Video Capsule Specification

Purpose

This Capsule provides a starting point for applications that need to handle video.

Functionality

The video capsule supports the object model in Appendix A. The terminology is explained in Appendix B. The capsule provides tools or methods for:

- Installing a federated database with a preloaded schema and at least one template database.
- Creating, updating and deleting instances of each of the object classes, including the ability to link new instances to other instances, generally of a different class.
- The methods will use a sample placement model (segmenting the data into databases, containers and object clusters).
- Querying or iterating over (automatically generated) all instances of an object class .
- Querying or iterating (automatically generated) from an object to associated objects.
- Creating and traversing (“up” and “down”) many-to-many structures using the Project and Object classes. These may be replaceable by separate Team Modeler and Structure capsules.
- Versioning of Timelines, Tracks and Scenes without duplication of Clips or Frames.
- Importing MPEG-4 and JPEG2000 format video/audio files and/or streams into instances of the object classes. Channels (audio,video or effect streams) will be maintained individually.
- Exporting an MPEG-4 or JPEG2000 format video/audio file or stream from a federation, based on a Project, Timeline (or sub-division of one), Track or Scene.
- The ability to display the exported video stream using an industry standard tool.
- *To Be Determined:* A GUI, such as an ooAssist plugin, supporting the above functionality.
- There will be no Objectivity-proprietary video viewing, editing, special effect, watermarking, format translation or object recognition capability. The application builder will provide those.

