

# Contextual Navigation in a Multimedia Journal

Stewart Greenhill and Svetha Venkatesh  
Department of Computing, Curtin University of Technology  
stewartg@cs.curtin.edu.au, svetha@cs.curtin.edu.au

## ABSTRACT

This work presents a framework for multimedia journaling, maintaining strong relationships between the document and embedded media. This enables media archives that are robust to changes in software environments, such as changes in web-sharing services, proprietary file formats and enables portability across operating system. We develop a journaling application using an existing multimedia framework, and show the power of the paradigm with specific case studies.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.5.2 [Information Interfaces and Presentation]: User Interfaces

## General Terms

Algorithms, Measurement, Design, Experimentation

## Keywords

multimedia browser, personal media management, photo, video, audio, filter, query, cluster, journal, blog.

## 1. INTRODUCTION

As it becomes easier to capture digital media, we are acquiring larger and more diverse personal media collections. Portable devices like camera phones mean that it is now rare that significant events in our lives cannot be recorded using photos or video. This offers new possibilities for people who record their experiences via a diary or journal. These traditional forms of self expression are finding new outlets on-line in blog culture.

A blog is a just one type of *multimedia journal*. This medium consists of short text articles which may include embedded images, videos, or audio clips. The article may be about the embedded media, or the media may be included to illustrate a point, or for aesthetics. Often the articles are associated with dates and can be viewed sequentially.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.  
Copyright 2008 ACM 978-1-60558-303-7/08/10 ...\$5.00.

Several options exist for multimedia journaling. With the advent of mass-market word-processors like MacWrite and Microsoft Word in the mid 1980s it became possible to combine text and images in a convenient graphical interface. Extending this to arbitrary media types requires a framework that allows the editor to delegate part of its interface, display and storage to a separate software component. This was solved most successfully by Microsoft, with the introduction of OLE in 1990. Other component frameworks such as OpenDoc exist, but have failed to be widely adopted.

With a modern word-processor, it is possible to embed a variety of different media types into a document. The way this is handled depends on the media type. Images are generally handled natively by the application. Under Windows, streaming media (video and audio) are usually handled by one of two components: the MCI player or the Windows Media Player. Under Mac OS, streaming media are handled by the Quicktime player, which is also installable under Windows. Each player supports different combinations of media file format and codec.

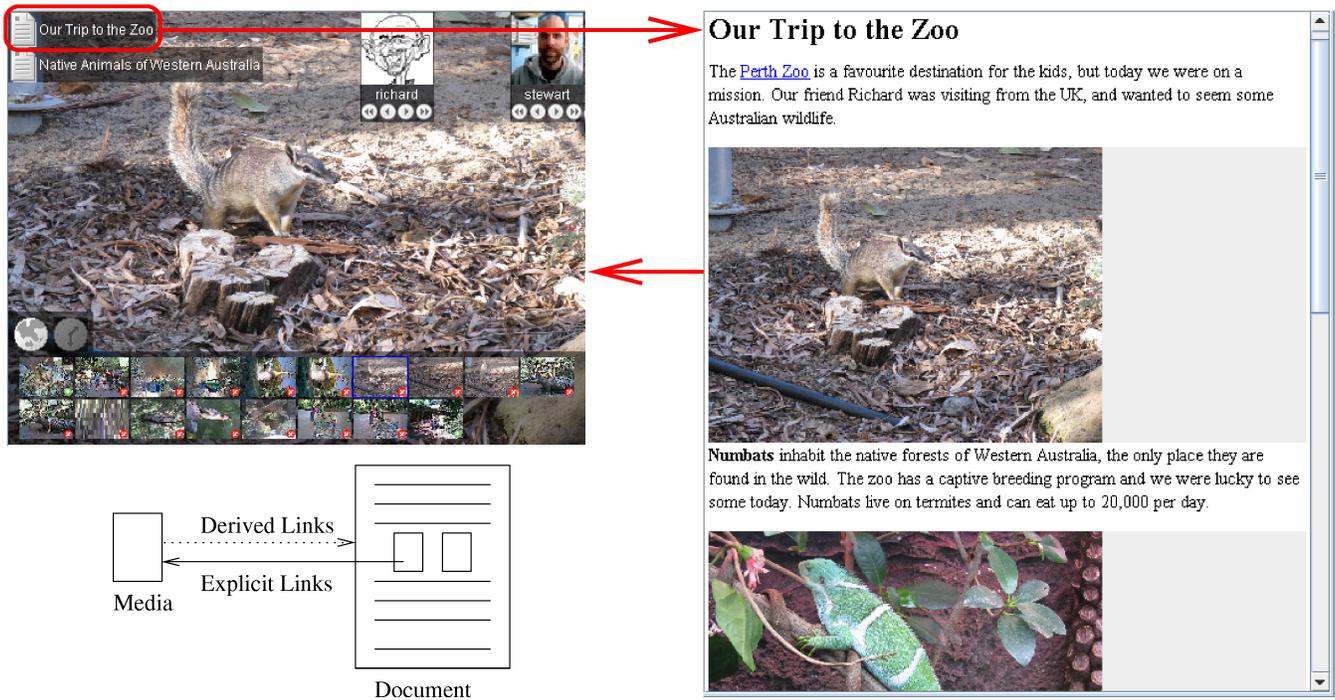
A weakness with this approach is that different operating environments may use different players to render media, and even if the required player is available, the necessary media codecs may not be installed. This makes the resultant document *fragile* because even when its browser is installed, it may not be viewable on some machines.

