Hyperdoc: An Adaptive Narrative System for Dynamic Multimedia Presentations

David E. Millard, Christopher Bailey, Timothy Brody, David Dupplaw, Wendy Hall, Steve Harris, Kevin R. Page, Guillermo Power, Mark J. Weal
Intelligence, Agents, Multimedia,
Dept. of Electronics and Computer Science,
University of Southampton, U. K.
{dem, cpb99r, tdb01r, dpd98r, wh, swh, krp, gp98r, mjw}@ecs.soton.ac.uk

ABSTRACT
Previous approaches to adaptive presentation have highlighted conflicts of interest between adapting the content, media type or quality and structure of a presentation. By using the three level model of narrative as a starting point, we can gain a greater understanding of the relationship between these. In this paper we present the prototype Hyperdoc system, which applies adaptive techniques at the separate levels of narrative in order to achieve a tailored web presentation within a certain domain.

1. INTRODUCTION
As new methods of media delivery become available, such as digital television and broadband Internet access, opportunities arise for more interesting user interaction.

For several years now the research communities of multimedia and hypermedia have investigated how this could be done, exploring the ways in which dynamic multimedia presentations could be generated automatically. These range from systems that make choices between alternative media according to bandwidth restrictions [4] to those that extract metadata from previously opaque media and use that information to reconstruct customised presentations, for example analysing a news feed and then re-sorting a subset of the items according to the users expressed interests [9].

Other research has applied adaptation to the production of material itself, creating a virtual videographer that chooses shots and images automatically [8], for example choosing either to focus on the presenter or the projection during the recording of a seminar.

The choices made in producing a presentation are similar to those made by a hypertext system when offering readers links. The similarity between cinematic narrative and hypertext has been noted before [10] as well as the need for those choices to either maintain the rhetoric that the author intended [17] or to produce alternative rhetoric based on the same base of data [5].

The ability of human editors to preserve or create narrative is not easy to replicate with a machine. One method is to limit the choices at every stage to those that maintain the narrative by using the hypertext solely to make cinematic decisions rather than rhetorical ones, for example choosing different camera shots of a particular scene [7].

In this paper we describe the Hyper-documentary system (Hyperdoc) that we have developed to address the problem of supporting narrative in multimedia presentations. The Hyperdoc is not a system for information analysis or extraction, rather it is a demonstrator that promotes a non-linear approach to production that enables adaptation at every level of the narrative.

1.1 Levels of Narrative
We have taken Bal's view of Narrative as the uppermost of three layers [1] (shown in Figure 1):

1. *Fabula.* This represents the raw chronological events and information. For example individual events that occurred during the Napoleonic wars.
2. *Story.* For any given fabula we can imagine multiple stories, each is a particular trail through the fabula, presenting a subset of the information from a particular perspective. For example, events concerning Napoleon's participation in the Battle of Waterloo.
3. *Narrative.* Any given story needs to be given a form to be told, in effect it needs to be rendered. For example, a novel following Napoleon through the battle. A film of the same story would be regarded as a separate narrative.

Although this model has been developed for use in the realm of critical theory, a tool with which to examine existing narratives, we can use it to understand the ways in which existing work on multimedia adaptation fit together. For example, we can categorise the virtual videographer [8] as

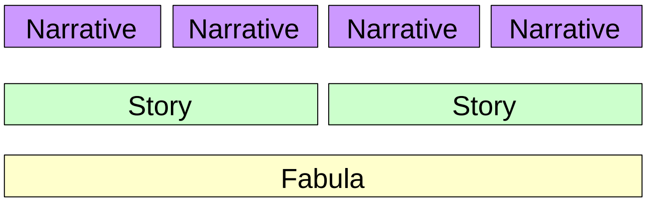


Figure 1: Bal's Layers of Narrative

concerned with adaptation at the level of Fabula, while the work on preserving rhetorical structure [17] is about generating the narrative layer from the story layer without breaking the story structure.

1.2 What do we mean by Hyper-Documentary?

With a traditional documentary it is the choices made by the production team that build up the narrative layers. We term a hyper-documentary as one in which the layers are built up automatically from a large *fabula* base, choosing from many *story* structures. Thus many linear multimedia documentaries (*narratives*) can be created by the system, tailored to the requirements of the viewer.

In the next few sections we describe the Hyperdoc system that we have developed to demonstrate this approach. The Hyperdoc project was undertaken by around fifteen people over a period of approximately two months. The objective was to explore the issues of producing a hyper-documentary, from generating the media in the fabula, through the authoring of story templates to rendering the presentation for viewing on the web. Firstly we will examine the interface.

2. THE INTERFACE

The Hyperdoc interface (shown in Figure 2) asks the user to provide three pieces of information; the type of documentary they would like to see, the subject of the documentary and their preferences for style and content.

For example, if the user chose to view a documentary on a particular individual, the subject would then be the individual's name.

Gathering user preferences for style and content is more complex because the interface has to be simple, yet expressive enough to provide a complete set of user requirements to the Hyperdoc engine.

