Towards a unified data model for audiovisual active reading

Bertrand Richard
LIRIS UMR 5205
University of Lyon
43 Boulevard du 11 Novembre 1918
69622 VILLEURBANNE CEDEX
FRANCE
Email: bertrand.richard@liris.cnrs.fr

Yannick Prié
LIRIS UMR 5205
University of Lyon
43 Boulevard du 11 Novembre 1918
69622 VILLEURBANNE CEDEX
FRANCE
Email: yannick.prie@liris.cnrs.fr

Sylvie Calabretto
LIRIS UMR 5205
INSA, Campus de la Doua
20, avenue Albert Einstein
69621 VILLEURBANNE CEDEX
FRANCE
Email: sylvie.calabretto@liris.cnrs.fr

***Abstract*—Audiovisual active reading is a commonly practised activity in different fields. This creative process consists in describing, criticizing and building an analysis during the reading of a document. More and more tools aim at providing ways to perform part of this activity, built on diferent description models. However, the whole activity is lacking a full support. In this paper, we present our work on a unified model for active reading of audiovisual documents, taking into account each part of this activity. Our approach is based on active reading data and activity modeling so as to propose a pertinent, generic and adaptive model.**

I. Introduction

Active reading of audiovisual document is a process consisting in reading and analysing a document, and re-using the object of the reading. Softwares currently well adress different parts of this process, such as annotating. However, the whole process, annotating, organizing annotations and building hypervideos from annotations, is currently not well supported. In this paper, we describe our work about a data model to support active reading of audiovisual documents. We first present our analysis of active reading as an activity and we propose a set of structures to build a model adapted to the audiovisual active reading process. We then present usefull transformations of this model, frequently performed by an active reader during his activity. We conclude with the implementation and validation of this model in the Advene software.

II. Audiovisual active reading

Active reading is a complex activity in which the reader reads a document in order to produce related content or to criticize it [1]. For example, taking notes, extracting parts of a document for later reuse or reflecting about the content while reading a document, are different kinds of active reading. From a general point of view, active reading is an activity based on a document, which aims to enrich the original document and/or produce a new document about it, while reading it.

In the case of audiovisual documents, active reading becomes an activity in which a user interacts with a movie to enrich it or create new content about it while visualizing it. The most common form of production during an audiovisual active reading session is annotation, but other products such as an abstract of the movie or an interactive table of content are also possible. This is an important and frequently performed process, either by film enthusiasts, researchers, teachers or critics.

Audiovisual active reading generaly means to annotate an audiovisual document. *Annotations* act as notes, and reminders for future exploitation. *Relations* between these annotations can be used as a base for video hyperlinking. An *annotation structure* composed by annotations and relations may be organized according to a *description model*, defining categories of annotations and relations. Finaly, presentations can be built based on the document and its annotations, such as summaries, subtitled presentations or interactive comments of the movie.

Active reading can take advantage of automated annotation process, based on feature extraction or shot detection for instance, however, our proposition in this paper focus on manual annotation of audiovisual documents as an active reading process.

Here is an example of audiovisual active reading session. Eric prepares an analysis of a stock market news. His first task consists in annotating the sequences of the news by determining the different topics. Then he decides to annotate the characters appearing on screen, and the objects, such as progression diagrams. At the end, he decides to build two summaries to browse the news by people appearance and by topic. Doing so, he notices that he can improve his work by separating the newscaster and his guests, to present a better summary which differenciates the questions and the answers in the news.

III. Our research problem

The process of active reading is often associated with annotating and visualizing a movie or hypervideo (an hypermedia built around a video [2]). It is an important and frequently performed action in various fields. Unfortunately, its technological support mainly addresses the aspect of annotation and structuration of annotations but ignores the presentation aspects.

978-0-7695-3454-1/08 $25.00 © 2008 IEEE
DOI 10.1109/ISM.2008.92

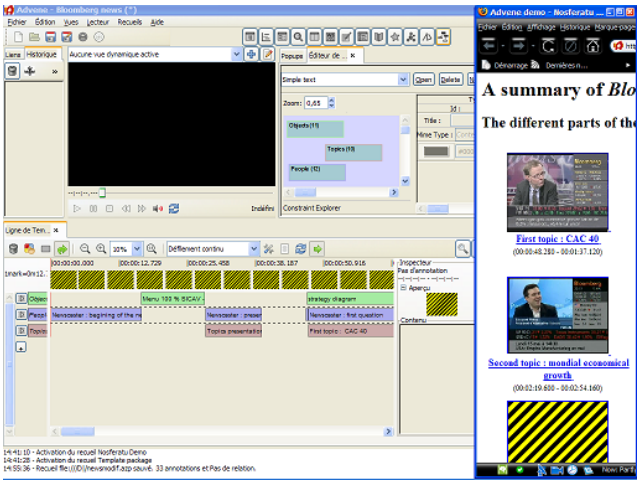

Fig. 1. Example of an active reading session with the Advene software: right next to the video player, a schema displaying three annotation types (categories) used to annotate the news. At the bottom of the screen, a timeline, displaying annotations according to the time of the video and the type they belong to in the schema. At the right of the screen, a summary view of the news, linked to the annotations and to the video.

