The Pipeline of Enrichment: Supporting Link Creation for Continuous Media

Richard Beales, Don Cruickshank, David De Roure, Nick Gibbins,
Ben Juby, Danius Michaelides, Kevin R. Page

Intelligence, Agents, Multimedia,
Department of Electronics & Computer Science
University of Southampton, SO17 1BJ, UK

Abstract The application of open hypermedia to temporal media has previously been explored with respect to the link service, in particular link delivery and generic linking. This paper is based on the notion of *continuous metadata*, in which we use metadata in a temporally significant manner to capture and convey the information required to support linking. With a focus on link creation and *live* processing, our approach enriches hypermedia content with additional metadata at a number of points between capture and delivery. We illustrate this approach with a tool which assists metadata capture by annotation of continuous media according to a simple ontology.

1 Introduction

Previously we have investigated the application of open hypermedia to temporal media [7]. This paper extends this work in three ways: we address link creation rather than link resolution and delivery; we generalise link or anchor/endpoint streams to be instances of *continuous metadata* and we consider this metadata throughout the system; we support live processing rather than assuming we are working with stored media.

The scenario which motivates this work involves capturing an event, such as a seminar or meeting, and making this available retrospectively as a hypermedia resource. We will focus on video recordings and associated presentational material as this is the resource we have available for study, but we are looking towards the richer set of information that might be obtained from 'smart' environments where the event is supported by augmented (and possibly virtual) reality. In this respect, our work is informed by the meeting room scenario discussed in [21].

Our goal here is to provide a user with a means to annotate (enrich) the original material with metadata which will support linking. We see this as an example of processing that may occur several times during the capture, production and use of the material: the pipeline of enrichment. The annotation is related to an agreed vocabulary (a simple ontology) for describing the event, and different events will have different ontologies. We are interested in generating the annotation interface automatically from the ontology. In the smart workshop scenario, we envisage a variety of personalised interfaces which enable live annotation by those involved in the event.

In the next section we introduce our notion of continuous metadata that underlies this work, enabling us to generalise link streams and address live processing throughout. This is followed in section 3 by a description of our tool which supports continuous metadata. In section 4 we consider the simple ontology, based on RDF. Section 5 combines the tool and the ontology, and there is a discussion in section 6.

2 Introduction to Continuous Metadata

In previous papers we have introduced the notion of *continuous metadata* [18][5], built upon earlier work linking temporal media in the audio domain [10][6]. We view metadata simply as data about data, which can be encapsulated in a number of formats. In this situation it is not the type nor content of the metadata that is of premier importance; we are more concerned with its temporal relevance and continuous nature.

We define a *mediadata* flow to be the streamed content from and against which continuous metadata flows are derived and synchronised. The mediadata flow would normally be a multimedia stream, such as audio and/or video, transported using a real-time network protocol. The metadata, distributed as a separate flow (possibly through a number of intermediate filter nodes) might be generated on a "just-in-time" or live basis, but cannot be downloaded in its entirety before presentation begins. Although it might be the case that it is the volume of metadata that warrants streaming, we do not presume that the continuous metadata flow will be saturated with a non-stop transfer of information.

3 The HyStream System

3.1 The Temporal Linking Service

The Temporal Linking Service (TLS) was built to demonstrate the concepts of mediadata flows and metadata flows. The TLS comprises of a streaming media server, a continuous metadata server and a client that can receive and synchronise the resultant streams from both servers.

The prototype media server uses the Java Media Framework (JMF) to stream media to the client using the Real-time Transport Protocol (RTP) [19]. The client application also uses JMF to view the incoming media flow. The JMF component of the client maintains a media clock, which JMF uses internally to maintain a steady video image. The client application also uses this media clock to synchronise the incoming metadata with the media.

The persistence of metadata within the server is performed by an XML back-end. The stored form of the metadata consists of:

1. A start point for the link (a URI [2] which points to the media for which the link is relevant).
2. An endpoint (a URI to the destination of the link).
3. A human readable label for the link.
4. The time period for which the link is relevant in the respective media flow.

