the *item* chooses the appearance and metadata set belonging to it.

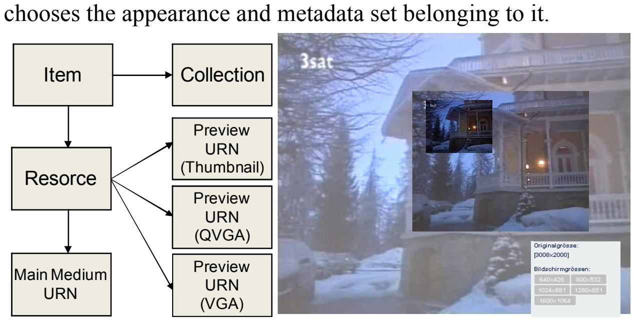

Figure 5: *DILPS metadata model with preview images*

Additional structuring entities are collections and URNs (Uniform Resource Names). Every item belongs to one collection, which is the core classification of the objects within the database. The item has a core metadata set to allow easy comparison and a dynamic one to reflect the different types of items. The item can have references to several resources which are the manifestations of the item. The metadata of the resources is primarily technical metadata and is generated by ingest process. Resources must have a reference which points to the so called master (digital or physical) object. Every object referenced should be represented by several preview images, which are used to show their content in different sizes within the user interface.

Ingest

Ingest of objects must be divided into ingest of metadata and the digital representation of the object. Ingest means copying the media object to a reliable storage and reference a metadata set with the digital asset. Besides this, the system tries to achieve as much metadata from the asset as possible. This *technical* metadata contains object size, dimensions (width, height, length) and many other attributes. Sometimes, the digital object itself contains metadata like the EXIF or ITPC/XMP metadata within digital images [10] [11]. In this case, the archive system has to distinguish between technical and semantic metadata.

In most cases the master asset cannot be displayed directly. There is need for several derived media objects, which can be used as thumbnail, as a displayable version of the master or just an excerpt.

Normally, archives are dealing with images, audio, video and text as digital master objects. Specialized archives deal with CAD drawings, animations and other file formats. They shall be handled in a specific way. Besides the format, the size of the media objects and the number of objects per ingest process influences the ingest method heavily.

Ingesting a limited number of small objects (<50MB) can be done as synchronous ingest. This means, that the user starts the ingest and waits in front of the screen until the ingest is finished. This is normally the case for image or audio data.

When ingesting large amounts of data (large number of objects or large object sizes), asynchronous ingesting should be used. These ingest queues may be processed sequential or parallel. The system has to ensure, that ingest is an atomic operation. This means, that there must not exist the possibility of a partly ingest (i.e. metadata set is in database, but object is not copied or transcoded).

Ingest Example: Single Image

The first step of the ingest process is to get information from the media object to determine how to handle it. The following screen shows exemplary commands which could be used by the ingest process.

```
# file -i iptc.jpg
iptc.jpg: image/jpeg
# identify iptc.jpg
iptc.jpg JPEG 800x600 800x600+0+0 DirectClass 8-bit
27.2363kb
# exiv2 -p i iptc.jpg
Iptc.Envelope.ModelVersion      Short        1  4
Iptc.Envelope.CharacterSet      Undefined    3  27 37 71
Iptc.Application2.RecordVersion Short        1  4
Iptc.Application2.ObjectName    String      17  Testing123
Iptc.Application2.Headline      String      24  Testimage
Iptc.Application2.Copyright     String      16  CC
Iptc.Application2.0x00e1        String      14  Foto
```

The first commands (‘file’) gives the mime-type of the media object (image/jpeg) which is used to decide how to continue. For objects with the mime-type image/* the program ‘identify’ is used to get further information (JPEG, 800x600px) [12]. There are several ways to extract IPTC metadata. In this example the software library ‘exiv2’ [13] is used.

After determining technical metadata, the system has to decide which derived objects has to be generated. In the case of images, a fixed list of target sizes, starting with the thumbnail size up to the size of the source image, will be generated. All derived objects are inserted into the archive system and referenced by potential metadata.

