Continuous Metadata

Kevin Page, Ben Juby, Richard Beales and David De Roure

Intelligence, Agents, Multimedia
Department of Electronics and Computer Science
University of Southampton, UK

{krp,bpj00r,rmb00r,dder}@ecs.soton.ac.uk

Metadata for multimedia content can describe the detail of content in order to facilitate processing, for example identifying events along the time axis in temporal media, as well as carrying descriptive information for the overall resource. In both cases the metadata is essentially static and may be associated with, or embedded in, the multimedia content; it may also convey low level signalling data or higher level knowledge such as annotation by users. This paper discusses the case for working with semantically rich metadata as one or more distinct and continuous live flows, managed, delivered and processed separately from the content. It discusses a prototype system designed to explore the use of continuous metadata in videoconferencing and its extension to smart meeting spaces.

Introduction

The most familiar form of metadata for multimedia content is perhaps the information that describes a particular multimedia object; e.g. catalogue information for archival purposes, or information relating to the production process (formats, equipment). Metadata can also describe elements within the multimedia content, and this is sometimes embedded within the multimedia representation. In this paper we consider metadata that is by necessity live and associated with multiple multimedia flows, and hence does not fit these traditional views. Although this can be regarded as an encoding of signals relating to the multimedia production process, we are also interested in capturing higher level knowledge relating to the process and content – this can be regarded as multimedia annotation.

Key notions of metadata are introduced in the next section. Section 3 describes our view of continuous metadata. The videoconferencing scenario, in which the metadata is live and two-way, is considered in section 4 with our prototype implementation. In section 5 we extend this to consider the richer metadata associated with linking smart spaces.

What is Metadata?

Metadata is ancillary data about other data, and as such is data itself. In general it refers to any data used to aid the identification, description and location of some other electronic resource. Many forms of metadata exist; some are very simple while others are quite complex and richly featured.

The definition of what metadata is and what it can represent is very broad, and while the creation of metadata is no simple task, it is the interchange of metadata that is the focus of most research. The whole point of metadata is to aid the understanding of other data, so there must be a way to decode the metadata into useful information or it becomes as useless as the data it is augmenting. Without common structures and standards for metadata there can be no interchange and translation between systems; without consistent interpretation metadata has no value. Hence the information conveyed by metadata is part of a shared, simplified view of the domain (a conceptualisation), for example a shared vocabulary. The specification of this conceptualisation is called an *ontology*.

Semantic Web

While much of the information on the Web is designed for human consumption, there is an increasing need for machine-to-machine communication. To achieve this, the information must be machine-understandable. This has led to the notion of the *semantic web* in which metadata describes resources in a machine-processable syntax, and the schema can similarly be specified and shared.

The Resource Description Framework (RDF)[1] is an infrastructure that enables the encoding, exchange, and reuse of structured metadata on the WWW. RDF does not prescribe semantics for each particular resource description community, instead it provides the ability to define new metadata elements as needed using an XML syntax. Its data model defines a *resource* as any object uniquely identified by a URI (Uniform Resource Identifier), and resources have *properties* which express the relationships of *values* associated with that resource. The values can be either atomic (strings, numbers etc.) or other resources (which may have their own properties).

Collections of properties about a particular resource form a *description*; collections of properties used to describe resources within a particular resource

[1] RDF, XML and URIs are World Wide Web Consortium (W3C) recommendations; the reader is referred to the W3C web site www.w3.org

description community form a *vocabulary*. An RDF vocabulary can be defined by an RDF Schema (RDFS), which has some predefined concepts and properties.

RDFS is more expressive than RDF but lacks features that the used by the knowledge engineering community . This has led to the activity in DAML+OIL [vanHarmelan00], which extends RDFS and supports inference, providing a more powerful query mechanism.

Metadata and Multimedia Content

A metadata schema for temporal multimedia could be considered of greater importance than one for the (mostly text-based) WWW, since it is even more difficult to generate metadata for media such as audio and video without commonly recognised descriptions. In the multimedia realm metadata has many attractive potential uses: semantic searching, indexing, retrieval and filtering of multimedia databases; image understanding for intelligent vision and surveillance; and conversion between media (speech to text etc.).

