The Data Model in Clipper

John Casey Project Manager

This project report reviews the data model used in Clipper and discusses option for future development.

The Clipper project would also like to acknowledge the input and advice of Gayle Calverley, the project manager of the JISC TBMAP study. Any analysis or conclusions based on these inputs is entirely the responsibility and opinion of the author.

The Clipper tool is based almost entirely on the concept of metadata (data about data) and uses it to create and manage representations of *Resources*, *Clips*, *Annotations* & *ClipPlaylists.* These terms in italics, which we shall call Clipper 'elements' (are 'things' a user can create with the Clipper tool. We have chosen these elements to simplify discussions in the project, and allowing the same vocabulary to describe the production elements of the Clipper tool design (an example of a 'local' controlled vocabulary). The Clipper export playlist is an xml document (which uses a .che[1] file extension) and is really mostly a metadata record. That said, the main thrust of this first phase of the Clipper project is to build a prototype tool that concentrates on delivering certain functions in manipulating online video. As expected, this has helped to clarify what potential users need and to give an idea of how to further develop the metadata side of the tool.

This short document has been informed by some useful studies and reports into metadata that have been funded by JISC:

- Learning Materials Application Profile Study[2]
- Time-based Media Application Profile to Support Search & Discovery (TBM-AP)[3]
- Semantic Technologies in Learning & Teaching[4]

The metadata structures in Clipper map roughly onto the work of the TBMAP and at the end of this document we discuss this and other areas for further development.

The video resources that the Clipper project is concerned with are those that can be used in any educational context (training, teaching, learning, research), which is very broad. We see many of our potential users operating firmly in an individualized and 'casual' mode, using Clipper to manage and share their online video resources neither knowing, or caring about metadata or standards. On the other hand, we also see the possibility of using a tool like Clipper in a more 'serious' information management mode, such as for supporting archivists, researchers, and students to find, access,

[1] .che = 'Clipper Hierarchical Expression' format

[2] http://www.icbl.hw.ac.uk/lmap/lmapreport.d3.pdf

[3] http://wiki.manchester.ac.uk/tbmap/index.php/Main_Page

[4] http://www.jisc.ac.uk/publications/documents/semantictechnologiesreport.aspx

create and share their Clips and Annotations. A study by the Cancore[5] metadata project in Canada nicely describes these apparently contradictory requirements for metadata indicating that they:

- present a highly simplified, "kernel" for those wishing to work with only very basic metadata records.
- accommodate a high level of complexity in terms of element structure and semantics, but which enable, not impose, complexity.

So, what do we do? Here is a brief summary of where we are up to in our thinking:

- The Cancore approach seems a sensible way to go, i.e. offer a lightweight metadata framework but with the possibility of accessing more complex or specialized facilities[6]. Further development of Clipper should accommodate this.
- Use automated metadata creation[7] to gather available information even if it results in fields populated with “Put your description here” or “Untitled Clip”.
- Understand and allow for the fact that different people and automatic agents may well share metadata creation over time (ref only needed here).
- Never force a user fill in a metadata field (they have to want to). Allow users to go back and edit if they need to(an automatically generated unique identifier for each Clipper element/item is the backstop in Clipper).
- Understand that there is a considerable amount of visual and spatial ‘metadata’ available to a user of a tool like Clipper, that should not be discounted.
- Use the existing system info/metadata architecture as much as possible. For instance there can be lot of metadata in the video file (depending on the original encoding) as well as the file name and location. The spread of the Adobe Flash format makes this easier.
- Explore using Adobe XMP (Extensible Metadata Platform) further for embedded metadata[8]
- Seek to inform the work in Clipper with the JISC Semtech report (http://www.semtech.ecs.soton.ac.uk), particularly in the area of ‘soft’ semantic technologies.
- Metadata will need to be embedded in the Clipper.che document for transport.
- Seek to align the work of Clipper with the work of the W3C Media fragments Group.[9] [10] [11] [Note: in Clipper the Annotations created by users are neither

[5] http://www.cancore.ca/betterlom.html

[6] It’s worth noting this approach has been adopted by the UK Jorum learning object repository project in the light of experience and user feedback

[7] The JISC LMAP report points out that a lot of metadata is already generated by systems (such as cameras) and it makes sense to at least store it for further use

[8] Here is a useful intro to XMP http://www.adobe.com/products/xmp/overview.html and here is a good presentation about XMP and relations to RDF http://www.linkarchy.com/XMP-Presentation.pdf

[9] See http://www.w3.org/2008/01/media-fragments-wg.html

[10] Interestingly, the W3C seem to be making use of XMP – here they describe using it as a ‘pivotal vocabulary’ http://www.w3.org/2008/01/media-fragments-wg.html

[11] It seems likely that the W3C work aimed at creating a ‘URL syntax to enable media fragments’ will fit into the Clipper tool architecture especially as we are examining using the XMP framework, which is also favoured by the W3C. Although there is likely to be a

> subtitles nor timed text so we shall not be looking at the W3C 'Timed Text' project to inform our work'].