3.2 The Temporal Linking Transfer Protocol

The Temporal Linking Transfer Protocol (TLTP) is used between components of the TLS, and is described in more detail in [5]. In this protocol, on-time delivery of metadata is preferred over guaranteed delivery. To support small devices and features of the network layer with small buffering capability, we opt to transmit metadata "just-in-time". Since we cannot usefully incorporate late metadata into a continuous media flow the desired behaviour is to simply drop late metadata. Due to the timeliness of metadata, we dispense with the requirement for the client to acknowledge the arrival of particular fragments of data, as the server cannot usefully retransmit the data into the metadata stream.

As part of the setup process between a TLS server and client, the client determines the effective latency of payloads from the server to the client. To achieve this, the client requests the server to return the time value of the server's local clock. The client records the value of its local clock when the response is received. This action allows the client to determine the effective time difference of server to client messages. This method does not assume that the local clock on either the client or the server has been set correctly.

After calculating the effective time difference between the client and the server, the client then informs the server of this difference. Thus the server is capable of determining at any time whether a metadata payload is too late for delivery, allowing it to drop such metadata.

3.3 The Seminar Application

Our first application of the HyStream system was to deliver seminar activities held at the IAM group with metadata on slide changes. Generally, this involves a speaker, a set of presentation slides and a video camera at the event. This metadata takes the form of links into a publicly available set of slides that correspond to the ones given in the video. Figure 1 shows a clip of a seminar presentation at the IAM group. The top left of the figure shows the video of the seminar, and the temporal metadata is shown below the video, in the form of hypertext links. The section to the right of the window shows the resultant slide of the visible link.

The target of a link is determined by the mime type of the endpoint. If the mime type of the endpoint is a media type supported by the Java Media Framework, the video window is redirected to that endpoint. If the Mime-Type is not supported by JMF, the endpoint is invoked on the right hand side of the window. This allows the metadata producer to augment the continuous metadata stream with links that refer to other video resources, either live or stored.

The buttons above the presentation slide contain links that refer to other slides within the same presentation, allowing the viewer to navigate the presentation slides independently of the video window. During the seminar, the current slide is always available as a link under the video window, allowing the user to "synchronise" back with the presenter.

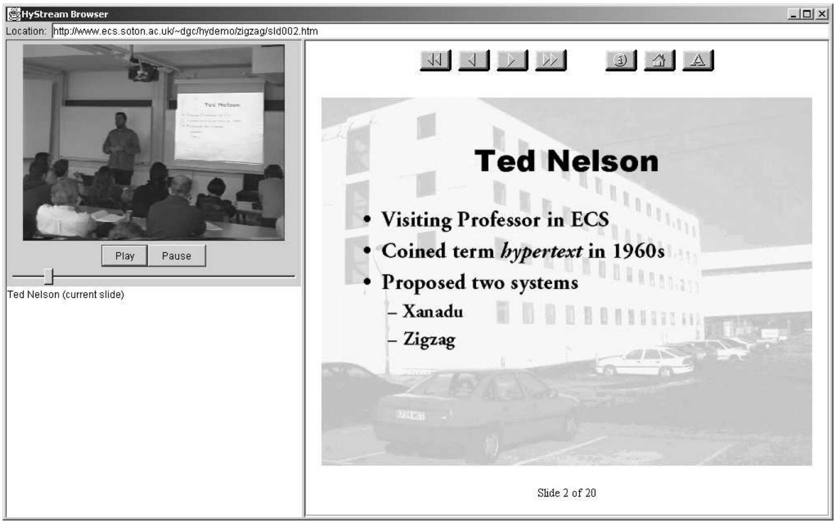

Figure 1. The HyStream seminar application

4 Authoring Continuous Metadata

Earlier research, including the HyStream system, has focused on the mechanisms to distribute and deliver continuous metadata. This work complements technologies such as RTP [19], RTSP [20], and Quality of Service provision [3][4] for network transportation of continuous mediadata. Synchronised presentation of such material within an Open Hypermedia System is usually based on concepts introduced in the Amsterdam Hypermedia Model [11] and latterly included in SMIL [1].

In this paper we will focus on the creation of continuous metadata to accompany streamed multimedia, the beginning of a branching and repetitive process of authoring and annotation; a pipeline of data enrichment.