Our work is to adress the whole activity of active reading. It is situated in the context of the Advene project [http://liris.cnrs.fr/advene], which aims at addressing the problem of annotating audiovisual material for creative distribution on the internet.

Our aim is to create and propose tools and models to support the active reader during his activity, according to an analysis of the actual practices of active reading. The first step toward this is the understanding of the different phases, mechanisms and operations belonging to active reading. This analysis helps to define criteria for supporting an active reader correctly.

For all the process of annotation and phases of active reading, the reader needs a highly flexible and transparent enough model which does not restrict him. This model needs to take into account each possibility of active reading, from annotations to presentations. Moreover, it is necessary to define basic transformations of the elements of the model, based upon the operations frequently performed during the active reading activity.

Finaly this model is the basis for the implementation of interfaces designed for active reading software allowing to manipulate and visualize each element of the model.

IV. Related works

Nowadays potential structure models, such as ontology languages and ontologies (owl [3], rdf, lom, mpeg-7, mpeg-21, etc.), thesauri (dublin core) and tagging systems [4] (flickr, youtube, etc.) either can express too few constraints on the data (tags) or are difficult to understand and manipulate by the reader (ontologies). Studying systems as Osiris [5], which gives the user the opportunity to annotate photos with terms or concepts taken from one or more ontologies, we noted that we could not use such structure for active reading. As active reading covers many domains, it would require an ontology for each of them. Moreover, ontological structures currently can not evolve properly at the time the reader uses them. Standards such as MPEG-7 [6] and MPEG-21 [7] formats specially designed to describe audiovisual documents and their content provide a large number of annotation structures but they are nevertheless far too rigid [8] to be applied for a dynamic process like active reading. There are tools to annotate audiovisual documents like Anvil [http://www.dfki.de/ kipp/anvil/], and tools to build hypervideo or hypermedia presentations like Hypercafe [9], but none actually propose to annotate and build hypervideo from annotations. Existing annotation applications, such as ANVIL or ELAN [http://www.lat-mpi.eu/tools/elan/] greatly restrict the user to establish annotation structures by offering only basic annotation templates. Lots of active reading tools concentrate only on annotating with predefined schemas or with only one type of annotation, like Madcow [10] which offers the possibility to annotate with multimedia content, and then visualizing the result. Other systems aim to address the entire process of audiovisual active reading. More particularly, the ongoing Advene project [http://liris.cnrs.fr/advene] at LIRIS is a project which mainly addresses the problem of annotating audiovisual material for creative distribution on the internet. The Advene software allows to annotate videos, create annotation schemas and build hypervideo centered on a movie. Its aim is to develop tools and models that help the active reader during his activity, based on thorough analysis of the actual practices of active reading. Other tools mainly care about video presentation: annotation process takes place before using these tools or just at the begining of the process and is lightly supported. Example of these visualization tools are youtube annotation tool [http://uk.youtube.com/t/annotations_about], Hyper-Hitchcock [11] or Hypercafe which provide interactive stories through movie vizualisation. Other montage tools also belong to this category and mainly base on video presentation rather than annotation and annotations organization.

V. Unified model for audiovisual active reading

Our aim is to support readers during their activity of audiovisual active reading, and one way to do so is to propose an appropriate data model. In order to build an adequate model, we first need to study audiovisual active reading as an activity.

A. Activity of audiovisual active reading

We define the process of audiovisual active reading partially based on Kuuti [12] hierarchical decomposition and on field investigations with teachers, film critics and amateurs, as the combination of five general types of actions. These actions are:

- *Annotation* which consists in adding new temporaly situated content to the document.
- *Classification* corresponding to the abstract organisation (categories) of the annotations.

- *Restructuration* including modification and linking of the annotations.
- *View building* consisting in creating and modifying ways to present the enriched document.
- *Browsing* which consists in navigating through the document.