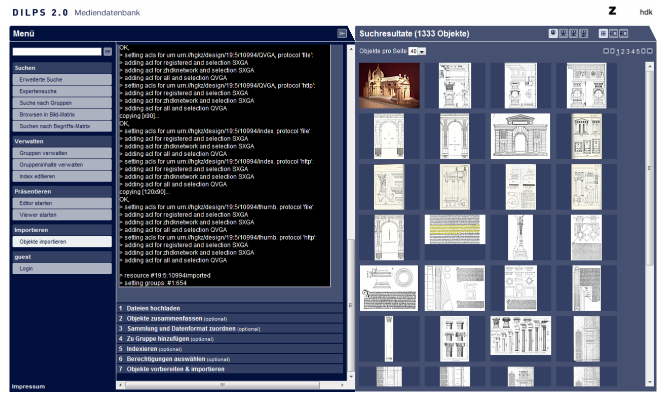

Figure 6: *Live Ingest of image data*

Ingest Example: Video Library

Ingest of video data is much more complicated than image ingest. Video data contains larger amounts of data than image data and the automatic generation of derivates can have many different aspects. Normally, the system has to generate several streaming media formats (high resolution, low resolution) and video shots which can be used as thumbnail or within a slide show. In this example, the Quicktime/Darwin streaming server is used to play the videos. Additionally, several might be published via video podcasts. This means that three video formats (high quality, low quality and IPod) have to be produced. The following table shows statistics from a production archive system.

Exemplary effort for transcoding 1h video

Codec	Size/Quality	Duration
Apple H.264	4CIF/multi pass	1:56h
Apple H.264	CIF/multi pass	0:47h
Apple IPod		0:59h
PNG Video shots every 10sec		0:08h
	Total time	**3:50h**

The machine used was an eight core MacPro. The Quicktime Pro based transcoding software uses only two of the eight CPU cores. This means, that people working with these desktop computers did not recognize the video encoding process which was running in background and therefore, approx. 30 machines of the universities classrooms became part of the transcoding cluster, which allows a transcoding throughput of more than 200h of video per day. The encoding is slowed down by people rebooting there desktop machines and as a result of the fact, that the download of the video source has to be decelerated to avoid fast hard disk write operations which would slow down the machine too much. The master video and the transcoded video streams and video shots have to form together with the metadata a archive entry.

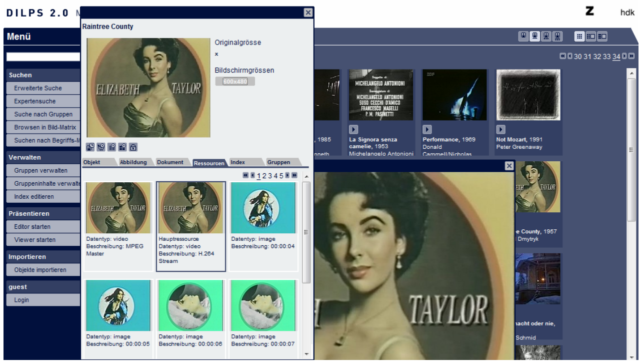

Figure 7*: **Video Work with master, stream and video shot resources***

To gain the performance required, the whole ingest infrastructure is implemented as a fully automatised networked grid system. The transcoding grid sends the results to a storage disposer, which collects the components and prepares them for final storage. The object will be delivered to their final destination, where they will reside and are available for the media access systems (playout).

Storage

The sustainability of an archive system depends essentially on the quality of the storage system. The archive host has to find a pragmatic balance between the sustainability of the storage system and the (initial and current) costs.

The metadata store will not be subject of further comments because normally the storage and backup are done via standard database management systems. These products incorporate the functionality needed.

In order to figure out the storage requirements, there must be some estimations on the quantity of media objects and the value of the objects. The digitized copy of a unique video artwork with an unusable physical master is of much higher value than the derived streamable video object. This means, that the archive maintainer should basically distinguish between the so called 'digital master' and its derived objects. Depending on the value of these media objects, the storage system has to be selected carefully. A balance between chance of data loss, storage price and value of the stored objects has to be met.

Special characteristics of digital media objects provide the possibility of using cost efficient storage systems. Digital objects within archives aren't subject of change. This leads to the perception, that data backup must be done only ones (while or directly after ingest). Due to this, there is no need for high end storage systems which are capable of snapshots, block based real time mirroring etc. The main constraint is, that ingest, backup and playout systems must be able to access the storage.

Example: Media Archive with TV Recordings

A typical storage infrastructure based on NAS (Network Attached Storage) system could look like the following structure on an archiving infrastructure.

```
10.10.10.202:/share/main0       3.7T /media/mestore02/main0
10.10.10.202:/share/main1       3.7T /media/mestore02/main1
10.10.10.205:/share/icst1       2.8T /media/mestore05/icst1
10.10.10.206:/share/main10      6.8T /media/mestore06/main10
10.10.10.206:/share/main11      6.8T /media/mestore06/main11
10.10.10.201:/vol/media/main    4.6T /media/mestore01
10.10.10.206:/share/main12      6.8T /media/mestore06/main12
10.10.10.205:/share/icst0       1.5T /media/mestore05/icst0
10.10.10.206:/share/main13      6.8T /media/mestore06/main13
```

This small infrastructure with approx 40TB of storage consists of several NAS systems which are mounted into a common file system. If storage has to be expanded, new NAS systems can be easily added. Other ways of managing the storage is using expandable NAS systems, distributed file systems or SAN (Storage Area Networks) systems.