User modelling and adaptive systems often employ explicit modelling where users are questioned on their preferences before interacting with the system. While this provides a high level of detail about the user, explicit modelling is often seen as time consuming and intrusive [13, 15]. The alternative technique, implicit modelling, obtains user preferences through observations of the users activities as they interact with the system [2]. However implicit modelling provides only positive-exemplars of user actions and requires both lengthy user interaction and the ability to track users over multiple sessions [18].

The Hyperdoc system does not currently provide enough interaction to allow implicit user modelling. The compromise is to combine the user preferences into a single interface component – in this case a slider (a graphical object that can be moved between various settings using a drag-and-drop mouse action). The user is presented with a slider to indicate the required style of presentation loosely based on two well known television styles. The slider moves between the labels 'BBC ' and 'MTV'. BBC represents a formal, content-rich presentation while MTV represents a quick, light-hearted presentation. The user chooses a position between these two settings depending on their required presentation. The slider produces a value that is later mapped to style meta-data such as depth and mood (described in section 4.2).

3. ARCHITECTURE

Figure 3 shows the Hyperdoc architecture. The diagram has been coloured to reflect the parts of the system which deal with the three levels of Narrative identified in Figure 1. Each of the three levels will be discussed in more detail later; this section seeks to present a basic overview of the architecture.

3.1 The Fabula

The fabula consists of the raw building blocks of the narrative. In the case of the hyper-documentary this constitutes the raw data files and meta-data information concerning them. The raw media files are stored on a central server. The potential temporal nature of the media requires the server to be equipped with streaming capabilities to deliver temporal media streams when requested.

The meta-data about the media files is stored in an SQL database. PHP scripts allow the information to be entered via a Web interface in the form of a series of Web Forms. A more complete description of the database including the tables held within it is given in Section 4.2.

3.2 The Story

The story is constructed when a request is made to the Hyperdoc system for a documentary. The request comes in the form of input from a viewer via the web front end interface (Figure 2). This provides both user preferences and a request for a particular type of story and content.

The HyperDoc engine uses story templates to instantiate a story using the meta-data information in the database retrieved through an SQL connection.

3.3 The Narrative

Once instantiated, the story template is passed to a renderer. In this configuration of the architecture it is a SMIL renderer but the renderer is designed to be interchangeable and other renderers could be created for text, HTML or more complex media such as Flash. The renderer makes decisions about the most appropriate way of rendering the story into a narrative, based on the delivery format and also the preferences specified by the user. The final output is then displayed to the user through the appropriate technology; in the case of SMIL, this might be RealPlayer One launched from a web browser.

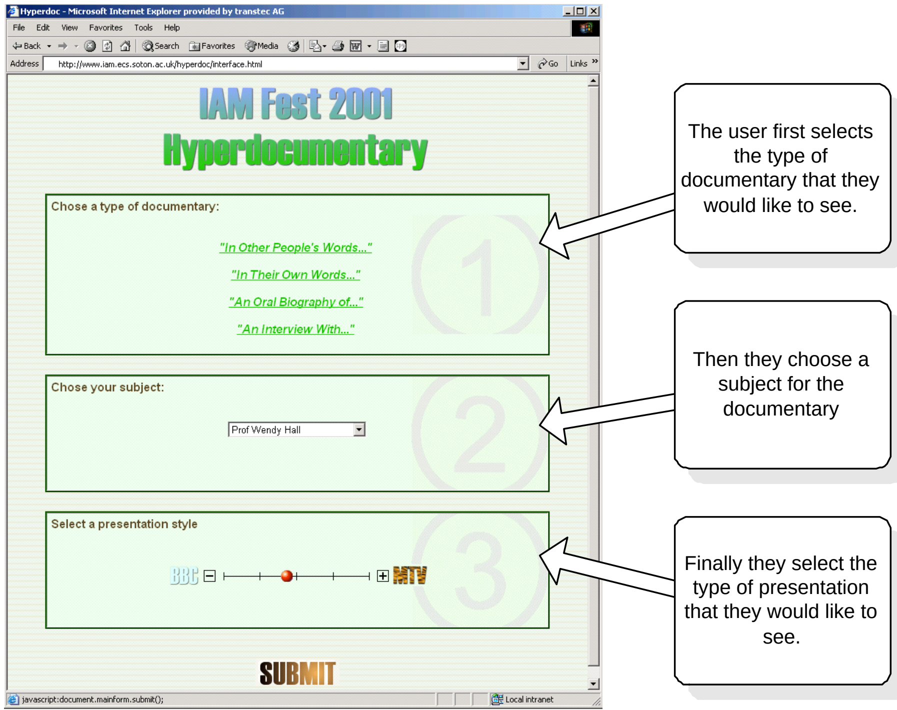


Figure 2: The Hyperdoc Request Form.

4. FABULA: GATHERING CONTENT

Content becomes more important when trying to produce multiple stories and narratives. Without a large corpus of media fragments, possibilities for dynamic adaptation are limited. For the Hyperdoc we had added requirements in that we needed the media fragments to be as atomic as possible, deliverable over the web and respectful of copyright.