There are four fundamental ways of describing multimedia data that could form useful metadata [Nack99]:

- **Medium-based** descriptions are of the medium in which the data is expressed, such as the sampling rate or the camera's focal depth.
- **Perceptual** description breaks the media into perceptual objects such as colour, texture or sound.
- **Physical** descriptions are of features that do not correspond to human perception, and can be easily derived from raw multimedia data. Examples include 'level' and 'frequency' (compared to perceptual 'loudness' and 'pitch').
- **Transcriptive** descriptions represent a reconstruction of the real worlds structure as captured by the data. For example, a musical score can represent audio data.

For any particular segment of multimedia data, several of these description classes can be used to give different views of the same data. Any multimedia metadata standard must not only accommodate these independent viewpoints, but also make them complementary rather than mutually exclusive. There may also be a need for an architectural description to formalise the structure of the other description classes, the data they represent, and relationships between them.

The Moving Pictures Expert Group (MPEG) of the International Organisation for Standardisation has produced several standards for the coded representation of temporal audio and video, with varying levels of metadata support:

- The MPEG-1 standard for storage and retrieval includes a mechanism for multiplexing a data stream (which could be metadata) into the MPEG-1 stream, but does not prescribe how to format this data (which has led to proprietary, incompatible, implementations).
- MPEG-2, the digital television standard, is an extension to MPEG-1 that utilises a higher resolution. It offers limited additional metadata support through a structured information block in its header, which can be used to encode copyright and access information.
- MPEG-4 is a standard for the production, distribution and content access of multimedia, and is designed to be applicable to a wider range of fields than the earlier standards. It still deals with streams, but subdivides audio-visual content into objects. Metadata can be attached to these objects, but again there is no standard structure or format.
- The Multimedia Content Description Interface, or MPEG-7, is not a standard for transmitting or storing multimedia data. Instead, it aims to standardise a core set of quantitative measures of audio-visual features (*Descriptors*) and structures of descriptor relationships (*Description Schemes*). MPEG-7 will also introduce the *Description Definition Language* (DDL) to specify new Description Schemes, which has the same aims for multimedia metadata as RDF does for the WWW. RDF is not suitable for inter-operation of multimedia metadata since it has no linking mechanisms to spatio-temporal sections of data and limited data typing. However, the MPEG-7 DDL will use the XML Schema language as its basis with a view to future interaction with non-MPEG-7 metadata.

Since the earlier MPEG standards have mechanisms to include metadata, but no standard metadata format, it is envisioned that they will use MPEG-7 to improve their content description capabilities, although this does not preclude using MPEG-7 with other, non-MPEG encoded, media.

A conceptual framework for continuous metadata

While stored temporal multimedia must be streamed because of its size, the associated metadata would normally form a much smaller quantity of data. For shorter volumes of media it could be argued that the metadata can be pre-loaded into the client by downloading one file before presentation of the media begins. This is analogous to traditional handling of metadata within the broadcast production environment: carried on a separate floppy identified by the SMPTE Unique Material Identifier (UMID) of the referenced essence material, or more often simply recorded on a paper form or label which is bundled with the essence media.

For greater lengths of media it might be the case that the amount of metadata has become large enough to warrant streaming, but it is for *live* transmission of media that streamed metadata becomes essential. In this scenario the metadata will be generated on the fly at the same time as the media, and must be transmitted in parallel with it. The metadata might be produced during the live production process, but it could also be the result of live processing or annotation when stored media is broadcast.

Although the metadata in our conceptual framework is streamed, it may be better to think of it as continuous metadata. The word ‘stream’ has become closely associated with real-time audio and video, and often (incorrectly) implies a non-stop flow of relatively high bandwidth data. Continuous metadata need not be high volume, and there may be significant lulls between bursts of data (although the transporting channel persists); however the transmission timing of the metadata does have significance, and it will often be augmenting continuous, streamed, media data.

In this framework it is neither the type nor content of the metadata that is important, rather that it is some kind of metadata and that it is handled in a continuous manner. The classification and exchange of metadata can already be described by standards such as RDF and MPEG-7 there is no reason why the metadata carried by the framework could not be encoded using these standards.