The large number of files located on different places within the archive file system could necessitate an abstraction layer. A good solution for this layer would be an URN (Uniform Resource Name) based abstraction. This means, that every object will get a unique URN which is used as a fixed reference to the file. This abstraction layer could reside in a relational database system. Additional technical metadata from the media object could help to manage the large amount of files by using SQL queries. The following table shows an example of file quantities and sizes.

Exemplary file quantity overview

Master videos (MPEG2)	
Number	~3'700
Length	~4'000h
Size	~12TB
Streamable videos (H.264, small)	
Number	~3'000
Length	~3'100h
Size	~1.2TB
Streamable videos (H.264, large)	
Number	~3'000
Length	~3'100h
Size	~3.5TB
Video shots (PNG, full resolution)	
Number	~1'100'000
Size	~500GB
Video shots (medium resolution)	
Number	~1'100'000
Size	~350GB
Video shots (thumbnail)	
Number	~1'100'000
Size	~60GB

According to this, 4000 hours of mpeg2 video with a bitrate of approx 7mbit/s need up to 18TB of storage, if there are two streamable versions and video shots every ten seconds.

URN based object management

Within the storage system, every object is normally identified by its path. This identification can be used for direct file access only. If files are moved or duplicated, the handling becomes very difficult for the metadata store. As a result of these problems, file and metadata management should be separated in order to allow movements and replication of the media objects without interfering with the metadatabase. This can be avoided by using URNs as unique identifiers for media file identification. URNs are strings which starts with ‘URN:’ followed by the namespace and the identifier (ex. URN:ISBN:0-395-36341-1). The URN namespaces can be registered at IANA, but for internal use, unregistered namespaces can be defined [14]. With URNs as unique identifier, the content can be structured by using well structured URNs. The following table shows an example of a URN which can resolve to different URLs based on the required protocol. Furthermore, the object is available twice (backup).

Exemplary of a urn

URN:ZHDK:TMSMedia:Design_2004_03_TIFF:CH-1989-0531.tif	
file	file:///media/mestore06/main12/TMSMedia /Design_2004_03_TIFF/CH-1989-0531.tif
http	http://media.zhdk.ch/object/4805267
file	file:///media/mestore02/main10/TMSMedia /Design_2004_03_TIFF/CH-1989-0531.tif

The *URN resolver* is responsible for this task, but could have much more functionality. The URN database can keep track on digital copies on any media (e.g. tapes, removable disks) and enables the use of backup copies for load balancing systems etc. Due to the fact, that copies of digital media object must be identical, there is no need for the definition of a master.

Meta Archives

Meta archives are archive systems, which allow searching within third party archives. They differ from search engines on the fact, that they provide much more structured metadata on the archival objects.

Metadata model & harvesting

The ‘Open Archives Initiative’ (OAI) is the most commonly used resource for meta archives [15]. OAI provides a ‘Protocol for Metadata Harvesting’ (OAI-PMH) which defines a mechanism for harvesting metadata. There are open source software tools, which allow to implement a metadata harvesting system based on OAI-PMH at minimal costs. There is no special metadata model related to the harvesting protocol, but unqualified Dublin Core should be used to provide a basic level of interoperability. The MARC 21 XML Schema (MARCXML) is a good starting point because of its compatibility to library database systems, which are using the MARC standard as exchange format [16].

Network of Libraries

A special type of meta archives are networks of connected libraries. They normally use controlled vocabularies and authority lists to provide high quality comparable metadata entities. In Switzerland one of the largest network alliances is NEBIS, which uses a MARC based metadatabase [17]. There are more than eighty libraries which publish their content.

If a media archive is connected to such a library, the central MARC store could be used to distribute the archive content and store the metadata.

Figure 8: *MARC data record of a video within the NEBIS website*

Additional information about the media object allows simple implementation of nice looking web pages based on MARC records from the library system. They can be queried in real time from the central store or from a MARC dump, which is copied into a local database. It is helpful to include a link (see Figure 8, line 5) to the new representation of the work into the MARC record in order to achieve visitors from the originating library system.

Figure 9: *The MARC data record with ‘facelift’*

Example: European Media Art Meta Archives

Within the field of contemporary media art, it is quite problematic to implement a meta archive which allows the comparison of several international content providers. This is basically the result of the fact, that extensive controlled vocabularies and authority lists are not available. The project ‘GAMA – Gateway to Archives of Media Art’ (co-funded by the Community programme eContent*plus*, 2007-2009) has to implement a meta archive with media art related archives of initially ten european content providers [18]. The harmonization of the records is one of the main goals of the project. The project will provide several tools, which will help harmonizing harvested metadata without changing the original metadata records. The search engine of GAMA will support these harmonized metadata records within the platform.