Several standards are already defined within the broadcast entertainment industry which utilise metadata:

– TV Anytime [8] annotates media to allow navigation and integration between and within temporal content (stored on a user's device from broadcast) and external Web based data.
– MHEG marks up multimedia as collections of related objects for interchange and synchronised presentation, with a procedural language to describe interaction and presentational semantics [9].
– The Multimedia Content Description Interface, MPEG-7, defines a core set of quantitative measures of audio-visual features and structured relationships between them, associated with the original media as metadata [15].

These standards focus on augmenting content with metadata about the media itself, normally within a combined media and metadata stream. For our workshop scenario we will record continuous metadata not directly represented within or by the streamed content (video); we will represent people, objects, and events as abstract data entities. More specifically, we will record information about the workshop attendees, the video recording of the proceedings, presentation slides, and any annotations a participant might add (perhaps from a personal linkbase). However, before any such metadata can be authored, we must specify a structure to define any relationships between the entities, as well as the entities themselves - an ontology.

The Resource Description Framework (RDF) [12] is an infrastructure that enables the encoding, exchange, and reuse of structured metadata. 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) [2], 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) [14].

Collections of properties about a particular resource form a *description*; collections of properties used to describe resources within a particular resource description community form a *vocabulary*. RDF includes a mechanism for declaring and publishing vocabularies as XML Schemas, so that RDF can support any number of descriptive requirements without needing to define them. For example, the Dublin Core Metadata [22] is a simple vocabulary designed for resource discovery on the WWW which is defined within RDF. Vocabulary semantics can therefore be understood, reused and extended, in a modular manner using the XML namespace mechanism by any system supporting RDF.

Once an RDF vocabulary has been defined, gathering the metadata itself can be a manual or automated process; for a simple scenario such as ours it is trivial to collate the information by hand, however technology is already available to aid generation and, at least in part, do so transparently:

– On registering at a conference, it is usual for a delegate to be given a bar coded name badge; whenever the delegate enters a workshop or seminar room the bar code is manually scanned, compiling an attendance list typically used for marketing purposes, but also ideal for creating metadata.
– Presentation resources (such as Power Point, AppleWorks, and so on) can be parsed to build self-describing RDF sequences. Meaningfully cataloguing resources not contained within a standard presentation format (for example demonstrated software applications) would be more involved, requiring a background application to monitors file activity for the duration of the presentation.

Nonetheless, the laborious operation of temporal annotation - identifying and recording when a person is speaking, or a resource is displayed - still remains. *Smart spaces* leverage pervasive technology, combining computing embedded within the existing infrastructure with mobile devices such as PDAs [13]. Use of smart spaces, combined with more recent audio and video analysis could be applied to automate this process of temporal annotation:

– Proximity identification systems such as RFID, coupled in the longer term with speaker recognition and video analysis may be employed to reliably identify the current speaker.
– The Palette project [17], developed at Xerox FX Palo Alto Laboratory, offers a more novel approach to identifying the current slide than video analysis. Paper printouts of the slides to be shown are produced, each carrying a barcode that uniquely identifies that slide. To display their chosen slide, the speaker places the paper representation under a barcode reader, which is linked to the presentation PC. By distributing pen-style barcode readers and printed booklets of slides amongst the audience, during a question and answer session, an audience member may quickly select for display the slide which they wish to query. Thus this technology offers instant tangible benefits during the presentation, in addition to expediting the production of metadata for future use.
– Video and audio analysis techniques such as gesture recognition may allow the mood of the workshop to be captured, allowing someone to, for example, search for instances in the presentation when the speaker looked uncomfortable.

Within the IAM Group at Southampton we are investigating how pervasive computing devices can exploit RDF to communicate wirelessly within a smart environment. Other work in the group involves the exchange of relevant links between participants in a H.323 video-conference, by piping metadata over a T120 data channel. Combining these ideas in the scenario of a workshop or conference, members of the audience could exchange links amongst themselves that they find relevant as the presentation proceeds. Such ambient metadata may also be captured, potentially offering a fascinating insight into the audiences first impressions; the extent to which this information can be usefully exploited will of course depend on the richness of the ontological structure in which it is defined.

The scope for automatic metadata capture in the future is so vast that ultimately rather than simply adding new metadata, the authoring process may be as much concerned with filtering that metadata which has already been electronically harvested, to produce a personalised view or window onto the event.