Metadata and Mediadata

In this framework we consider mediadata (*cf.* essence), and metadata, where the continuous metadata flow carries additional data about a corresponding temporal multimedia, or mediadata, flow. The mediadata will normally be a multimedia stream, such as audio or video, and can be characterised as a continually evolving flow of data - one frame of a video generally has a direct relationship with the previous. Metadata, on the other hand, will be split into discrete chunks of information within the continuous metadata flow.

In general the metadata is transported through the framework separately from the mediadata, rather than multiplexed within it. This allows for a much more flexible framework of distributed processing and presentation. The arguments against embedding metadata are well-rehearsed in the hypermedia research community [Davis95]. The distinction between mediadata flow and metadata flow may in reality be blurred, if metadata essential to the reproduction of the mediadata is interlaced into the mediadata flow.

It can be argued that what may be metadata in one case should be mediadata in another, and in many ways this is true. While a flow of MIDI information would be metadata for a raw audio mediadata flow in one case, in another there may be no audio stream and the MIDI might form a mediadata flow augmented by other metadata. We should note that just because a metadata flow may develop a derivative metadata flow ‘about’ it, this does not make the derivative flow ‘meta-meta’-data, nor does it imply the original metadata should become a mediadata flow. The derivate flow merely becomes another metadata flow based on the original mediadata, albeit one with a more complex relationship with other metadata.

We will first consider how the framework should handle point-to-point media and metadata flows by introducing the various elements that make up a simple version of the framework.

Sources and Flows

There must be a point at which the mediadata enters the framework, and we refer to this point as the mediadata source. For simplicity, we initially presume that each mediadata flow is derived from a single source; with a more complex implementation there is no reason why a mediadata flow cannot enter the framework in a distributed manner. The method by which the content of the mediadata is transported through the framework should be suitable for that data type, e.g. RTP for audio or video.

The metadata source is the point at which a continuous metadata flow enters the framework. This may be at the same point as the mediadata source or it may be distributed at a different point: for a live news feed a provider might construct a metadata flow of relevant links at the same broadcast point as the mediadata; while viewing a video of a pre-recorded lecture a user may wish to receive metadata annotations from a source other than that of the original lecture.

Presentation

We will refer to the point at which a user views and uses a combination of media and metadata flows as a presentation point. (This is a deliberate avoidance of client / server terminology since it will become apparent that within this framework a "client" to one "server" can be a "server" to another.) There is no reason why a presentation point should only be the convergence of a single mediadata and metadata flow; it should pull together and synchronise as many metadata flows as the user requests. Since a mediadata flow is the timer against which other flows are synchronised, any metadata flow used at a presentation point must have been derived from that mediadata at some point. Multiple presentation points for multiple mediadata flows can exist on one machine, for one user, at the same time, but they should be dealt with as separate entities within the framework.

The presentation mechanism also starts to place requirements on the information the framework must encode in the metadata flow (in addition to the metadata itself):

- The metadata must have an identifying code (for example a UMID). Not only is this code needed to deal with packets from a particular flow in a consistent manner, it must also allow identification of the mediadata flow with which the metadata must be synchronised.
- To synchronise the media and metadata, each packet of metadata must have a pair of validity timestamps bounding when the metadata is true in

relation to the mediadata and the timing information embedded within it.

- For user presentation there should be another pair of timestamps bounding a extension around the valid time, during which it is is suggested that the metadata is displayed (although this could be overridden by user presentation preferences).

To present the metadata in a suitable manner there must be a code to describe the content type of the payload the metadata packet is carrying. Although the content code needs to be standardised within the framework, the format of the content itself need not be.

Filters

While the ability to select different metadata sources for a particular mediadata flow is useful, the real flexibility of the framework is through the introduction of processing nodes between the metadata source and the presentation point.