Besides harvesting metadata, there is a different possibility of building a meta archive. An example is given by the Open Archiving System with Internet Sharing (OASIS-Archive) (co-funded by the Community within the framework of 'Culture 2000') [19]. This search engine does not copy any metadata to a central data store. All content resides at the partners repositories. In order to enable the communication interchange, it uses a software called *database adaptor* (DBA). The DBA is installed on each attached database system and enables real-time queries onto the connected archives. One of the advantages of this system is, that the content providers have full control over their content and any update is immediately available for the meta archive queries.

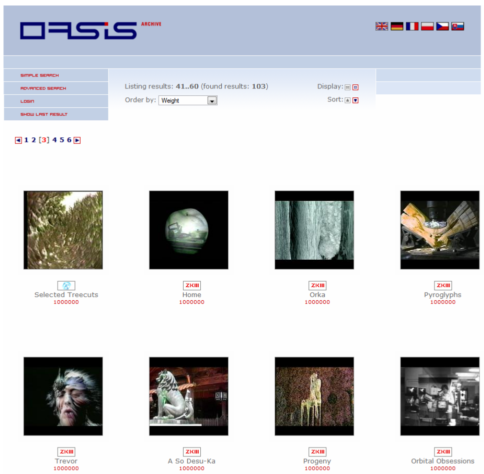

Figure 10: *Meta Archive Search Engine OASIS-Archive*

Asset management systems

Within the open source community, there are different solutions, which are capable of handling digital assets or metadata. The following three products are a very small excerpt of available systems.

Fedora Commons provides open-source software to ensure durability and integrity of digital content, use semantics to contextualize and inter-relate content from many sources, and to enable the creation of innovative, collaborative information spaces. (http://www.fedora-commons.org).

DSpace captures your data in any format – in text, video, audio, and data. It distributes it over the web. It indexes your work, so users can search and retrieve your items. It preserves your digital work over the long term. DSpace provides a way to manage your research materials and publications in a professionally maintained repository to give them greater visibility and accessibility over time. (http://www.dspace.org).

OpenCollection is a full-featured collections management and online access application for museums, archives and digital collections. It is designed to handle large, heterogeneous collections that have complex cataloguing requirements and require support for a variety of metadata standards and media formats. (http://www.opencollection.org/).

References

[1] The Dublin Core Metadata Initiative (DCMI), http://dublincore.org/
[2] Library of Congress: MARC Standards, http://www.loc.gov/marc/
[3] The International Federation of Library Associations and Institutions (IFLA), http://www.ifla.org
[4] IFLA: Functional Requirements for Bibliographic Records (1998), http://www.ifla.org/VII/s13/frbr/frbr.htm
[5] IFLA, DIGITAL LIBRARIES: Metadata Resources (2005), http://www.ifla.org/II/metadata.htm
[6] The Library of Congress: PREMIS, http://www.loc.gov/standards/premis/
[7] ZKM/ICST: IDEAMA Archive, http://ideama.icst.net
[8] IMEB/ICST: IMEB Archive, http://imeb-archive.icst.net
[9] DILPS, http://www.dilps.net
[10] JEITA: Exif, http://it.jeita.or.jp/document/publica/standard/exif/english/Exife.pdf
[11] ITPC, http://www.iptc.org/
[12] ImageMagick, http://www.imagemagick.org
[13] Exiv2, http://www.exiv2.org
[14] Internet Assigned Numbers Authority (IANA): Uniform Resource Names (URN) Namespaces, http://www.iana.org/assignments/urn-namespaces
[15] Open Archives Initiative (OAI), http://www.openarchives.org/
[16] Library of Congress: MARCXML, http://www.loc.gov/standards/marcxml/
[17] Netzwerk von Bibliotheken und Informationsstellen in der Schweiz (NEBIS), http://www.nebis.ch
[18] Gateway to Archives of Media Art (GAMA), http://www.gama-gateway.eu
[19] Open Archiving System with Internet Sharing (OASIS-Archive), http://www.oasis-archive.eu

Author Biography

Jürgen Enge is an expert in development and research on media art archives. He received his diploma in information technology from Karlsruhe University (TH) in 2000. From 2001-2003 he was head of the Zurich University of the Art (ZHdK) study program 'Mobile Application Design'. Besides managing the institute for net development at the ZKM (2003 -2004) he has been leader of the EU-project OASIS Archive. Presently he is scientist at ZHdK and Karlsruhe University of Arts and Design and works at the EU-project GAMA.