To this end we decided to produce a Hyper-Documentary about the IAM Group at the University of Southampton. This is a large group of around one hundred researchers many of whom have worked with one another for many years. The Hyperdoc could then be added to the group's web site allowing visitors to obtain a novel view of the people and research activities via the use of the group's own technology.

We discussed the content that would be appropriate for this subject and also could be produced in the right forms, finally deciding on several initial areas:

1. Video interviews with academic staff. There are approximately fifteen members of academic staff in the group. By interviewing them all with a fixed set of questions we hoped to gain a body of interviews that would form the basis of the documentaries. Many of the questions were in relation to other staff members and projects. E.g. *Who is the best person you have worked with and why?* Questions of this type can be reused in multiple stories. As well as the video, the interview scripts were also added to the archive.

2. Photographs from the groups social archives. These formed a small social history of the group over the last ten years. In addition to the photographic material we asked Professor Wendy Hall (the group's founder) to record an audio narration for each photograph.

3. Pictures and Video shots around campus and the groups laboratory. These were taken to give the rest of the material some kind of context.

4. Brief textual descriptions of the group's projects taken from the IAM web pages, which could be used as additional exposition.

It was hoped that by having a large set of multimedia fragments containing information related along several informatic axis the story layer would be able to create many different paths through the information.

4.1 Music

One of the media forms that we have currently omitted from the Hyperdoc is music. Music is often used in documentaries as a method of creating continuity between discrete scenes. Continuous, looping, unsynched "mood" music might be adequate for the presentation to achieve certain effects based on the user's preferences (for example, an upbeat piece for a light hearted presentation), however music based on the story structure itself would be more desirable.

This would mean that the scenes would be synched to the music structure (for example changing the scene on music beats), which requires substantial processing at the presentation layer to detect the appropriate markers and care would have to be taken during the presentation to ensure the presentation did not seem disjointed.

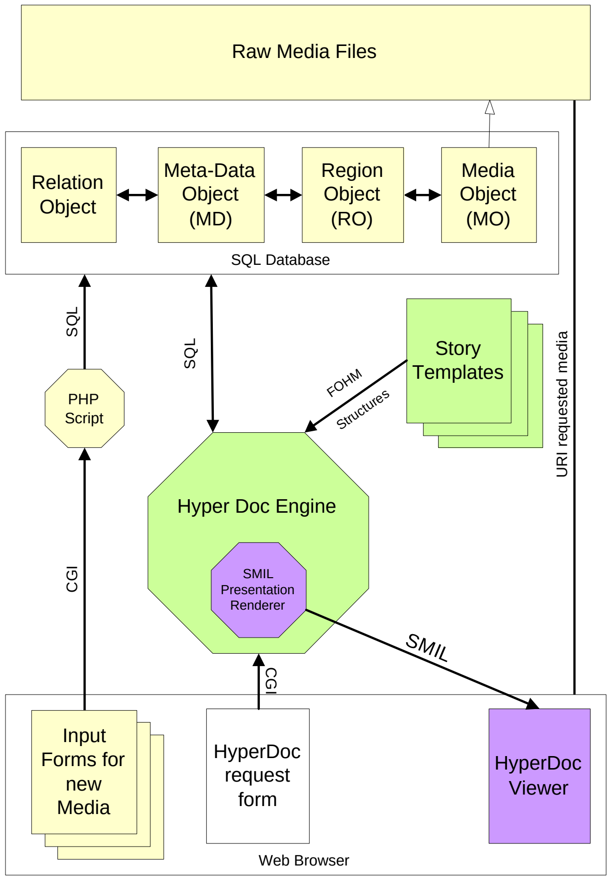


Figure 3: The Hyperdoc Architecture.

Because of these complexities we have not added any music to the Fabula of the prototype Hyperdoc. Our intention is to experiment with music at a later time. The music tracks would have to be royalty free, of which there are many sources available on the Internet, commercially, or bespoke for the project.

4.2 Storing the Fabula

The media that was captured was stored on a central infrastructure machine with URLs used for access. URLs were chosen as they provide an abstraction over the method of file delivery, as videos are streamed using RTSP, while images and text are delivered using HTTP. We did not break the data into fragments for storage (e.g. each interview was stored as a single video file), instead we store indexes and other meta-data in a free, relational DBMS (MySQL) (e.g. the database treats individual answers as separate objects).

The database schema consists of four distinct types: Media Objects, Region Objects, Metadata Descriptors, and Relations.

A Media Object (MO) describes a physical file, for example a RealVideo recording. The MO provides the URL and MIME type.

A Region Object (RO) sub-divides an MO into distinct, describable chunks. For example, a time range for a single answer in a video interview, a paragraph of text from a document, or an area of an image, representing an individual from a group photograph.

A Metadata Object (MD) contains metadata describing a fragment of information. It was decided that Dublin Core would be used as a basis for the metadata, with some specific qualifications that could be used by the 'higher' narrative layers (describing the mood and style of the content). It was initially considered that anyone would be allowed to enter metadata, however, in the first instance we were constructing the categorisation system as we were encoding the data, it was therefore easier for the encoding to be carried out by a few individuals (in order to be consistent). A list of the metadata used is shown in Table 1.