These filter nodes are distributed throughout the framework, taking one metadata flow as their input, modifying the metadata in some way, and then outputting a new metadata flow. The output of one node can be linked to the input of another so that the end result of metadata processing between source and presentation is formed from a series of simpler, more specialised, processing steps within the framework, thus extending the concept of the Microcosm filter chains [Davis92]. Each filter is expected to perform a relatively specific form of processing, and by doing so it can be located at a point where the resources it may require are best available. As a result of this, individual metadata flows within the framework should carry specific types of metadata payloads to allow maximum flexibility between filters. A filter should not have to demultiplex a metadata flow so it can select only relevant data. Separation of the metadata from the mediadata flow means that many filter nodes will not need to receive the original mediadata flow, conserving network resources.

The end effect of a filter should be to either add or subtract metadata from that which a user receives at a presentation point. To add data, the filter output flow can be synchronised with the original metadata flow at the presentation point. To remove, or truly filter, the metadata, the filter output should be the only flow accepted at the presentation point: the original flow must be dropped. To accommodate this, metadata flow identities must incorporate the notion of derivatives, such that the history of a flow can be traced back through filters to the original metadata source identity. Suggested presentation relationships (flow x must be presented with flow y, but should not be presented with flow z) also need to be encoded in the metadata flow.

Control

Even with buffering at the presentation point, network congestion could delay metadata flows which need to be hard synchronised with others; in this situation the stalled flow can either be dropped, or the remaining flows must be paused while waiting for resumption. Other pauses or temporal movements may be user controlled, since the metadata flows must be paused once the mediadata has been stopped. To provide such functionality within the framework there must be control channels between the presentation point and the various filters and sources that feed it.

There are two general approaches to propagating the control messages: Send the control message from the presentation point to the media and metadata sources, then propagate the message to the next filter in the chain; or send the control message to all the filters one hop ‘upstream’ of the presentation point, then propagate the message up through the filter chains to the sources.

Prototypes

Two earlier prototype systems have been developed to address continuous metadata applied to hypermedia:

- For streaming stored media, RTSP has been combined with the Fundamental Open Hypermedia Model (FOHM). This system involves prefetching metadata.
- For live (synchronised) metadata, the ‘HyStream’ application uses a software agent framework and simple ontology [Cruickshank01].

Here we focus on a videoconferencing scenario. The main difference between this scenario and previous scenarios is that mediadata (i.e. audio and video) cannot buffered for any significant time as it would introduce unacceptably high latency in the conference. In this scenario metadata needs to be generated as quickly as possible on the fly. Possible sources of metadata in this context include annotations, slides, meeting minutes and shared whiteboard activity. These can be captured using laptop PCs, PDAs and document cameras. Since the mediadata cannot be buffered, the captured metadata must be sent as soon as it is available. This means that the metadata could potentially lag behind the video and audio due to the overhead of it being captured and processed. This is unavoidable, and steps must be taken to ensure that the metadata processing time is as short as possible.

H.323 is the dominant standard in videoconferencing, and through T.120 data conferencing can support continuous metadata delivery. The T.120 series standard is part of the H.323 standard and is commonly used to add functionality to videoconferences such as shared whiteboard, text chat and file transfer. T.120 has a number of features that make it suitable as a continuous metadata transport mechanism. These features include:

- Real time, reliable delivery
- Independence from the conference media streams
- Support for multiple data channels
- Multicast support

Additionally, T.120 allows one of four priority levels to be assigned to data. This allows timing critical metadata to be assigned the highest priority level, while other operations such as file transfer can be assigned lower priorities.

The function of metadata processing nodes can be carried out by specialised conferencing terminals. Such a terminal can receive a T.120 metadata stream, process it and retransmit the transformed stream. Feeding the output of one filter node terminal into another filter terminal can create a filter chain. Since all T.120 data can pass through a Multipoint Control Unit (MCU), some additional filtering could potentially take place in MCUs.