One of the issues we face in Clipper, in common with scenarios described in the LMAP report, is that the metadata we are dealing with is brought together from different sources (machines and humans) rather than originating in a single instance of creating a record. So, we are faced with "a set of statements from diverse sources, each describing specific entities or properties" (LMAP p. 37), which we will need to collect together and transport in the Clipper.che document.

In connection with automated metadata generation we have taken the approach that there is lots of 'free' metadata that already exists inside and around the resources we are dealing with. Part of this is inspired by the work of the Information Retrieval research community, for instance the idea of using the common internal structures of resources to identify and produce metadata 'on the fly' to assist with searching and retrieval[12]. This approach was also raised and discussed at the CETIS JISC conference in 2008[13] in the context of Open Educational Resources.

So, we might characterize the kind of approaches that a tool like Clipper should aim to support as a mix of:

- Very basic lightweight metadata (the current approach)
- The canonical model of professional cataloguers and metadata standards
- Personal tagging as epitomized in folksonomies
- Shared metadata creation over time by humans and machines
- Automated.

The trick is, as Cancore points out[14] to choose the right metadata regime to fit your needs and means and not be bound by them. It's a question of choosing the right tools from your information toolbox, and recognizing that there is more than one tool in the box!

Clipper Information Model and XML Data Structure

There is one Clipper XML document; *clipper.che*, which is used for export and import. This contains information about the Resources, Clips and Annotations that combine to make up a ClipPlaylist. As this is a very short project whose final output will be a working prototype, we are initially working with a very lightweight metadata structure.

time gap before that and HTML 5 appears, a tool like Clipper could and should be built with these developments in mind – to enable later alteration to work with the new standards

[12] See 'Information Retrieval and Structured Documents' by Yves Chiaramella in *Lectures on Information Retrieval* (Third European Summer-School, ESSIR 2000 Varenna, Italy, September 11–15, 2000) Edited by Maristella Agosti, Fabio Crestani, Gabriella Pasi, Springer Berlin / Heidelberg (2001) ISBN 978-3-540-41933-4

[13] http://blogs.cetis.ac.uk/johnr/2008/12/08/open-educational-resources-metadata-and-self-description/

[14] http://www.cancore.ca/betterlom.html

In the Clipper application, when a Clipper element is created (*Resource, Clip, Annotation, ClipPlaylist*) a set of Properties (or metadata) are created for each element and stored in the database (they constitute metadata schema, initially very loosely based on the Dublin Core standard and MPEG 7). These Property fields are automatically generated and where possible automatically populated with data (that the system has access to). The Properties are also offered to the user to edit and finally stored back in the SQL database within Adobe AIR. The Clipper wireframe prototypes document illustrates the proposed Properties interfaces.

The initial Clipper metadata schema is as follows below, the indents show a parent-child relationship between the Clipper elements along these lines; Resource>Clip>Annotation. Note: 'auto' indicates that the values are generated automatically 'Date created/last modified' refers to the metadata record

Resource

- Title for the Resource – auto from the last part of the URI or Path, but the user can edit
- Description – auto but user can edit
- Duration – auto from query of .flv file properties/metadata
- Original & Local Addresses – auto from system
- Date created/last modified – auto from system (useful for version control)
- Resource Identifier – auto (Number)

  Clip

    - Title for the Clip – auto/user generated
    - In and Out points for the clip – auto from the user set in/out points with reference to the duration of the parent resource but user can edit
    - Description – auto but user can edit
    - Duration – auto from subtraction of the In value from the Out value
    - Date created/last modified – auto from system (useful for version control)
    - Clip Identifier (Number)

    Annotation

        - Title for the Annotation – auto/user generated
        - Insert Point – auto from the time code at insertion
        - Description – auto but user can edit
        - Date created/last modified – auto from system (useful for version control)
        - Annotation Identifier – auto (Number)

ClipPlaylists

A Clipper.che document is created when a user decides to export a ClipPlaylist from Clipper. This document carries all the information needed to recreate the content of a ClipPlaylist for another user of Clipper. This means that the Properties (metadata) for the parent Resources and their selected child Clips are included as well as those of the Annotations that are attached to each Clip. Note: The actual text for each annotation is also included in the Clipper.che document.