5 Extending HyStream with RDF Metadata

In the context of the workshop scenario, we have extended the HyStream seminar application to interact with an RDF knowledge base when creating the link data. By describing the smart space with a set of RDF entities, we can produce an interface within the HyStream application which contains all the possible links that can occur during the meeting. In our mock-up smart space, we have produced a set of RDF descriptions that would have been created by our smart room (figure 2).

The RDF vocabulary, shown as a simplified overview in figure 3, contains:

1. Video / Media - defined by the URI of the media on the server
2. Person - expressed in the VCard schema. A number of Person entities would attend the workshop
3. Presentation - given by a particular Person(s), which will contain

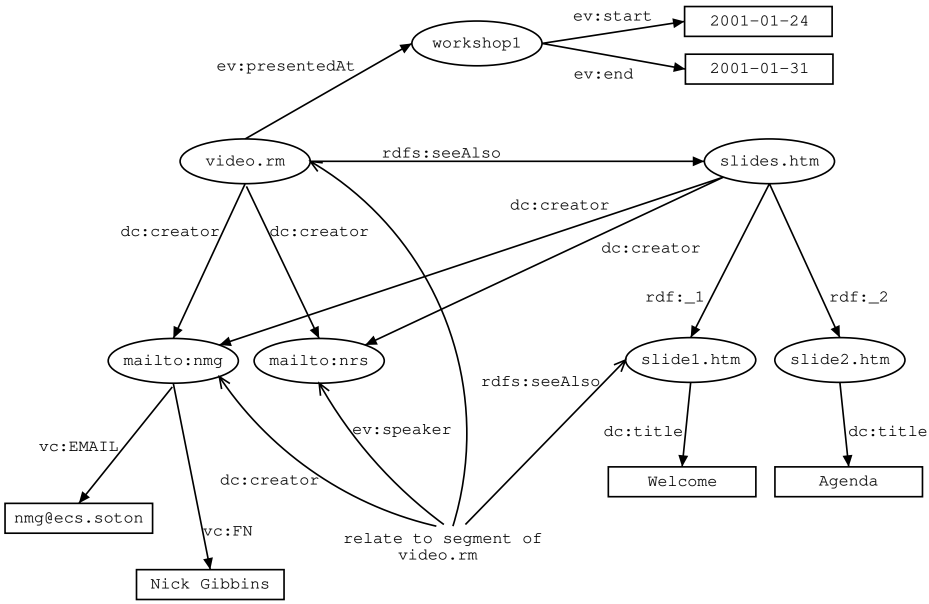

Figure 2. RDF Entity Diagram

4. Slide - the separate slides within a Presentation, which the HyStream viewer can display along with the workshop video
5. Relevant Link - an annotation offered by a Person (not necessarily the Person giving the Presentation) linking to some relevant information. This might be generated at the workshop, or at a later point when the Person is viewing the recorded Presentation

To deliver this content when displaying the workshop video the HyStream application queries the Temporal Linking Server as before. To generate the continuous metadata stream using TLTP, the server now locates valid entities for each particular time point within the RDF description of the workshop, and then sends the data to the client.

The HyStream client can also be used to author the original metadata or annotations, as shown in figure 4. A customised HTTP server generates a dynamic user interface derived from the RDF description, delivering the resources as click-able hypertext links. Each dynamically generated link is accompanied by two extra links, labelled “appear” and “disappear”, which can be used to inform the HTTP server of events that happen in the media. The authoring links invoke JavaScript to encode the current time from the media window (which can be paused to allow more accurately timed annotations) into the HTTP request. Upon receiving the HTTP requests, the server adds the new events into the RDF knowledge base.

Events that are captured by the HTTP server relate to a precise time during the video. This must be the case during a live recording, because the period of time that the event remains is unknown when the event is recorded. When the metadata is played back to the user through the continuous metadata stream, the links have time ranges. This does not pose a problem when we author and view the metadata stream sequentially. If we try to perform these in parallel, we find that we cannot deliver metadata with a known end-time. This area requires further research. Our current solution is to deliver the link with a preset duration when the duration is not known.