The process of *Annotation* stands for a set of marks applied to a part of the document. A mark is generaly a temporal spatial reference that possibly contains information. It can be considered opened if the active reader needs to be reminded that additional information is required to complete it. The process of *Restructuration* consists in reorganising existing marks, and creating links between them. In this phase, the reader begins to organize his marks to prepare their future exploitation as ways to access the document. The process of *Classification* of already established marks consists in the categorization of these marks. It can be done by the reader either before or during the *Annotation* process or after the first phase of *Annotation* has been completed. This process of *Classification* shapes the future exploitation of the annotations. The *View Building* process is the phase where the reader exploits his previous work. It consists in building ways to present points of views about the document, based on the annotations and their structure, and which can be reused later. The process of *Browsing* allows the active reader to revisit the already annotated document. During this process, the reader uses the annotations and their presentation structures to modify his browsing experience. He can easily fall back from this process into the first three processes to improve his previous work.

B. Constraints on the data model

According to the previous analysis of active reading, we established some criterias that an appropriate data model should satisfy.

First, it is necessary that the content of annotations and relations is structured by the reader himself, to allow him to express every idea he wants. Thus, this structure needs to be build by the user, and needs to be free, flexible and scalable enough to adapt to the user mind.

Secondly, to classify these annotations, the active reader should be able to create abstract categories, corresponding to the concepts he wants to annotate. Thus, the structure needs an abstract level corresponding to personal practices of the active reading activity. As these concepts may be shared by other active readers, this structure should be easily shareable and understandable.

Building point of views is similar to selecting a subset of annotations and presenting them in a special way. The construction of a view results from the application of "queries" on the annotations of the document. That is why the structure need to be strong enough to be able to search through it and to do some reasoning on it.

Data structures are constantly evolving during the process of active reading, and it is therefore necessary to take this evolution possibility into account in the model. Indeed, it often happens that the reader change its annotations to finalize them with additional information he had not specified when he was annotating before. Due to the nature of the activity, these structures have to frequently evolve to take into account the context of the changes introduced by the continuation of the activity.

C. Audiovisual active reading data model

Based on the previous analysis we propose our model, composed of four main structuring levels hierarchically organized.

The structure of content provides the support necessary to establish an annotation. This level covers attributes, such as begin time, end time, and any attribute described by a name/value pair. Thus each element of the content of an annotation can be directly adressed or strained.

The annotation structure level offers concrete representations of organization structures, allowing the reader to create relations between annotations. This structure results from the instanciation of the classification structure level.

The classification structure level defines schemas, which are organizations of types of annotations, types of relations and constraints on them.

Finaly, the view structure level offers ways to create presentation structures based on the other structure levels.

The model resulting from these structures is composed of the following elements:

An *attribute* is a property of a temporal fragment description. It is always an instance of an attribute type. An attribute (at) possesses a name (n) and a value (v):

$$at = (n, v) \qquad Example : at_1 = (age, 30)$$

A fragment is a part of the audiovisual document, defined by two temporal marks *tb* et *te*.

$$f = (t_b, t_e)$$

$$Example : f = (0:00:00, 0:02:00)$$

An *annotation* is the description of a temporal fragment of the document. Annotations always are instances of an annotation type. An annotation (a) possesses an id, a fragment (f) and a list of attributes:

$$a =< id, f, at_i >, i \in [1..n]$$

$$Example : a_1 =< pers01, f_1,$$

$$[(lastname, Harker), (firstname, Jonhattan), (age, 30)] >$$

A *relation* is a link between annotations. Relations always are instances of a relation type. A relation (r) possesses an id, a list of attributes, and a list of linked annotations:

$$r =< id, at_i, a_j >, i \in [1..n], j \in [1..m]$$

$$Example : r1 =< rel01, [], [pers01, pers02] >$$

An *annotation structure* SA is a set of annotations and relations.

$$SA =< id, a_i, r_j >, i \in [1..n], j \in [1..m]$$

An *attribute type* AT defines a category of attributes. It possesses an id, a name (n), a type of value (vt), and a default value (dv):

$$AT =< id, n, tv, vd >$$

$$Example : AT_d =< id,' freetext', string,'' >$$

An attribute type AT is linked to his attributes (at) by a relation of instanciation I:

$$I_1 : AT \rightarrow at_i$$

An *annotation type* defines which attribute types compose their content. An annotation type A possesses an id, a name (n), a list of attribute types and a list of constraints (C).

$$A =< id, n, AT_i, C >$$

$$Example : A_d =< id, freeannotation, AT_d, \emptyset >$$

An annotation type A is linked to its annotations a by a relation of instanciation I:

$$I_2 : A \rightarrow a_i$$

A *relation type* R defines which object its relations will link. A relation type possesses an id, a name (n), a list of attribute types, and a list of linked object (O). O is a set of any element in the model:

$$R =< id, n, AT_i, O >$$

Concerning the particular case of relation types linking annotation types, we obtain:

$$R =< id, n, AT_i, A_j >, i \in [1..n], j \in [1..m]$$

A relation type A is linked to its relations (r) by a relation of instanciation I:

$$I_3 : R \rightarrow r_i$$

A *schema* S represents a model of description. It possesses an id, a name (n), a list of annotation types, a list of relation types, and a set of constraints C.

$$S =< id, n, A_i, R_j, C >$$

$$Example : S_d =< s0, freeschema, A_d, [], \emptyset >$$

A set of *constraints* C can be represented by a list of rules related to a schema or a type. An example for a rule can be that the set of all the annotations of a type should cover the whole duration of the document.