The ClipPlaylist also has its own Properties:

ClipPlaylist

- Title – auto/user generated
- Duration – auto from the sum of the duration of the constituent Clips
- Description – auto/user generated
- Date created/last modified – auto from system (useful for version control)
- Clip Playlist Identifier – auto (Number)

Note: once a Clip (and its annotations) is included in a ClipPlaylist it cannot be edited. This is to prevent recursive actions that might result in infinite loops and breaks etc especially when referencing parent elements in Clipper that also might have been deleted or modified by the user since the ClipPlaylist was created.

XML Structure

A Typical Clipper.che document structure is shown below with the properties in a 'collapsed view' (without the constituent metadata for the Clipper elements). Note the multiple annotations in Clip 2 of Resource 1 and that Clip 1 in Resource 2 has no Annotations.

```
Declarations etc
        ClipPlaylist
                Resource 1
                        Clip 1
                                Annotation 1
                                        Annotation Text
                        Clip 2
                                Annotation 1
                                        Annotation 1 Text
                                Annotation 2
                                        Annotation 2 Text
                                Annotation 3
                                        Annotation 3 Text
                        Clip 3
                                Annotation 1
                                        Annotation1 Text
                Resource 2
                        Clip 1
                Resource 3
                        Clip 1
                        Annotation 1
                                Annotation 1 Text
```

Mapping the Clipper Data Model to demonstrate alignment with TBMAP FRBR approach

Here is an overview of the FRBR approach by way of introduction[15], from the IFLA (International Federation of Library Associations and Institutions) final report on FRBR (Functional Requirements for Bibliographic Records):

"The entities defined as *work* (a distinct intellectual or artistic creation) and *expression* (the intellectual or artistic realization of a *work*) reflect intellectual or artistic content. The entities defined as *manifestation* (the physical embodiment of an *expression* of a *work*) and *item* (a single exemplar of a *manifestation*), on the other hand, reflect physical form."

And basically they are not all 1-1 relationships, e.g. one artistic concept can be expressed in more than one way to give expression more than once that look different, giving lots of manifestations), and finally you can have the same content in several formats (item) etc:

"The relationships depicted …. indicate that a *work* may be realized through one or more than one *expression* …. An *expression*, on the other hand, is the realization of one and only one *work*. An *expression* may be embodied in one or more than one *manifestation*; likewise a *manifestation* may embody one or more than one *expression*. A *manifestation*, in turn, may be exemplified by one or more than one *item*; but an *item* may exemplify one and only one *manifestation*."

Here is a useful summary of the four concepts:

1. **Work (**a distinct intellectual or artistic creation**)**
2. **Expression (**the intellectual or artistic realization of a *work***)**
3. **Manifestation (**the physical embodiment of an *expression* of a *work***)**
4. **Item (**a single exemplar of a *manifestation***)**

A Real World Example[16]
Taking the film Jules and Jim as an example, the concept of the film is a distinct **Work** - it can have different **Expressions** such as the original French version and the version with English subtitles added, which are considered two separate **Expressions** – or realisations - of that same **Work**. Similarly an original version and a "director's cut" might be considered two different **Expressions** of the same **Work**. Any intentional change in content signals the existence of a new **Expression, as long as they hold true to the original concept that defines the Work**.

And for any one of those **Expressions**, releases in the form of e.g. a 35mm film, a VHS video cassette, a DVD, a stream in Windows Media Format, and a downloadable QuickTime form would be considered five distinct **Manifestations** - embodiments - of that **Expression** - as long as the content in each case remained the same.

[15] This taken from the TBMAP project

[16] http://wiki.manchester.ac.uk/tbmap/index.php/ModelOverview#FRBR_.26_Moving_Images

Finally, individual copies - concrete instances - of those forms are considered separate **Items** (e.g. the two different copies of the 35mm film held in two different film archives, or the copy of the DVD in a library as distinct from the copy purchased by an individual).

Mapping the Clipper Data Model to the TBMAP

The terms and figures in brackets refer to the TBMAP 'Description Set Profile' and this mapping is provisional[17].