A Relation Object provides a typed pair relation between media, region, and metadata objects. This requires four tables (MD-MD, MD-MO, MD-RO, RO-MO).

Structural triples were used for metadata and relation objects. These are three columned tables of a 'source' / 'relation' / 'target'. For example a metadata entry might be '121' / 'Contributor' / 'mjw', where '121' is the id of the metadata object, 'Contributor' is the triple type, and 'mjw' is the value. A relation might by (in the MD-RO table) '97' / 'describes' / '217', where metadata object '97' *describes* region object 217.

Using table joins, the triples provide an extensible and flexible mechanism for expressing queries, for example *"return the region object ids, where there is a metadata object that describes that region object, which has a relation 'Contributor', and that contributor is 'mjw' "*. The drawback of triples is the computational complexity of computing large numbers of table joins (especially when the result sets are large).

4.3 Authoring

With a system in place to store the fabula, simple mechanisms were needed to enable authoring of the constituent material. Encoding the media into a suitable format is a relatively straightforward process which is easily automated, whereas marking up the material with metadata is often perceived as an extra burden by the author. While not a fundamental rationale for the Hyper-Documentary, the authoring of metadata forms a crucial part of the process; without this metadata an adaptive story cannot be constructed.

The majority of source media for the Hyper-Documentary was authored in situations where simultaneous metadata creation was not possible (e.g. recording with a video camera) and markup was a secondary activity occurring at a subsequent time. In the future we envisage additional metadata production at even later stages, when users of the system author their own annotations that further enrich the content of the fabula.

To aid this scenario we developed a set of web pages using the PHP server-side scripting language to query and create objects in the hyperdoc database. These comprised of four main dialogues, allowing the author to:

MD Element	*Description*
Title	short human readable description
Creator	id of author
Subject	string representing the subject matter
Publisher	publishing organisation
Contributor	id of persons who contributed
Date.Created	the creation date of the source
Type	semantic type of meta-data, e.g. question, answer etc.
Coverage.From	starting date of the subject
Coverage.To	ending date of the subject
Rights	organisation that owns the rights
Description.Mood	the 'mood' of the data (e.g. trivial, serious, etc.
Description.Depth	the amount of detail in the data (e.g. deep, shallow, etc.)

MO and RO Element	*Description*
Format	the mimetype of the described media
Source	id of original item, e.g. ISBN, URL, etc
Language	language used

Table 1: Metadata Elements

1. add a new Media Object based on a media URL and MIME type
2. create Region Objects and associate them with one or more Media Objects
3. generate a Metadata Object and associate it with one or more Media or Region Objects
4. create a relationship between two Metadata Objects (e.g. 'answerOf')

Although these web forms ease the process of metadata authoring, it is still a relatively laborious and lengthy task. We explored several ideas through which we could semi-automate or assist metadata generation.

The first involved the recording of short video-diary entries using a PC with an attached web-cam. We made this facility available to the IAM Group for a week and encouraged them to make entries. Because of the controlled environment, much of the metadata relating to media itself (type, time and date, length etc.) could be captured automatically, with the remaining metadata gathered through prompted questions that were part of the recording process.

The second initiative provided members of the research group with an email address to which they could carbon-copy emails concerning the hyperdoc project. We could harvest much of the metadata automatically from the original email, and with enough data we hoped to build a body of background information that might be useful within the Hyperdoc.

In both cases the user response was disappointing and did not produce enough material to add to the final system. Even these simple metadata gathering systems appear to present enough of a barrier to prevent wide participation. From this limited experience we might conclude that future metadata authoring will either need to be fully automated (with the associated technical and privacy problems), or come only from those with a vested interest in creating it.

5. STORY: STRUCTURING THE FABULA

From a certain perspective hypertext can be seen as a sequencing task, which allows dynamic paths to be followed through a collection of interlinked fragments. Traditional hypertext systems allow the reader to explore the information space freely but as a consequence offer little support for narrative. Several technologies offer solutions, in particular trails or tours [11] create a rigid structure that readers may choose to follow. A more dynamic approach is offered by sculptural hypertext systems [3, 19], in which everything is initially interlinked and links are removed based on a set of conditions.

5.1 Story Templates

A Hyper-documentary requires a middle route, a dynamic trail which is flexible, while remaining structured. To this end we take inspiration from Story Grammers [16]. These effectively form a story schema or templates that describe generally how the parts of the fabula should be sequenced into a story. Given an appropriate fabula the template can be instantiated by slotting in the bits of the fabula that fulfil the requirements of the template at each point.

To make story templates adaptive they must become contextual structures. In other words the templates must change according to the context of the viewer (e.g. a documentary structure for experts might contain more detailed exposition than one for novices).

In the Hyperdoc system this was accomplished by authoring the templates in the Fundamental Open Hypermedia Model (FOHM) [14], this is an implementation independent model for representing arbitrary hypermedia structures (such as links, tours, etc.). The FOHM templates were stored as XML and then loaded into the Auld Linky structure server [12], which allows clients to query the FOHM structures in context (inappropriate parts of the structure are removed before the structure is returned).

