

A Knowledge Schema for a preservation of audio-visual objects

Loredana Versienti¹

¹ ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, versienti@isti.cnr.it

Abstract. This paper describes the knowledge schema for expressing the digital preservation. Digital audiovisual preservation is nerve of the research nowadays in this digital world, where use of audio visuals in creation and storage of research data has increased rapidly. Thereby it has created many opportunities for new problems regarding their maintenance, preservation and future accessibility. Lack of awareness about the preservation tools and applications is a big issue today. To express the audio visual needs formally it has mapped a knowledge schema. The knowledge schema was first cut and needed evaluation, so it's being evaluated through this study in terms of its ability to represent the Needs of different communities of practice, classes, their association and ability to represent requirements of Audiovisual community through properties of its classes.

Keywords: Audiovisual preservation, knowledge schema, OAIS reference mode

Introduction

In the course of the twentieth century, audiovisual media became the prime means of capturing human experience at every level from the personal to the most public, recording everything from the minutiae of family life to the great events of modern history. The world of audiovisual media is rapidly changing and the preservation of the historic record is no longer the exclusive preserve of memory institutions, but an opportunity for many thousands of organization and millions of people worldwide: recording and digitization tool have become widely affordable, and sharing and storage platforms online are easily accessible and robust. At the same time, many of the organizations, companies and individuals producing digital audiovisual content have little idea of the issues of digital preservation and are severely limited in terms of resource, capacity and expertise. Digital preservation is the first essential step in a chain of activities that runs through cataloguing, restoration, and access to exploitation.

There is an urgent need for organization at every level to learn about the technologies available, evaluate tools and services evolving in widely separated institutions, and exchange the experience and knowledge spread across a wide range of commercial and non-commercial domains to develop suitable approaches for different application.

The objective of this technical report is to present the schema for expressing the digital preservation. The schema is required in order to structure the needs in forms of formal

knowledge statement that can be automatically processed in order to support the match of the needs with the research output.

The Knowledge Schema

Inside the Presto4U project has been identified nine areas interested in audiovisual preservation and for each one it created a Community of Practice (CoP). [1]

The nine areas are:

- Music and Sound Archives
- Video production and postproduction
- Film collection and filmmakers
- Personal collection
- Broadcast
- Art & museum object
- Learning & teaching repositories
- Footage sales libraries
- Research and scientific collection

The knowledge schema is divided into a header and five section. A UML representation of the classes comprising the schema is reported in Figure 1.

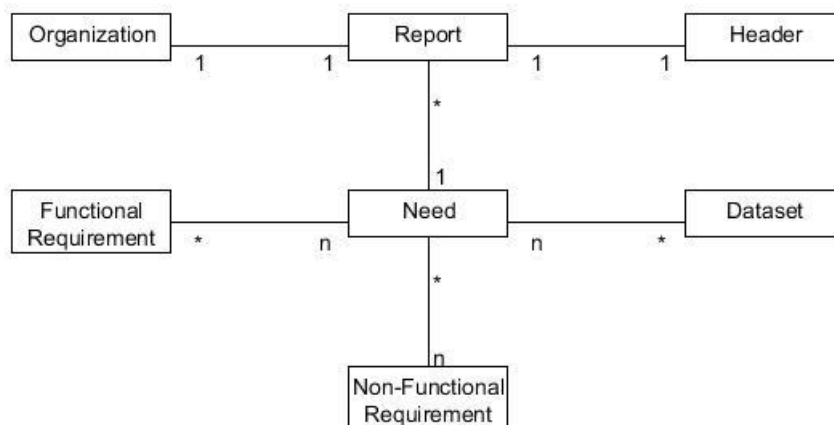


Fig 1 Classes and associations in the knowledge schema

The Report class represents the reports that are produced in the knowledge transfer process that takes place in the Presto4U project. Each report that is an instance of the class Report, is the result of a *knowledge transfer event*. An event of this kind takes place every time a CoP member provides knowledge about their preservation needs, by using the knowledge schema as a template. Each report is associated with an instance of the class Header that reports salient features of the knowledge transfer event that produced the report. In particular, the Header of the knowledge schema allows to capture:

- the Cop where the person(s) providing the knowledge belongs
- identification of the person(s) that provide the knowledge

- time and place of the event
- the modality of the event, that is a statement whether the report was generated autonomously or in an assisted way
- the specific circumstances in which the event occurs, if any (such as a project, meeting, a conference, and the like)
- a last modification date, in case the event occurs in successive stages
- an annotation about the event, reporting details that are deemed as relevant.

The first four properties of the header are mandatory, in the sense that a value for them must be provided in every report.