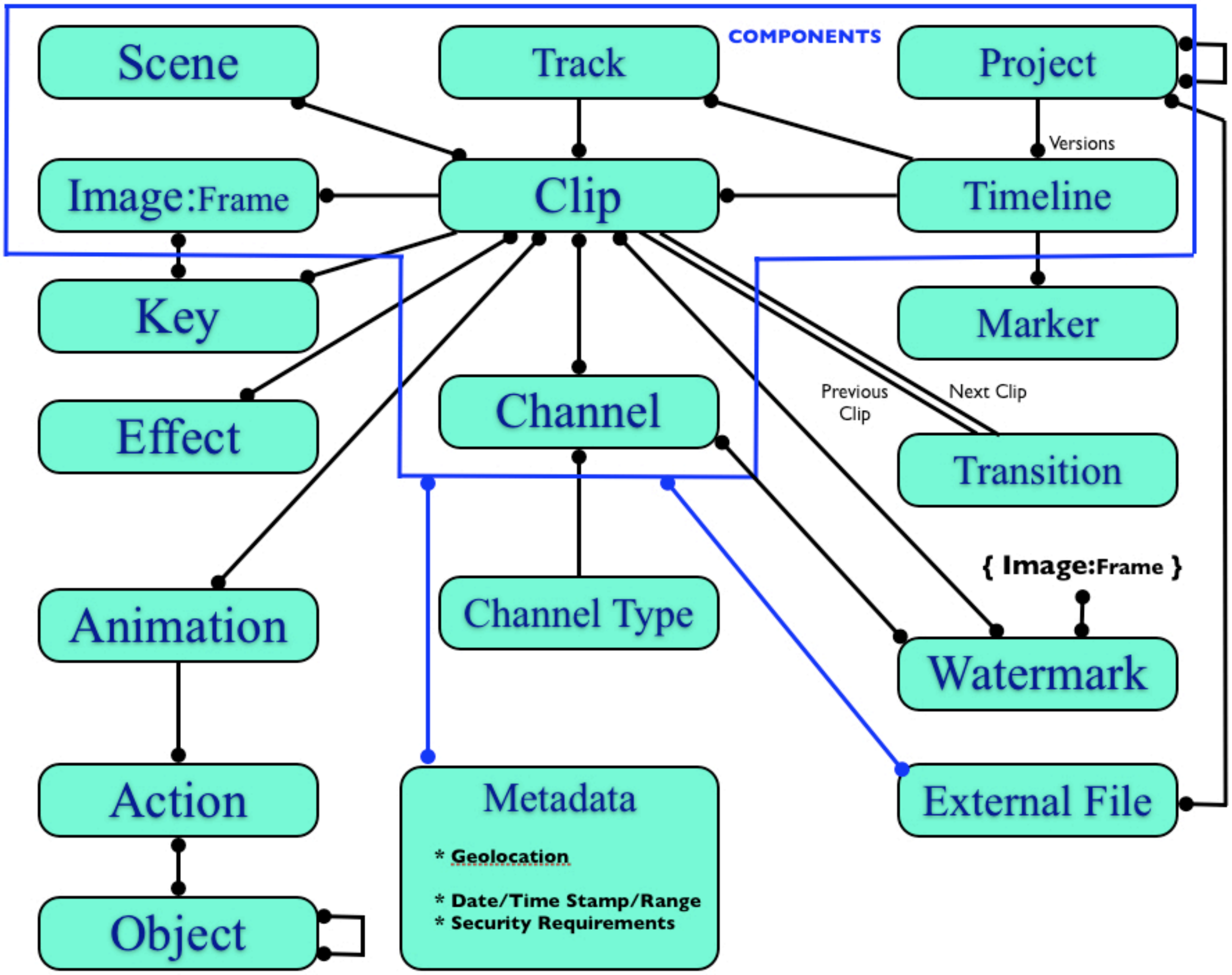
Platforms and Languages

- Windows and Linux.
- C++, Java and C# (later). Objectivity for Java and C# have no object level versioning API.

Suggested Pricing

- 60-day trial, then a one-time license fee of $250 per developer. Apple Final Cut costs $999.

Appendix A – Object Model for the Video Capsule



Appendix B - Object Class Descriptions

1. **Project** - A videography project and its related sub-projects. A sub-project may be a part of multiple projects.
2. **Clip** - A short piece of video and/or audio, often containing an individual Scene.
3. **Scene** - A single video sequence, usually shot in one continuous take. For editing purposes, it is useful to capture or trim video material to store each Scene as an individual Clip.
4. **Image** - A digital image.
5. **Frame** - An individual subclassed Image. A Clip is a sequence of Frames controlled by a Timeline.
6. **Timeline** - a collection of Clips into a production with multiple overlapping tracks. A Timeline provides a view of multiple sources being combined over time, with separate Tracks for video, audio, and superimposed video, as well as Transitions and Effects. A storyboard is a particular kind of Timeline, used to sketch out Scenes and vital Images and Clips.
7. **Track** - A sequence of video or audio Clips in a video editing Timeline that are to be combined and superimposed into a final production.
8. **Transition** - A visual effect to segue from the end of one Clip or Scene and the start of the next. The most basic transition is a cut, in which the last frame of one clip is immediately followed by the first frame of the next clip. More interesting transition effects include fades, dissolves and wipes between adjacent clips.
9. **Marker** - A placeholder used to mark a specific time-code in a sequence. Used to keep track of changes, events, or synchronization points in a longer sequence. In and Out point markers can be used to mark a Clip to be captured from a source tape, to mark part of a Clip to be trimmed, or to mark a portion of the Timeline to be played.
10. **Effect** - The result of processing audio and video Clips to enhance, improve, or distort them.
11. **Key** - A region of a Frame or video Clip to be used as a mask for transparency. Used to make part of the Scene transparent or semitransparent, and then composite it with other superimposed Images or Tracks. The region can be specified using features such as color or intensity, or with a separate alpha mask or image matte. An alpha mask, or alpha channel **is** extra information stored with an image to define transparent areas used for keying and superimpositions.
12. **Channel** - The subcomponents of a Clip. For Images, an alpha channel can contain a matte or mask image to key certain regions of the Image:Frame to be transparent. For audio, the separate left and right channels of a stereo Clip.

13. **Channel Type** - Identifies the type of encode/decode/processing algorithm required for a Channel, e.g. identifies an audio, raw video channel or alpha (transparency) channel.
14. **Metadata** - Used for geotagging, date/time stamp/ranging, security classification etc. The objects will generally be indexed (possibly in multiple ways) or include collections (sorted or unsorted) to support queries and application requirements. This functionality could be linked with other capsules, such as GIS and Security, when they become available.
15. **Components** - A superclass that can be linked with Metadata
16. **Watermark** - A small, semitransparent graphic that identifies a scene or speaker. Most TV broadcasts use a watermark to let you know which channel you're watching.
17. **Object** - Any feature, physical object or computed artifact that appears in one or more Frames, such as a title, a superimposed logo or a cartoon character. Networks of objects can be created to represent groups of vehicles, parts of a complex structure etc.
18. **Action** - An operation that changes the appearance, visibility or position of an Object.
19. **Animation** - A sequence of Actions that move and manipulate an object over time.
20. **External File** - Retains information (file type, file path, name etc.) about imported and exported files. This may be replaceable by a separate File Management capsule.