This pruning takes place because structures in FOHM can have Context objects attached to them. Context objects are attached to parts of the template describing when that part of the structure is visible. When a query for the template is made a query context is also sent, if that doesn't match with the context on the structure then that part of the structure is removed before it is returned.

For example, consider a link expressed in FOHM that contained two destinations, one of which is only appropriate for adults to see. In this case we could attach a Context object to the destination that expressed it was not for children. When Linky is queried the user can specify the context of that query. If they queried in the context of a child they would only see one destination, whereas if they queried in the context of an adult they would see both.

It is also possible to attach Behaviour objects to the structures. These describe actions that should be taken when that part of the structure is parsed by a client.

The templates are stored uninstantiated, this means that they contain Behaviour objects specifying queries into the database rather than referencing media fragments directly. The Hyperdoc story builder parses the template and instantiates it into a story structure (also expressed in FOHM) by performing the queries. This is then passed to the renderer.

The story structure is composed of the following sub-structures:

- The basic structure used is the *Sequence*. This represents a list of fragments or sub-structures that have to be rendered in a given order.
- A *Set* structure contains one or more fragments or sub-structures, all of which should be rendered but the order is not specified.
- A *Concept* structure contains one or more fragments, any sub-set of which may be used (in any order)

The templates operate via a blackboard of named variables. When a template is invoked several variables might have to be set up initially based on the user's preferences. Figure 4 shows a simple template that represents a single question and a set of answers. In this case there is one initial variable called *?question* that contains the id of the Metadata Object that represents the question to use.

The story builder crawls over the template and looks at each sub-structure in turn. There are two ordered members of the question template. The first will be instantiated into the question, the second into the set of answers. It is the behaviour objects attached to the template that tells the story builder how to instantiate the queries. There are four types of behaviour event:

- *buildconcept*. This instructs the builder to retrieve the metadata object with the given id and construct a concept structure with it. In the case of Figure 4 it will build a concept containing all the different media versions of the question (audio, text, etc.).
- *query*. This is a behaviour that adds new variables to the blackboard. It instructs the builder to make a query and fill in the named variable with the results (note each variable can store a set of possible values). E.g. (EQUALS, answerOf, ?b, ?question) retrieves all the metadata objects which have a relation *answerOf* with the question metadata object with id of *?question* and stores them in *?b*. The variables containing the user preferences are used at this stage to ensure that only those fragments that fit the chosen style and mood are selected.
- *foreach*. This behaviour takes some of the values already on the blackboard and uses them to either build a data item (which represents a single media fragment) or invoke a sub-template. In this case it instructs the builder to iterate over the values in the variable *?b* and build a data item with each of them.
- *validitycheck*. This checks that, after the queries have been made, we are left with a valid structure. In this case *(EQUALS, 2)* means that the question structure must have two members (i.e. we do not present a question if there are no answers), while *(GREATER,0)* ensures that the set of answers must have at least one member.

  The results of the validity checks propagate up the structure. So if the set contained no answers then its validity check would fail and the set would not be added to its parent sequence. In turn that would mean that the sequence only had one member and so that would fail as well, which would mean that there was no valid question template for the initial question id.

In summary, this example template has a question id as its input. It retrieves the object(s) that represents that question and inserts them as a concept at the start of the sequence. It then retrieves all the answer objects that correspond to that question. It then iterates over those and inserts them, as a set, into the second position of the sequence. Finally it carries out a validity check to ensure that the set has at least one member and that the parent sequence has two members.

Starting with simple templates to represent low level structures (e.g. questions), we can then define more complex structures that use these as sub-structures (e.g. an interview) and finally incorporate that into an overarching structure (e.g. a documentary).

Once the queries have all been made the instantiated template is passed to the renderer, whose job it is to create the final Narrative layer.

6. NARRATIVE: PRESENTING THE STORY

To produce the final narrative a presentation mechanism was needed. Because of the wide variety of materials being produced for content a flexible multimedia format was required. SMIL 2.0 was chosen for a number of reasons, namely:

```
<association id="question_template">
   <structure>sequence</structure>
   <relationtype>question and answers</relationtype>
   <description>Question and Answer structure</description>
   <feature>position</feature>
   <behaviour>
      <event>validitycheck</event>
      <behaviourvalue key="bindings">(EQUALS, 2)</behaviourvalue>
   </behaviour>
   <binding>
      <behaviour>
         <event>buildconcept</event>
         <behaviourvalue key="id">?question</behaviourvalue>
      </behaviour>
      <reference><data/></reference>
      <featurevalue feature="position">1</featurevalue>
   </binding>
   <binding>
      <reference>
         <association>
            <behaviour>
               <event>validitycheck</event>
               <behaviourvalue key="bindings">(GREATER,0)</behaviourvalue>
            </behaviour>
            <structure>set</structure>
            <relationtype>answers</relationtype>
            <binding repeatable="true">
               <behaviour>
                  <event>query</event>
                  <behaviourvalue key="clause">(EQUALS, answerOf, ?b, ?question)</behaviourvalue>
                  <behaviourvalue key="include">true</behaviourvalue>
                  <behaviourvalue key="queryname">query4</behaviourvalue>
               </behaviour>
               <behaviour>
                  <event>foreach</event>
                  <behaviourvalue key="queryname">query4</behaviourvalue>
                  <behaviourvalue key="index">?b</behaviourvalue>
               </behaviour>
               <reference><data/></reference>
            </binding>
         </association>
      </reference>
      <featurevalue feature="position">2</featurevalue>
   </binding>
</association>
```