Note: One thing that this comparison has raised for future attention is the tricky question of dates – in our case the need to distinguish between the date of creation and changes to the properties/metadata record about a Clipper element and the date of creation and changes to a Clipper element itself e.g. a change to a Clip 'in point'. In the next version of Clipper we need to provide extra metadata for the Clipper element creation/alteration date and separate the metadata creation/alteration date information into a separate meta-metadata section. Other points for attention are identified below.

Clipper '**Resource'** properties mapped to TBMAP

A Clipper Resource from the TBMAP perspective might be viewed as an Expression of a Work.

- Title for the Resource – auto from the last part of the URI or Path, but the user can edit (1.3.5 Expression Title – and would normally be the same as the Work)
- Description – auto but user can edit (1.2.5 Work: Abstract)
- Duration – auto from query of .flv file properties/metadata (Expression: Duration 1.3.13)
- Original Addresses – auto from system (1.3.3 Expression Identifier - this assumes that original address is a reference to where the Expression is stored)
- Date created/last modified – auto from system, useful for version control, this refers to the date of the creation of the metadata about the Resource (meta metadata).
- Date created/last modified – auto from system, useful for version control, this refers to the date of the creation/alteration of the Clipper element itself (1.3.9 Expression: Date Created.
- Resource Identifier – auto a number a local identifier in the Clipper tool, (1.3.3 Expression Identifier. This can be defined in a separate statement from the address above)

Clipper **'Clip'** properties mapped to TBMAP

A Clip from the TBMAP perspective might be viewed as a shorter Expression of the whole Work from which the 'parent expression' is drawn.

- Title for the Clip – auto/user generated (1.3.5 Expression: Title)
- In and Out points for the clip – auto from the user set in/out points with reference to the duration of the parent resource (1.3.14

[17] http://wiki.manchester.ac.uk/tbmap/index.php/DescriptionSetProfile

Expression: Extract Start Time & 1.3.15 Expression: Extract End Time)

- Description – auto but user can edit (1.3.6 Expression: Description)
- Duration – auto from subtraction of the In value from the Out value, (Expression: Duration 1.3.13)
- Date created/last modified – auto from system, useful for version control, this refers to the date of the creation of the metadata about the Resource (meta metadata).
- Date created/last modified – auto from system, useful for version control, this refers to the date of the creation/alteration of the Clipper element itself (1.3.9 Expression: Date Created.
- Clip Identifier – auto a number (1.3.3 Expression: Identifier)
- N.B. Development note – (Needs a TBMAP property 1.3.21.4 Is Extract Of, to identify relationship with the parent Resource)

Clipper **'Annotation'** properties mapped to TBMAP

An Annotation of a Clip, from the TBMAP perspective might be viewed as a 'is summarization of' the Expression of the Work constituted by the Clip (1.3.21.8 Is Summarization Of). So, the metadata record for the annotation assumes the annotation to be an expression in its own right.

- Title for the Annotation – auto/user generated (1.3.5 Expression: Title)
- Insert Point – auto from the time code at insertion (1.3.14 Expression: Extract Start Time), probably easiest to automatically create a Extract End time, say 0.1 secs from the start to provide a duration the metadata can handle
- Date created/last modified – auto from system, useful for version control, this refers to the date of the creation of the metadata about the Resource (meta metadata).
- Date created/last modified – auto from system, useful for version control, this refers to the date of the creation/alteration of the Clipper element itself (1.3.9 Expression: Date Created.
- Annotation Identifier – auto a number (1.3.3 Expression: Identifier)
- Development notes – (Needs a TBMAP property of 1.3.21.8 Is Summarization Of, to tie annotation to the Clip)

Clipper **'ClipPlaylist'** properties mapped to TBMAP

N.B. A ClipPlaylist, from the TBMAP perspective, is in itself a new Work.

- Title – auto/user generated (1.2.4 Work: Title)
- Duration – auto from the sum of the duration of the constituent Clips (Expression: Duration 1.3.13)
- Description – auto/user generated (Work: Abstract 1.2.5)
- Date created/last modified – auto from system, useful for version control, (1.3.9 Expression: Date Created)
- ClipPlaylist Identifier – auto, a number (1.2.3 Work: Identifier)
- Development note - the relationships between the constituent Resources/Clips/Annotations in a ClipPlaylist could be described with:

- 1.2.12.1 Has Part (for Resources),
  - 1.3.21.4 Is Extract Of (for Clips),
    - 1.3.21.8 Is Summarization Of (for Annotations)