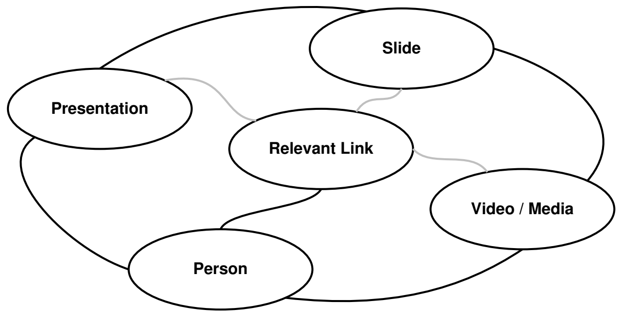

Figure 3. Simple RDF relationships

6 Discussion

6.1 Further Enhancements to the HyStream Client Using RDF

The current mechanism for creating metadata through the HyStream viewer is limited. Although the authoring interface is generated dynamically, the HTTP server expects an RDF vocabulary that conforms to the Person, Video / Media, Presentation, Slide and Relevant Link entities we have used. This should be extended so than a similar authoring interface can be generated from an arbitrary RDF knowledge base using a vocabulary previously unknown to the server.

The ontology we have used includes some basic relationships between entities. For example, a name and email together define a 'person'; that person can then be labelled as a 'creator of' a resource, and that resource in turn can have other creators attached. While it is currently assumed that any resource which is part of an RDF sequence represents a slide, a sequence might also be used to refer to a number of audio clips, or even odours released during the presentation. If the application is to sensibly handle types of resources previously unknown to it, then a hierarchy of relationships is required. The odours released might form part of a workshop on perfumery. It cannot be reasonably expected that a generic application such as ours will have in-built knowledge of this area. However a series of relationships can be defined, to explain that any particular fragrance is a member of a super class 'fragrances', and that objects of super-class 'fragrance' have odour and visual image, but do not emit sound. The application can then follow this hierarchy of relationships until it reaches a concept that it can handle - i.e. visual image, and place the resource accordingly within its user interface.

Figure 4. The HyStream authoring window

Rich ontological relationships will provide a further benefit. Each workshop is currently viewed as an isolated event, and it is not possible within the current ontological structure to reliably express the far-reaching professional networks that invariably connect such events in the real world. If rich relationships are employed, then, provided that they derive from a common set of base classes, a network of relationships between workshops can be established, allowing these lines of professional network to be traversed electronically through our application.

6.2 Representing Time in Hypermedia

Although the HyStream/RDF system has an awareness of time - it must do to serve continuous metadata - it does not store this as an intrinsic part of the RDF entities, nor as a direct relationship between them. Instead, when querying the server we specify a time parameter which returns all the objects valid at that moment - it is these objects which have relationships between themselves, not with time - the server *superimposes* a temporal perspective.

Thus, a slide is related to a video and is valid for a set time, rather than both the slide and video being related to a "first class" time entity.

Does this matter? As far as the user is concerned the result is the same. Could the way we discern time impact upon our ability to exchange metadata between hypermedia systems? Certainly the MPEG-7 group are not using RDF for storing metadata because of the lack of linking to spatio-temporal sections of data [16].

The predominant view of temporal media is of a discrete "block" within a hypermedia system - it is the synchronised presentation of the media with other hypermedia elements that has received the most attention. As such we tend to use mechanisms to jump to or from particular time indexes within the media - again, the timeline is superimposed so that we can synchronise hypermedia elements. For instance, within the HyStream/RDF application a point in the video is accessed as a link to the video itself appended with a time index.

The situation is further complicated when the temporal media becomes unbounded, such as in a live scenario. Here it becomes more difficult to deal with the media as a block within a carefully scripted hypermedia presentation, because the block has an infinite length.

So how could we represent time? One way might be to introduce time as a separate entity to which other entities are related. Although this sounds simple it changes the manner in which we create presentations - we would have to maintain a constant temporal perspective - and this would significantly complicate the system. Another alternative might be to use time as a context - viewing an entity from a particular time context could resolve that entity to the correct data for that moment.