In niche areas, journals can be combined with domain-specific record keeping or browsing. For example, sports journals allow the user to keep logs of training duration, and generate statistics over time. Garden journals organise time around yearly cycles, allowing the user to see what was they did at a particular time over many years.

Blogs are journals that are hosted on-line and read using a web browser. Blogs are usually written using a safe sub-set of HTML. Images are supported natively via the IMG tag, but streaming media requires external plugins to be used, such as Apple's Quicktime or Adobe's FLASH plugins. The Quicktime plugin only renders media, whereas FLASH also supports downloadable code which can be used to dynamically interact with the user. This has made FLASH a popular platform for web-based media players.

To embed a video into a blog post, one normally copies an EMBED code into the HTML source for the blog article. This identifies the plugin and URL source required for the browser to render the media, and may be generated automatically by media sharing sites like YouTube.

In both scenarios, only a weak relationship exists between the document and the embedded media. For both word-



**Figure 1: Contextual links for media objects point back to documents that contain them. Icons in the top left of the image link to two separate documents containing the image.**

processed journals and blogs, media may be embedded using components that are essentially opaque to the document browser. That is, the article may not “know” much about its embedded components, and the components do not know about the articles that embed them. This prevents us from doing some potentially useful things:

- For a media object, navigate to documents that include or reference it.
- For a document, find other documents that refer to similar media objects.
- Propagate attributes such as *location* and *time* from media objects to the documents that reference them.
- Associate terms in document text with referenced media objects to improve media findability.

A goal of this work is to develop a form of personal multimedia archive that is *robust to long term changes in software environments*. This means it should:

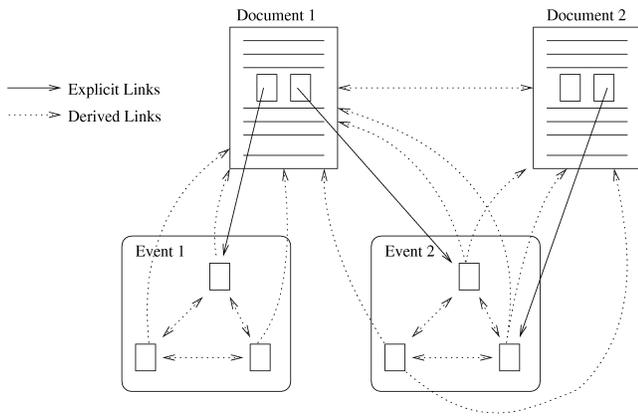
- Avoid the use of proprietary or closed file formats. Preferably, semantic data should be human-readable.
- Avoid reliance on external services that cannot be guaranteed in the future (eg. web-based sharing services). Where possible we can publish to these sites, but the original material *must be accessible and browsable offline*.
- Be self-contained and archivable as a whole.
- Be portable across major operating systems.

To explore some of these possibilities, we developed a simple journaling application using the Geode multimedia framework [2]. Geode manages personal media collections, allowing photos, video, and images to be browsed and navigated. It also handles streams of activity data originating from social applications like Twitter and Facebook. It uses spatial and temporal clustering to help organise and group media. However, it has no ability to deal with text or compound objects as are found in blogs and journals.

We implemented a document handler, using HTML as the basic model. We also implemented a simple blog-like editor, allowing articles to be entered as formatted text, or as raw HTML. When documents are saved, the system analyses them for references, and incorporates the cross referencing between media and documents into the browser.

The existing Geode contextual link display was expanded to add links back from media to the containing documents. This is shown in Figure 1. In the top left corner of the image display, the browser shows links to documents containing that image. These can be clicked to navigate to that document. In addition Geode shows links to other media at the same place (bottom), and to the creator of the image, and any co-present people (top right).