Figure 4: Example of a FOHM Question and Answers Template

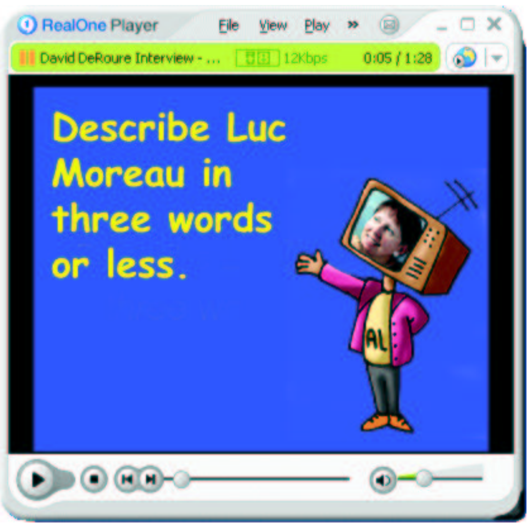
- It is an established standard.
- It is supported on multiple platforms by viewers such as Real One Player.
- Being textual, the format is easy to generate from a servlet, requiring no compilation by external software.
- It handles temporal media in a simple, effective manner.
- There are a variety of transition and presentation techniques built into the standard that could be used to provide interesting, customisable, presentations.

A SMIL renderer was written that takes the instantiated story template from the hyperdoc servlet and renders it as a SMIL document.

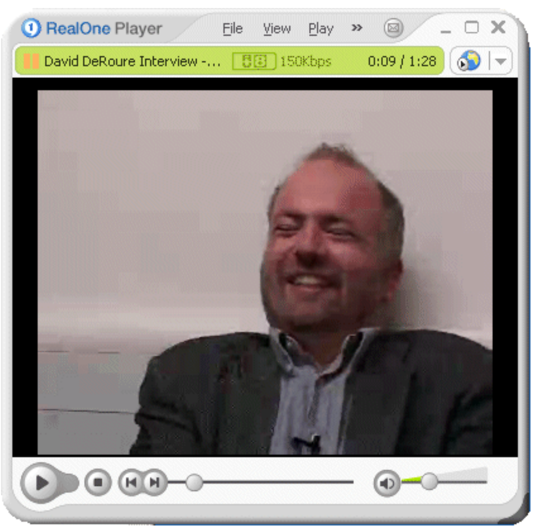
6.1 Rendering Decisions

The SMIL renderer is passed an instantiated template structure which it must parse to produce the final documentary. The structures in the template partially map to the structures supported by SMIL. A FOHM sequence structure can be constructed as a SMIL ⟨`seq`⟩ . The set structures, if applicable might be rendered either as a ⟨`seq`⟩ or as a ⟨`par`⟩ structure (for instance it might be a set of photographs that could be rendered as a collage). Where the set structure consists of temporal media, such as a set of answers to a question, the Renderer will play them sequentially but is free to choose the order of play. This might be based on the meta-data associated with the fragments of information. For example, the renderer may choose to play all of the serious answers first to end the section on a lighter note.

Currently, the Renderer makes no decisions on the overall timing of the piece and as it renders all the video fragments

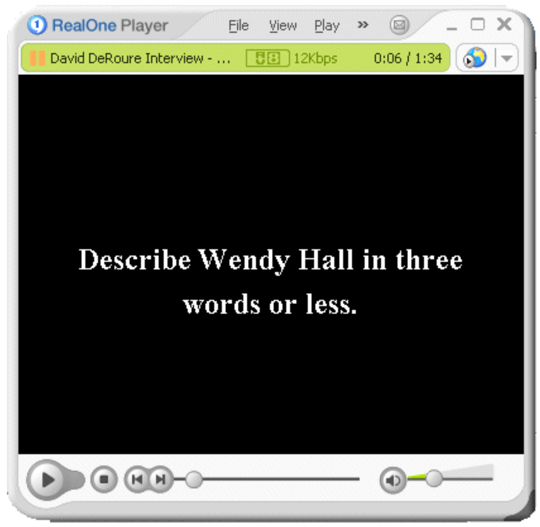


(a) The MTV style caption.