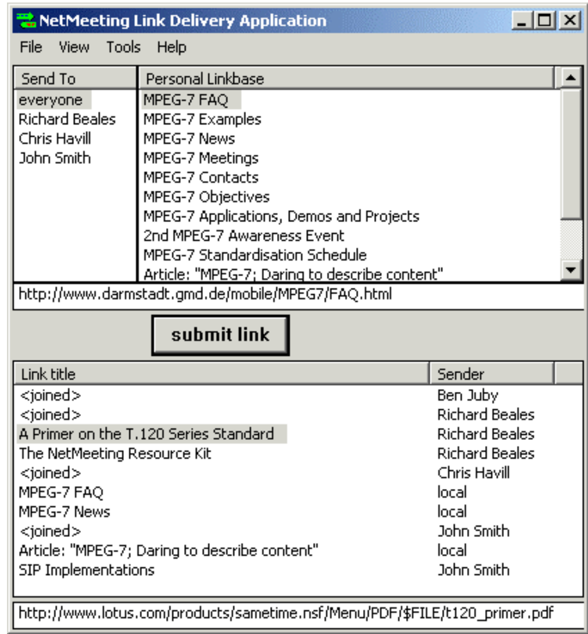
One weakness of T.120 is that there is no built in mechanism for synchronisation of data with audio and video streams. This is not a problem in a videoconference, as the mediadata and metadata is always displayed as soon as available. For presentation, a suggested length of time the metadata should be displayed for is more appropriate than a pair of timestamps.

Implementation

We have implemented a simple prototype system to demonstrate real time metadata delivery in an H.323 videoconference. The chosen videoconferencing client is Microsoft NetMeeting 3.01 and the metadata is handled by a plug-in written using a T.120 API available free as part of Microsoft's NetMeeting SDK.

The prototype application uses hypermedia links as an example of live conference metadata. Before the conference, each participant can prepare a personal list of links that will be relevant to conference and load this list into the plug-in. These links could be to documents and slides that are intended to be discussed during the conference. If during the course of the conference a participant decides that one of their links is relevant to the current topic under discussion, they can share the link with the other participants by selecting the link and clicking the send button. This automatically sends the link in real time to the other participants over a T.120 connection. The recipients will have a new link appear in their plug-in window. Clicking on the received link will automatically launch a Web browser displaying the target document.

A participant has the option of multicasting a link to all members of the conference or unicasting the link to a single member by selecting that member from a list of conference participants. The user interface to the plug-in is shown in the figure below.



Smart Conferencing Smart Meeting Spaces

Most traditional meeting rooms will be equipped with a range of objects and appliances – projector, flip-chart, possibly a basic video conferencing facility, and so on – intended to provide passive assistance to the participants of a generic meeting.

Smart meeting spaces are differentiated by their ability to support each specific meeting actively, adopting the role of invisible secretary, consultant and technician. They vary in the technologies used to achieve their smartness, but typically support the processing of visual and auditory cues (for example gesture or speech recognition), some facility for identifying the meeting context and goals and pre-empting the participants' requirements, and the ability to monitor and influence environmental and spatial factors. This is achieved with the support of a significant amount of networked embedded computing.

We are investigating the application to smart meeting spaces of 'Knowledgeable Devices' – a phrase we use to describe small, networked, embedded devices characterised by their having a local fact base and reasoning capability and ability to 'talk' RDF.

As an initial test-bed we have developed i-See!, a simple business-card exchange application, which automates the exchange of contact details between participants in a meeting. Implemented using a combination of Java and Dynamic HTML, our application currently runs on standard Windows laptops equipped with Lucent wireless Ethernet cards. Although trivial, i-See! embodies many of our key principles: reasoning ability, and an RDF-based, ontology-driven approach to knowledge exchange, dynamic service discovery and ad-hoc wireless connectivity. The possibility of networking our knowledgeable devices with a radical protocol built 'knowledge-aware' from the ground up, in place of TCP/IP, is under consideration.

While collaboration between devices is currently limited to swapping names and addresses, it its felt that, with a sufficiently enriched ontology and array of physical contextual inputs, much more imaginative multistep reasoning will be possible. A short term aim is to enable the devices to assemble snapshots of the meetings in which they are involved which can then be used to recognise another instance of the same meeting, and present the user with a choice of relevant resources, such as previous minutes. By negotiating amongst themselves, the devices will also be able to configure the room as they regard appropriate.