Optionally, links can be propagated to other media in the same event or place cluster. This means that if two articles refer to different images of the same event, all media in that cluster will link to both articles, and the articles will link to each other. This is shown in Figure 2. Links between media objects based on event and place are provided by the original Geode player. This work adds documents, the links back from media objects to documents, and the links between documents that refer to the same objects.



**Figure 2: Links between media objects are established based on cluster (event or place) membership and document containment.**

The current work differs from other offerings (see Section 2) in several ways. Firstly, it uses event and place clusters to enable *bottom-up navigation* of media collections – that is, the discovery of events, places, and their related media starting from individual media objects. Secondly, it incorporates text documents in the model. Thirdly, it derives relationships between documents based on the relationship between the media they refer to. This is particularly useful since Geode is a multi-user media system and this increases the discoverability of other users’ material. For example, if two users write articles using different media from an event, the articles immediately become visible to each other because the system knows they relate to the same event.

## 2. RELATED WORK

This work is an application of the Geode personal media manager [2]. Geode differs from conventional media browsers in that it uses context information collected via the users cell phone to improve the organisation and navigation of media. Media are clustered over location and creation time to derive *places* and *events*, which are used to group media during browsing. Geode supports photos, audio, video and activity streams (eg. from social applications like Twitter).

The Geode player includes contextual linking which can be used to navigate from a media object to related objects. Figure 1 shows examples of the link visualisation, which is activated when we move the mouse over the image. In the top right is the avatar image for the image creator, with navigation controls to access the next/previous item, or event. Other avatars indicate people who were present when the media was captured (based on GPS location or bluetooth co-presence). Their media streams can also be accessed via navigation controls. The film-strip at the bottom shows thumbnails for other events that occurred at the same place. The events are represented as clusters that can be opened to show the individual items.

Other media browsers use clustering in different ways. The Calendar Browser [1] implements hierarchical clustering, allowing users to progressively “drill down” into sub-events of finer resolution via a tree view or calendar view. PhotoCompas [4] extends this approach to include hierar-

chical location clusters which can be browsed on a map, or via automatically generated place names. The PhotoTOC [5] browser shows the complete set of images in a “detail” pane, and a set of clusters in an “overview” pane. Clicking on the overview scrolls the detail pane to the relevant image. MediAssist [3] uses location and content-based ranking to find images similar to a selected image.

## 3. IMPLEMENTATION

Geode is implemented in Java, and uses the Java Media Framework (JMF) to process and display streaming media (audio and video). We use the FOBS plugins (fobs.sf.net) which implement a JMF interface to the ffmpeg platform (ffmpeg.mplayerhq.hu), an open-source media framework that supports a diverse range of codecs, formats and media transport across many platforms (Windows, Macintosh, Unix). This makes Geode independent of the native codecs that may or may not be available on a particular platform.

Our document editor uses Swing’s JEditorPane and HTML-LEditorKit, which currently provides support for HTML 3.2, but is migrating toward 4.0. The standard implementation includes support for the OBJECT tag, which can be used to instantiate an arbitrary Swing component by specifying the class name, and providing parameters which are stored and retrieved using the normal Java-Beans getter/setter methods. Thus, we can use the following notation within the HTML file to embed media resources:

```
<object classid="geode.Player" width=600 height=300
mediaset="stewart:688;stewart:689">
```

Media objects have a unique name based on user and numeric identifiers. A separate data-base in the Geode repository defines the attributes of each object (media type, metadata, file or URL, etc). When documents are changed, we traverse their elements to find all referenced media. These links are propagated according to detected events, places and user preferences.

## 4. RESULTS AND USAGE

Figure 3 shows the user-interface. A spatial filter (top right) has been used to select media from a particular place, with the results presented in a scrollable thumb-nail display below (the “time-strip”, bottom right). Available filters include time, place, user and media type. These are processed from left to right, progressively refining the selection at each step. For example, Figure 4 shows the Time filter (top), used to select media from March 30, 2008. The Place filter (bottom) shows media captured on that day, and allows the selection to be further refined; selected places are shown in green, and unselected are in red.

The left hand side of the display shows the editor being used to compose an article about a visit to a local restaurant. We allow photos, videos, and audio to be dragged from the time-strip into the editor. This inserts an instance of the standard Geode player into the document which can be used to view and navigate the linked media. Videos can be played inside the document, and contextual links can be followed from the images to other documents or media. Figure 1 shows an example of the contextual links which are activated when the mouse passes over the media object. Whenever documents are saved from the editor, our system tracks the references between documents and the embedded media.