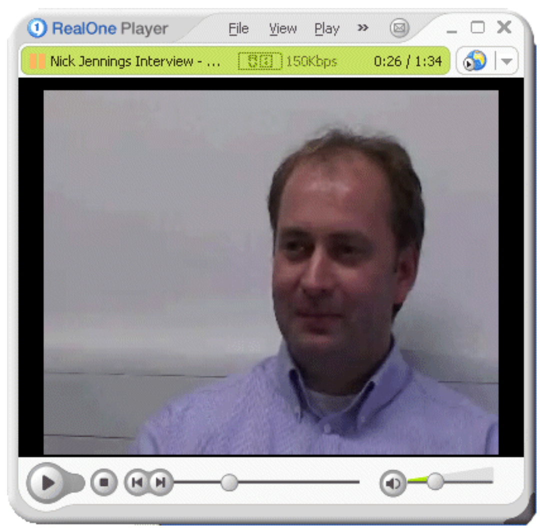


(b) A Lighthearted interview fragment.

Figure 5: The MTV style of presentation, both in caption style and content.



(a) The BBC style caption.



(b) A more serious interview fragment.

Figure 6: The BBC style of presentation, both in caption style and content.

into a sequence, the documentaries can potentially become quite long. One possible improvement to the renderer would be to allow it to chose more concise material in order to keep the documentary down to a specific length. A number of strategies could be employed in achieving this and this will form future work on the hyperdoc system.

The base structures in the template are concepts, with these the renderer must decide which of the available information fragments to use. The multimedia capabilities of SMIL lend themselves well to producing a documentary from streamed video and so if a video fragment is available it will be chosen. If not, the renderer may choose to place some text on the screen, perhaps running audio over the top of it if available. An HTML renderer might choose to use images and text over temporal media.

Where the Renderer has a piece of text to render (for example, the interview questions), it was decided to use a servlet to produce a graphical caption from the piece of text; this was due to the limitations of text rendering in SMIL. This also opened up opportunities for further customisation of the documentary style based on the user preferences.

6.2 Caption Rendering

The rendering of captions is based in part on the user preferences, passed to the renderer by the hyperdoc servlet. The objective is to make the captions fit in with the overall feel of the story. There are several different ways in which this can be done:

Caption Style - The captions generated for the text fragments are varied according to the preferences. If the user is requesting a more lighthearted documentary, the captions will be rendered on a multicoloured background using graphics and a 'fun' font. For more serious preferences the captions are rendered in simple white text on a black background. An impression of how this works can be gained by comparing Figure 5(a) and Figure 6(a).

Caption Pacing - For 'MTV' style documentaries the pacing is kept fast and snappy and the captions are displayed for a shorter period of time. For the 'BBC' style, the captions are displayed for longer.

Video Effects - SMIL 2.0 provides a variety of animation and video effects that could be used to change the presentation of the captions. The current version makes no use of these as yet, but clearly the tone of the documentary could be enhanced by the use of appropriate transitions, perhaps captions spinning into place for the 'MTV' style or slowly fading in and out for 'BBC'.

6.3 The Overall Effect

Figure 5 shows a rendered documentary with the more colourful (if not seen in black and white) captions. The content is generally more lighthearted, with more humorous clips appearing. In contrast Figure 6 has more austere black and white captions and the tone of the content is more serious.

7. CONCLUSION

The Hyperdoc system is still in its prototype phase, although we are hoping to make it accessible to the outside world via the group web site within the next few months. It is the contextual story templates, media content and metadata which is proving time consuming to produce, at the time of writing around half of all the content has been added to the system. Several templates have been written and we are now in the phase of re-evaluating our content (fabula) in the light of the produced narratives.

There is a view that states that in fiction the relationship between fabula and narrative is not causal in the way that the layers suggest but that the content of the fabula depends on a large extent to the purpose of the overall narrative [6], for example some story structures depend on a twist or surprise that is unveiled at the end of the story, fiction authors may well engineer such an occurrence in the fabula just to satisfy this structure.

This tension also exists in the Hyperdoc system, as it is difficult to decide what should be entered into the fabula without considering the story layer and thus the templates that will be used. We have found that the process is a cycle and are now producing more media in order to satisfy the more advanced story templates that we are authoring. It is an open question whether this issue would be resolved by a larger corpus of media and more detailed metadata to start with.

7.1 Levels of Adaptation

Within the Hyperdoc, adaptation is supported independently at the following three levels:

1. *Adaptive Content.* The media objects and metadata in our database create the fabula layer. Rather than storing media fragments directly, queries into the database are stored instead, resolving dynamically based on the user preferences.
2. *Adaptive Structure.* The story templates are stored in Auld Linky, which is a contextual structure server. As a result the templates change according to the context in which they are retrieved.
3. *Adaptive Presentation.* Once the templates have been instantiated by making the queries to the database, the completed structure is passed to a renderer, which has the job of converting it into a human viewable format (in this case SMIL). The editing and rendering decisions made depend on the users specified preferences.

We believe that by viewing adaptive presentation through three levels of narrative it is possible to achieve a greater understanding of the consequences of any adaptation and build systems that minimise unfavourable consequences for adjacent layers (e.g. the impact of choosing different media formats on overall rhetoric).

7.2 Future Work

While we are continuing to build up the media corpus of the existing Hyperdoc we are also exploring how we might use some of the more advanced effects of SMIL 2.0 to alter the presentation layer (fades, wipes etc.) and also considering adding additional metadata to the database in order to allow the story templates to build more subtle structures (for example marking appropriate exposition and labeling opinion vs. facts).