The nature of the ad-hoc, wireless connectivity employed is that communication from any device is limited to a range of several tens of metres. In a meeting where videoconferencing is used to link remotely situated parties, this could potentially lead to isolated pools of connectivity, each unaware of participants in different locations. However the provision of a metadata stream as a component of a videoconference session presents us with an opportunity to expand this collaboration beyond a single physical room. As the mediadata, or essence, stream provides an audio-visual link between meeting spaces, so the accompanying metadata will form a bridge between smart meeting spaces, effectively creating a single smart conference space. Many physical resources become as easy to manipulate remotely as the NetMeeting whiteboard, and the ad-hoc, person to person communication which would take place within a single meeting room can now proceed transparently throughout the conference space.

Intriguingly, one component of the smart meeting space may be some kind of short-term electronic memory, offering the possibility of 'spill-over' between current and previous meetings. So, for example, 'ghost' business cards may be passed between people who have never met, in either physical or virtual space and have indeed never stood in the same room. Similarly, if both metadata and essence are archived, then a degree of exchange will also be facilitated between participants of the meeting and those viewing a recording after the event.

The development of near-field radio based Personal Area Networks offers a further exciting possibility. This technique, whereby exchanges of personal or intimate information occur only when people touch, forming momentary 'inter-PANs', may be extended across the conference. A pair of transceivers, identical to those worn on the body, but built into a suitably tactile object in each meeting room function as remote body parts – while each is touched by a person a tunnel will exist, linking their PANs as though they were standing next to each other.

Conclusions and future work

We have made a case for metadata to be regarded as distinct flows of commonly understood data rather than descriptive information embedded in a multimedia content representation. We have illustrated this with a scenario based on videoconferencing and interconnecting smart spaces.

Our early prototypes focused on transporting hypermedia anchors and link information, a well understood form of metadata. Related work in the hypertext research community includes [Grønbæk00] and [Smith00]. In particular, we are interested in adapting multimedia presentations on the fly; an interesting example can be found in [Pan00].

We are currently establishing ontologies for hypermedia linking and other forms metadata, using towards RDF as a common metadata format. In particular we are developing ontologies for the 'knowledgeable devices' discussed in the previous section, considering the combination of pervasive computing and live multimedia.

We are also developing a tool for multimedia annotation according to an ontology, so that someone monitoring a live stream is able to mark it up with ease; this partial mark-up can then provide the navigational structures to facilitate further annotation.

As well as the challenges of creating these new ontologies, we anticipate that this work will raise issues of the 'knowledge lifecycle', e.g. where is the metadata created, how is it maintained and how long does it persist?

Bibliography

[Cruickshank01] Don Cruickshank, Luc Moreau, David De Roure. Architectural Design of a Multi-Agent System for Handling Metadata Streams. 5th International Conference on Autonomous Agents, May 2001 (to appear).

[Davis95] Hugh C. Davis. To embed or not to embed. Communications of the ACM, 38(6):108-109, August 1995.

[Davis92] Hugh C. Davis, Wendy Hall, Ian Heath, Gary J. Hill, and Robert J. Wilkins. Towards an integrated information environment with open hypermedia systems. In Proceedings, ACM European Conference on Hypertext ECHT'92, pages 181-190. ACM SIGLINK/SIGWEB, November 1992.

[Grønbæk00] Kaj Grønbæk, Lennert Sloth, and Niels Olof Bouvin. Open hypermedia as user controlled meta data for the web. In Proceedings, The Ninth International World Wide Web Conference, pages 554-566.

[Nack99] Frank Nack and Adam T. Lindsay. Everything you wanted to know about MPEG-7: Part 1. IEEE Multimedia, 6(3):65-77, September 1999.

[Pan00] Pengkai Pan and Glorianna Davenport. I-Views: A community-oriented system for sharing streaming video on the internet. In Proceedigs, The Ninth International World Wide Web Conference, pages 567-582.

[Smith00] Jason W. Smith, David Stotts, and Sang-Uok Kum. An orthogonal taxonomy for hyperlink anchor generation in video streams using ovaltine. In Hypertext 2000, pages 11-18.

[vanHarmelan00] Frank van Harmelan and Ian Horrocks. Reference Description of the DAML+OIL Markup Language. Available from www.daml.org, December 2000.