KMS Models for Video Files using Visual Mnemonics

Mike Leggett
Creativity & Cognition Studios, University of Technology Sydney, Australia.
+612 9514 4633 mleggett@it.uts.edu.au

ABSTRACT
A series of Models were built to explore and test the precept of navigating movies using gesture to control both forward and backward movement, and to launch movie files linked using visual elements associatively and semantically related to the knowledge domain represented within a movie collection.

Author Keywords
Video, indexing, hypermedia, mnemonics, meta-design

ACM Classification Keywords
H.5.4 Information interfaces and presentation: hypermedia.

INTRODUCTION
The browse searching of digital video files using proprietary software and commercial applications relies on alphanumeric indexing and keyword selection. This is appropriate for ontologies with established taxonomies and structures for maintaining Knowledge Management Systems (KMS) including information contained in movie files. But as movie files become ubiquitous in our everyday lives as a means of conveying information for a range of purposes, the design of computer systems for storage and retrieval employing other forms of visual mnemonics could add efficiencies of speed with ease of accurate access.

The research approach taken utilises the mnemonics contained in the motion-picture images of a movie collection and offers possibilities for non text-based interaction with a KMS. A series of Models were built to explore and test the behaviours of subjects navigating movie files, encountered as full screen motion-picture images, using either arrow keys on the keyboard, or mouse, to effect 4-way control: the playback of the movie – up for forward, down for backward; and launch movie files linked to mnemonics in the movie being viewed - left and right to link.

Linking to left and right is according to a schema, (from the Greek skhema, meaning shape), designed for each Model, that aids in the retrieval of movies in the collection.

Using visual elements associatively and semantically related to the knowledge domain represented within the movie collection, evaluation will compare Models operated by novice and expert groups.

C&C'07, June 13–15, 2007, Washington, DC, USA.
ACM 978-1-59593-712-4/07/0006.

METHODS
The software tool, Mnemovie, was developed by the researcher, in collaboration with a professional multimedia developer. The approach to this initial task has been described by Fischer, Giaccardi et al, using the seeding, evolutionary growth, and reseeding (SER) process model [1] p 492. The practice of building the software tool by Leggett and Hinshaw was guided by the fourth model described by Fischer et al for collaboration paths in software development, where both the domain practitioner and the software professional has some knowledge of the others practice (p 487).

Building on related hypermedia research by Girgensohn et al [2] the Mnemovie tool facilitated the rapid building of Models to create linkages between movie files as a means of realising proof of concept. Following analysis and reflection on the semantic domain of each of six movie collections, navigational schemas were designed before links were created between movie items. The tool enabled the researcher to define parameters within the code to link movie files and iteratively develop and refine the schema for each Model.

Retrieval Schemas
Four of the six schemas were advanced further in preparation for evaluation using a specific movie collection of people talking about their research activity.

1. The segmented Loop schema, where each segment is a compression of each movie item in the collection. Linking thereby has a direct indexical relationship between a Loop segment and the item, and vice versa to return to the Loop. (Figure 1)

2. The Pathway line schema, from A to B, linking to the researcher collection has an indirect connection, where a particular location along the path is the mnemonic for a particular movie item.

3. A development of the Pathway line to the horizontal Grid is applied to a collection of movies shot and linked by the system through the intersecting streets of an inner city block. Linking to the collection of movies about the work of the researchers has an indirect connection with the schema, a particular location in the grid of streets being the mnemonic for a particular movie item.

4. The Clock face schema, dividing by convention the passing of time and indexically, direct links to the proportional durations of a movie; or indirectly to different movies in the researcher collection.