Figure 3: Browser layout showing Place filter (top right), time-strip (bottom right), and document editor (left) including video (above) and photo (below).

We tested the system using data collected over 6 months from seven participants [2]. Devices included phones (Nokia 6110, N95), digital still cameras, video cameras, and audio recorders. A variety of encodings were used, including MP3, WAV, AAC, AC3, MP4 (MOV), H264, DIVX, 3GPP, MJPEG (AVI), JPEG and MPEG-2. All formats were handled adequately by the JMF/FOBS player.

## 5. CONCLUSION AND FUTURE WORK

This work explores some of the possibilities available when media and text are incorporated into a common browsing platform. By understanding the links between documents and embedded media, the system facilitates discovery of relationships between objects through navigation, and through the propagation of relationships over deduced places and events.

Unlike blogs, the system does not depend on external services to host and browse embedded media. It lacks some of the dynamic facilities of blogs, such as comments, but has the advantage that it is self-contained and can be edited and browsed off-line. This is particularly important for large media collections, as bandwidth and storage can be expensive in on-line browsing. One option for the future is to incorporate a blog-publishing facility using APIs such as the Blogger and MetaWebLog APIs. This would allow content to be viewed on-line, at the expense of some of the navigation options present in the current system.

## 6. REFERENCES

- [1] Adrian Graham, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Time as essence for photo browsing through personal digital libraries. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 326–335, New York, NY, USA, 2002. ACM.
- [2] S Greenhill and S Venkatesh. Geode: A framework for social and context-driven browsing for personal multimedia, 2008. Submitted for publication.

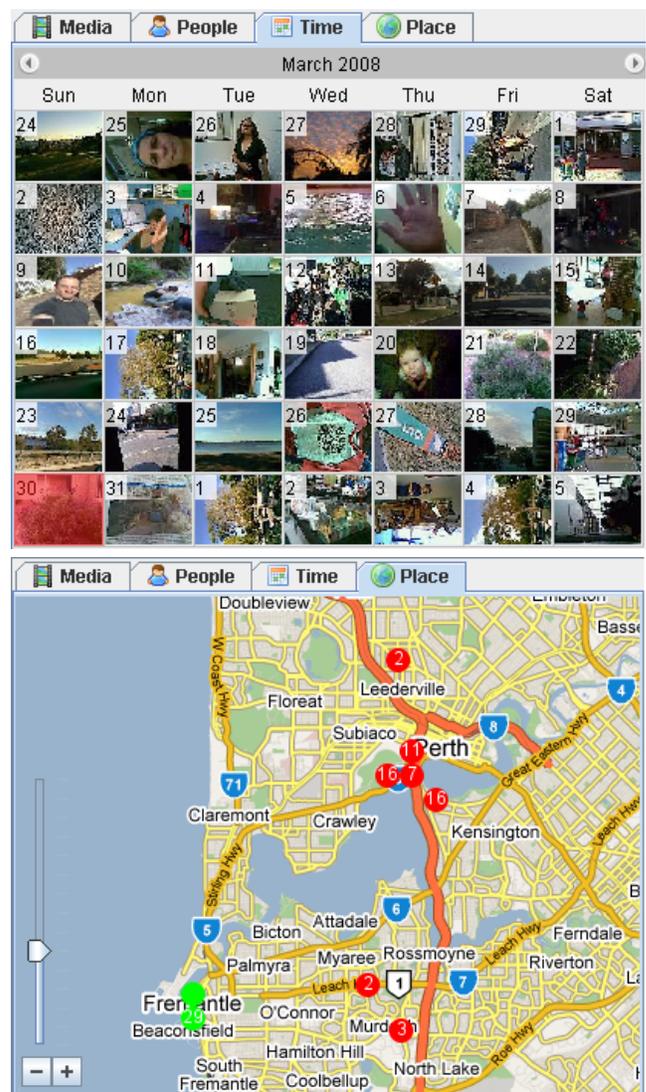


Figure 4: Filters allow selection of media based on Time (top) and Place (bottom).

- [3] N O'Hare G Jones, C Gurrin, and A F Smeaton. Combination of content analysis and context features for digital photograph retrieval. In *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, 2005.
- [4] Mor Naaman, Yee Jiun Song, Andreas Paepcke, and Hector Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 53–62, New York, NY, USA, 2004. ACM.
- [5] John C. Platt, Mary Czerwinski, and Brent A. Field. Photoc: Automatic clustering for browsing personal photographs, 2002.