The section of the Knowledge schema are:

1. *Organization*
2. *Need*
3. *Dataset*
4. *Functional requirements*
5. *Non-functional requirement*

Each of these section is detailed in a separate section below.

Organization

This section reports the instance of the class Organization that is associated one-to-one with a Report. The class provides properties for describing the main features of the reporting CoP member. It includes the following properties:

- *Mission*: a text describing the mission of the organization of the reporting organization
- *Position in the media lifecycle*: a text describing where the organization is placed in the cycle from content creation through "exploitation" and then coming back to content re-use. One form of re-use is a part of content creation - hence the cycle.
- *Position in economic and political space*: a text describing the organization from the economic and political point of view (a commercial company and a publicly funded heritage institution differ in many ways)
- *Usage of media*: a text describing how the organization uses the A/V media, the purpose (end) and the business (means)
- *Usage of technology*: a description of what technology the organization uses, in terms of hardware, software and methodologies.

Need

This section reports an instance of the classes Need that describes a presentation need of the reporting organization. An instance of the class Need is associated to a report, and a report will in general have one or more needs.

The class Need includes the following properties:

- *Need*: a statement of the need, expressed as much as possible in precise and technical terms
- *Involved Dataset*: an identification of the dataset(s) on which the need arise. Each of these datasets is described in detail in a separate section; the value of this property must therefore identify one or more of the separately described datasets.
- *Current used technology*: a text describing the product, service or methodology that is currently used to satisfy the need, if any.
- *Reason of dissatisfaction*: a description of the reason(s) why the current technology or service is not satisfactory.
- *Desired technology or service*: if there is an existing service or technology that can satisfy the need, it is described here, while the next property gives the barriers that prevent the adoption of the identified technology or service.
- *Barriers*: barriers that prevent the adoption of a desired, existing technology.
- *Requirements*: if the organization is not aware of any technology or service that can satisfy the need, then one or more requirements are here identified that express the need both in functional and non-functional terms. A detailed description of the requirements identified here is given by instance of the classes "Functional requirement" and "Non-functional requirement".

Dataset

Every reported need applied to one or more datasets held by the reporting organization. Each such dataset is described by an instance of the class Dataset.

There many standards that are adopted for the description of datasets, including MPEG ones. The properties of this class are based on the Multimedia Ontology [1] recommended by the World Wide Web Committee, indicated by the Broadcast CoP as a viable mean of dataset description.

The ontology includes the following properties:

- *identifier*: a tuple identifying a resource, which can be either an abstract concept (e.g Hamlet) or a specific object, using a URI. The type can be used to optionally define the category of the identifier.
- *title*: a tuple providing the title or name given to the resource. The type can be used to optionally define the category of the title.
- *language*: the language used in the resource. Recommended best practice is to use a controlled vocabulary such as [BCP47].
- *locator*: the address at which the resource can be accessed (e.g. an URL, or a DVB URI)
- *contributor*: a tuple identifying the agent (with either a URI, if it exists, or plain text) and the nature of the contribution, e.g. actor, cameramen, director, singer, author, artist.
- *creator*: the author of the resource and the role. The author identifier can be defined as either an URI (which is best practice) or as plain text. The role is defined as plain text.

- *createDate*: the date defines the data and time that the resource was created. The type defines the particular category of creation date (e.g. release date, date recorded, date edited).
- *location*: a location name and/or data where the resource has been shot/recorded.
- *description*: free-form text describing the content of the resource
- *keyword*: a concept, descriptive phrase or keyword that specifies the topic of the resource. A recommended best practice is to take this keyword from an ontology or a controlled vocabulary.
- *genre*: the category of the content of the resource. Recommended best practice is to use an ontology or a controlled vocabulary such as the EBU vocabulary
- *rating*: a tuple defining the rating value, the rating person or organization (as a URI or a string), and the voting range (min value, max value).
- *relation*: a tuple identifying a resource to which the current resource is related and optionally, the nature of the relationship. An example is a listening of content that has a relationship (possibly a named) to another content.
- *collection*: the URI (best practice) or the name of the collection from which the resource originates or to which it belongs.
- *copyright*: the copyright statement associated with the resource and optionally, the identifier of the copyright holder.
- *policy*: a description of the security policy applying to the media resource, or a reference to the security policy (e.g. Creative Commons). The type attribute can be used to provide more information as to the nature of the security policy (e.g. permission, access control, ownership).
- *publisher*: the publisher of the resource.
- *targetAudience*: a tuple identifying the issuer of the classification (parental guidance, issuing agency, targeted geographical region) and the value given in this classification.
- *fragment*: a tuple containing a fragment identifier and its role. A fragment is a portion of the resource, as defined by the [MediaFragment] Working Group.
- *namedFragment*: a tuple containing a named fragment identifier and its label.
- *frameSize*: the frame size of the resource, if applicable. For example: w:720, h:480. It is optional to specify the units; the default value is the pixel.
- *compression*: the compression type used. For container files (e.g. QuickTime, AVI), the compression is not defined by the format, as a container file can have several tracks with different encodings. In such case, several compression instances will exist. Thus, querying the compression property of the track media fragments will return different values for each track fragment. [RFC 4281].
- *duration*: the actual duration of the resource. The unit defined to be seconds.
- *format*: the MIME type of the resource (e.g. wrapper, bucket media types).
- *samplingRate*: the audio sampling rate. The unit is defined to be samples/second.
- *framerate*: the video frame rate. The unit is defined to be frames/second
- *averageBitRate*: the average bit rate. The unit is defined to be kbps.