References

1. Jeff Ayars, Dick Bulterman, Aaron Cohen, Erik Hodge, Philipp Hoschka, *et al.*. Synchronized Multimedia Integration Language (SMIL 2.0) Specification, September 2000. URL `http://www.w3.org/TR/smil20/`. Viewed 29/01/2001.
2. T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax, August 1998. URL `http://www.ietf.org/rfc/rfc2396.txt`. Viewed 03/09/2000.
3. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, *et al.*. An Architecture for Differentiated Services, December 1998. URL `http://www.ietf.org/rfc/rfc2475.txt`. Viewed 13/08/2000.
4. R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview, June 1994. URL `http://www.ietf.org/rfc/rfc1633.txt`. Viewed 13/08/2000.
5. Don Cruickshank, Luc Moreau, and David De Roure. Architectural Design of a Multi-Agent System for Handling Metadata Streams. In *The Fifth International Conference on Autonomous Agents*, pages 505–512. May 2001.
6. David De Roure, Steven Blackburn, Lee Oades, Jonathan Read, and Neil Ridgeway. Applying Open Hypermedia to Audio. In Kaj Grønbæk, Elli Mylonas, and Frank M. Shipman (editors), *Hypertext '98*, pages 285–286. ACM SIGLINK, 1998.
7. David C. DeRoure and Steven G. Blackburn. Amphion: Open Hypermedia Applied to Temporal Media. In Uffe K. Wiil (editor), *Proceedings of the 4th Open Hypermedia Workshop*, pages 27–32. June 1998. Technical Report CS-98-1, Department of Computer Science, Aalborg University Esbjerg, Denmark.
8. S. Draper, H. Earnshaw, E. Montie, S. Parnall, R. Toll, *et al.*. TV Anytime. In *Proceedings International Broadcasting Convention (IBC 99)*, pages 103–108. IBC, 1999.
9. Wolfgang Effelsberg and Thomas Meyer-Boudnik. MHEG Explained. *IEEE Multimedia*, 2(1):26–38, 1995.
10. S. Goose and W. Hall. The Development of a Sound Viewer for an Open Hypermedia System. *The New Review of Hypermedia and Multimedia*, 1:213–231, 1995.
11. Lynda Hardman, Dick C. A. Bulterman, and Guido van Rossum. The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37(2):50–62, February 1994.

12. Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification, February 1999. URL `http://www.w3.org/TR/REC-rdf-syntax/`. Viewed 03/09/2000.
13. W. Mark. Turning pervasive computing into mediated spaces. *IBM Systems Journal*, 38(4):677–692, 1999.
14. Eric Miller. An Introduction to the Resource Description Framework. *D-Lib Magazine*, May 1998.
15. Frank Nack and Adam T. Lindsay. Everything You Wanted to Know About MPEG-7: Part 1. *IEEE Multimedia*, 6(3):65–77, September 1999.
16. Frank Nack and Adam T. Lindsay. Everything You Wanted to Know About MPEG-7: Part 2. *IEEE Multimedia*, 6(4):64–73, October 1999.
17. Les Nelson, Satoshi Ichimura, and Elin Ronby Pedersen. Palette: A Paper Interface for Giving Presentations. In *CHI*, pages 354–361. 1999.
18. Kevin R. Page, Don Cruickshank, and Dave De Roure. Its About Time: Link Streams as Continuous Metadata. In *Hypertext '01*, pages 93–102. ACM, August 2001.
19. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications, January 1996. URL `http://www.ietf.org/rfc/rfc1889.txt`. Viewed 13/08/2000.
20. H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP), April 1998. URL `http://www.ietf.org/rfc/rfc2326.txt`. Viewed 14/08/2000.
21. Mark Thompson, David De Roure, and Danius Michaelides. Weaving the Pervasive Information Fabric. In S. Reich and K.M. Anderson (editors), *Open Hypermedia Systems and Structural Computing, 6th International Workshop, OHS-6, 2nd International Workshop, SC-2, San Antonio, Texas, USA, May 30-June 3, 2000 Proceedings*, volume 1903 of *Lecture Notes in Computer Science*, pages 87–95. Springer-Verlag, September 2000. ISBN 3-540-41084-8.
22. S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. Dublin Core Metadata for Resource Discovery, September 1998. URL `http://www.ietf.org/rfc/rfc2413.txt`. Viewed 03/09/2000.

Acknowledgements

This research was partially funded by EPSRC projects HyStream (GR/M84077), AKT (GR/N15764/01) and EQUATOR (GR/N15986). We are grateful to our colleagues in these projects for their support of this work, especially Luc Moreau, Nigel Shadbolt, Wendy Hall, Dave Millard and Paul Lewis. We also wish to thank Hugh Glaser and Jian Meng for the Sheffield AKT workshop case study.