In addition we are working on the Artequakt project (a joint effort between three major projects within the IAM group) to construct a system that extracts knowledge from the web and automatically populates an ontology. This will then form the fabula layer for narrative-based presentation tools, providing an alternative mechanism for authoring metadata.

8. ACKNOWLEDGMENTS

Many people have been involved in the discussion, planning and development of the Hyperdoc system, we would particularly like to note the contributions of Mark Bernstein, Ted Nelson, Elizabeth Coulter-Smith, Su White and Arouna Woukeu. This research is funded in part by EPSRC IRC project "Equator" GR/N15986/01 and EPSRC IRC project "AKT" GR/N15764/01.

9. REFERENCES

[1] M. Bal. *Narratology: Introduction to the Theory of Narrative*. University of Toronto Press, 1978. Trans. Christine van Boheemen. Torento. 1985.

[2] M. Balabanovic. An Interface for Learning Multi-topic User Profiles from Implicit Feedback. In *Proceedings of the AAAI-98 Workshop on Recommender Systems, Madison, Wisconsin*, 1998.

[3] M. Bernstein. Card Shark and Thespis: exotic tools for hypertext narrative. In *Proceedings of the Twelth ACM Conference on Hypertext and Hypermedia, Arhus, Denmark*, pages 41–50, 2001.

[4] Susanne Boll, Wolfganag Klas, and Jochen Wandel. A Cross-Media Adaptation Startegy for Multimedia Presentations. In *Proceedings of the Seventh ACM Conference on Multimedia, Orlando, Florida*, 1999.

[5] Michel Crampes, Jean Paul Veuillez, and Sylvie Ranwez. Adaptive Narrative Abstraction. In *Proceedings of the '98 ACM Conference on Hypertext, June 20-24, 1998, Pittsburgh, PA*, 1998.

[6] Jonathan Culler. Story and Discourse in the Analysis of Narrative. In *The Pursuit of Signs*, page 178. London: Routledge and Kegan Paul, 1981.

[7] Vardi G. Navigation Scheme for Interactive Movies with Linear Narrative. In *Proceedings of the '99 ACM Conference on Hypertext, February 21-25, 1999, Darmstadt, Germany*, 1999.

[8] Micheal Gleicher and James Masanz. Towards Virtual Videography. In *Proceedings of the Eighth ACM Conference on Multimedia, Los Angeles, California, USA*, 2000.

[9] K. Lee, D. Luparello, and J. Roudaire. Automatic Construction of Personalised TV News Programs. In *Proceedings of the Seventh ACM Conference on Multimedia, Orlando, Florida*, pages 323–332, 1999.

[10] C. Mancini. From Cinematographic to Hypertext Narrative. In *Proceedings of the Eleventh ACM Conference on Hypertext and Hypermedia, San Antonio, Texas, USA*, pages 236–237, 2000.

[11] Catherine C. Marshall and Peggy M. Irish. Guided Tours and On-Line Presentations: How Authors Make Existing Hypertext Intelligible for Readers. In *Proceedings of the '89 ACM Conference on Hypertext, Nov. 5-9, 1989, Pittsburgh, PA*, pages 15–26, 1989.

[12] D.T. Michaelides, D.E. Millard, M.J. Weal, and D. DeRoure. Auld leaky: A contextual open hypermedia link server. In *Hypermedia:Openness, Structural Awareness, and Adaptivity (Proceedings of OHS-7, SC-3 and AH-3), Published in Lecture Notes in Computer Science, (LNCS 2266), Springer Verlag, Heidelberg (ISSN 0302-9743)*, pages 59–70, 2001.

[13] S.E. Middleton, D. C. De Roure, and N.R. Shadbolt. Capturing Knowledge of User Preferences: ontologies on recommender systems. In *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001), Victoria, B.C. Canada*, October 2001.

[14] D.E. Millard, L. Moreau, H.C. Davis, and S. Reich. FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability Between Hypertext Domains. In *Proceedings of the Eleventh ACM Conference on Hypertext and Hypermedia, San Antonio, Texas, USA*, pages 93–102, 2000.

[15] M Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 272–281, 1994.

[16] Vladimir Propp. *Morphology of the Folktale*. 2nd ed. Austin: University of Texas Press, 1927. Trans., Laurence Scott. 1968.

[17] L. Rutledge, B. Bailey, J. V. Ossenbruggen, L. Hardman, and J. Geurts. Generating Presentation Constraints from Rhetorical Structure. In *Proceedings of the Eleventh ACM Conference on Hypertext and Hypermedia, San Antonio, Texas, USA*, pages 19–28, 2000.

[18] C. Stevens. *Knowledge-based assistance for accessing large, poorly structured information spaces*. PhD thesis, University of Colorado, Department of Computer Science, Boulder., 1993.

[19] Mark J. Weal, David E. Millard, Danius T. Michaelides, and David C. De Roure. Building Narrative Structures Using Context Based Linking. In *Proceedings of the Twelth ACM Conference on Hypertext and Hypermedia, Arhus, Denmark*, pages 37–38, 2001.