- *numTracks*. the number of trucks of a resource, optionally followed by the type of truck (e.g. video, audio, subtitle).

Functional requirement

Every need may be associated to one or more requirements. Each requirement, in turn, is described by an instance of this class for what concern the functional aspects of the requirement.

Typically, a functional requirement is equated with a use case, and described in terms of a set of de-facto standard properties. For the case at hand, these properties may be too detailed, in the sense that the owner of the requirement may not be able to give a precise account of, eg. pre-conditions, post-conditions or sequence of steps of the required functionality. In this case, only the generic properties of the class will have to be used in the report, such as actor, description, notes and maybe a few others.

The properties for describing a functional requirement are:

- *actor*: the role within the organization that owns the requirement, that is that going to use the required functionality.
- *description*: a brief description of the reason for and outcome of the use case, or a high-level description of the sequence of actions and the outcome of executing the use case.
- *notesAndIssues*: a list of any additional comments about use case or any remaining open issues
- *normalCourseOfEvents*: a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions.
- *alternativeCourses*: a description of courses of events that are less common but not exceptional.
- *exceptions*: any anticipated error conditions, that could occur during execution of the use case, and define how the system is too respond to those conditions.
- *includes*: any other use cases that are included by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need the common functionality.
- *specialRequirement*: indentify any additional requirement, such as non-functional requirement, for the use case that may need to be addressed during design or implementation. These may include performance requirement or other quality attributes.
- *assumptions*: list any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.
- *pre-conditions*: list any activities that must take place, or any conditions that must be true, before the use case can be started.
- *post-conditions*: describe the state of the system at the conclusion of the use case execution.
- *priority*: indicate the relative priority of implementing the functionality required to allow this use case executed.

- *frequencyOfUse*: estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

Non-functional requirements

Every Need may be associated to one or more requirement. Each requirement, in turn, is described by an instance of this class for what concern the non-functionality aspects of the requirement.

Non-functionality requirements describe the desired qualities of the functionality. These parallel assessment criteria against which software can be evaluated, therefore the relevant standards apply. In particular, we suggest the adoption of the international standard ISO/IEC 25010 (System and Software Quality Requirement and Evaluation SQuaRE - System and Software Quality models) as assessment reference model. Needless to say, the adoption of this standard also for the assessment of research outputs will reduce matching (of needs and products) to comparison of different objects along the same dimension.

This first edition of ISO/IEC 25010 cancels and replaces ISO/IEC 9126-1:2001, which has been technically revised. The international standard 25010 defines the following properties, classified as "Quality in use" and "Internal and external quality", and further hierarchically structured in characteristics and sub-characteristics.

Conclusion and Future Work

We have presented a knowledge schema to be used for expressing the preservation needs. Based on this conceptualization, we have been looking at several ontologies in order to identify vocabulary elements that are appropriate for specifying the conceptualization. In this process, we have considered general ontologies for requirement specification (both functional and non-functional), and specific ontologies for the description of A/V contents and metadata.

In particular, it is expected that the usage of the knowledge schema will allow:

- to identify a common schema, applicable into the nine areas
- to identify a suitable vocabularies for the properties that are specific of the schema; usage of controlled vocabularies in the report is paramount for mechanizing, at least in part, the process of matching needs and research products.

References

- [1] Video Format Identification Guide <http://videopreservation.conservation-us.org/>
- [2] FADGI Glossary <http://www.digitizationguidelines.gov/glossary.php>
- [3] EBUCore <https://tech.ebu.ch/MetadataEbuCore>
- [4] Multimedia Vocabularies on the Semantic WEB <https://www.w3.org/2005/Incubator/mmsem/XGR-vocabularies/>
- [5] W3C Media Ontology <https://www.w3.org/TR/mediaont-10/>
- [6] MPEG <https://www.vocabulary.com/dictionary/MPEG>