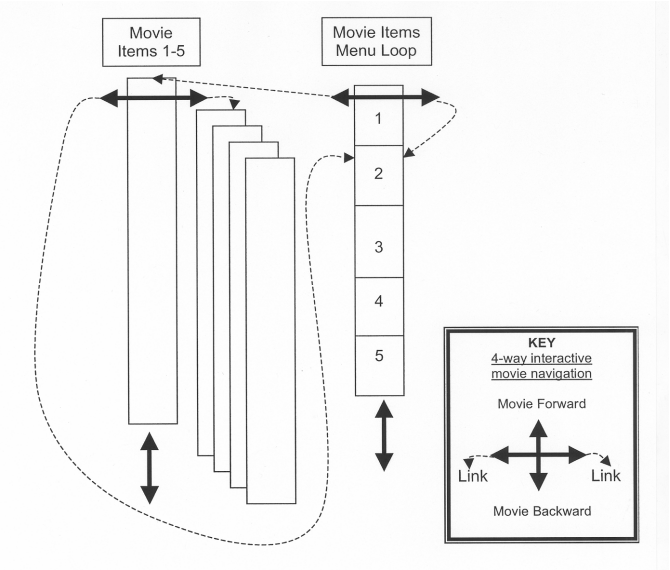

Fig 1. Sample movie collection using Loop schema.

Mnemovie Tool

Arising from an earlier prototype that used Macromedia Director to construct the system framework, conceptual and technical elements used for the PathScape project were extended into the requirements for the current investigation. Specifically this moved away from hard-coded 'content' to a modular and externalised framework, (.mov, .swf, .dcr files), subject to an "external importation routine" incorporating an XML file, "..more extensible to handle growth of later versions." This approach has more recently adopted by other developers whereby a:

"…presentation engine allows content authors to describe … content through associated XML files. Interpretation of those files, content layout, and all … communication is automatically handled by the presentation engine. The content is described external to the application, creating a natural separation from the … interface." [3]

In Table 1, the conceptual data model for building the Mnemovie tools includes the software framework, the presentation engine and the media directory.

Root Directory of Model	Presentation Engine	Media
Mnemovie b3	MNEMOVIE (application) •Mnemovie.dcr •movie_data.xml	MOVIES (video files directory)

Table 1: Mnemovie data model

Media files are prepared using a digital video editing application and saved with consistent resolution and frame size into the directory. The .dcr file compresses

specification data for the Director application and is prepared by the software professional. The file movie_data.xml contains a description of the tags and the layout of the program source code specific to the manipulation of the movie files contained in the adjacent Movies directory. The modular construction of the source code enables the researcher to expand the scale of the instruction set according to the requirements of the interactive Model.

The XML-file structure throughout was based on each <track> having a <movie id> for the forward motion movie and a different <move id> for a reverse motion movie. From each <track> links to other movies could be created. The Beta 1.0 used the following structure (sample):

```
<track id="PD"
 <movie id="PZF" file="movies/PulledZfore.mov" dir="F" ><!--F-->
        <link side="L" start_time="00:00:13" end_time="00:00:14"
              movie_id="11" link_start_time="00:00:00" />
        </movie>
<movie id="PZB" file="movies/PulledZback.mov" dir="B" ><!-- B -->
        <link side="R" start_time="00:00:12" end_time="00:00:13"
              movie_id="R11" link_start_time="00:00:00" />
        </movie>
</track>
```

As experimentation progressed iteratively, the <link_> group of tags operating within the system were increased and thus offered additional linking possibilities for schema design.

Evaluation

Each Model is in the process of predictive and operator evaluation with a movie collection to assess interaction efficiencies between Models, the quality of the experience and the interacting subject's ability to:

- navigate a movie collection using the schema approach;
- retrieve information contained within movie files;
- retain memory traces from the navigational process, such that subsequent interactions with the Models can demonstrate accumulated learning behaviour.

REFERENCES

1. Fischer, G., Giaccardi, E., Eden, H., Sugimoto, M. and Yunwen, Y. Beyond binary choices: Integrating individual and social creativity. *International Journal of Human-Computer Studies*, *63*. 482-512. 2005

2. Girgensohn, A., et al. *Designing Affordances for the Navigation of Detail-on-Demand Hypervideo*. in *ACM Advanced Visual Interfaces*. 2004.

3. Mentor, K. Director and SCORM 1.3 SCORM SCO Presentation Engine (S2PE) *Director Developer Center*, Adobe Inc, 2006.