A *view* is a presentation of a set of other elements of the model, enriched by some external elements. A view v possesses an id, a name, a set of elements of support (ES) and a set of external elements (EE):

$$v =< id, n, ES_i, EE_j >$$

For example, a html view of a summary can be based on a type of annotation "sequences", and its rendering is defined by external elements applying to html pages.

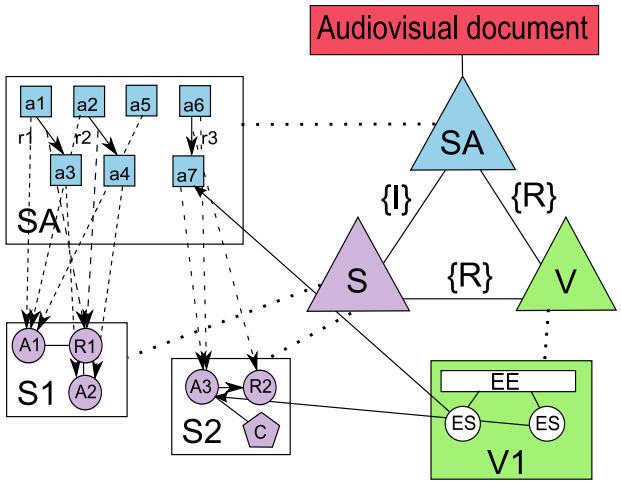

Fig. 2. Global view of the model, without the content structure detail. The three main poles of the model in the top right are detailed in views, schemas and annotation structure. The annotation structure, composed of annotations and relations is directly linked to the audiovisual document. Its elements are instances of annotation types and relation types composing the different schemas. Finaly, the view structure is based on external elements, schemas and the annotation structure.

Finaly, the whole active reading model ARM is composed of the annotation structure Sa, a set of schemas S, a set of views V and a set of relations R between these three elements:

$$ARM =< Sa, S_i, V_j, R_k >$$

The aim of this model is to take into account the criteria needed to better support active reading. These criterias are mainly transparency of the structures, structures close to the mind of the reader, an easy evolving of these structures and their ease of use. Moreover, this model not only supports the annotation phase of active reading, but also the categorization phase and the building of hyperdocuments based on the annotations through the management of views. This model is a unification of the different processes constituting the active reading activity.

VI. Transformations

As an active reader constantly evolves his description model, transformations of the elements of the models need to be studied. These transformations can be categorized in four main types:

- Transformations internal to an element type: for example, modifying an annotation, either by changing its fragment or its type.
- Transformations between concrete elements of the model: for example, transforming an annotation into a relation.
- Transformations between abstract elements of the model: for example, transforming an annotation type into a relation type.
- Mixed transformations: for example transforming an annotation into an annotation type.

These transformations can apply to the different elements identified before, such as attributes, annotations, relations, types, schemas, views or subset of the model. A transformation is an operation applied to one or more elements of the model to modify it and can impact other elements of the model. Any transformation applied to an abstract element has repercussions on the concrete elements corresponding to its instances.

We currently identified forty-seven transformations. Some are basic, such as the transformation which consists in changing the type of an annotation. For this transformation, the user needs to choose the annotation to modify, its new type and the new values for the attributes of the annotation if the models of content of the two types do not match. This transformation produces a new annotation, instance of the choosen type. An example illustrating this transformation is the transformation of an annotation of type "character" to an annotation of type "newscaster", precising that the character annotated is in fact the newsreader.

Some transformations are a little more complex, such as the one which transforms an annotation into a relation between two annotations. For this transformation, the user needs to choose the annotation to modify, the two annotations for the relation or the type in which two new annotations will be created, and the relation type for the new relation. It will produce the relation and eventually two new annotations. Here is a little example to illustrate this transformation. An active reader has annotated "important sequences" in a film, as well as "characters" appearance. He realizes that for future uses, describing a murder scene in an annotation of type "important sequence" does not allow him to easily find the characters involved in the scene. He decides then to transform this annotation into a relation between the two "character" annotations corresponding to the appearance of the victim and the murderer.

As these transformations are frequently used by an active reader, it is necessary to implement them in active reading tools. Instrumenting these transformations allows a better support of the user, as it gains time for the active reader.

VII. Implementation and validation

This model and some of the transformations presented above, mainly related to annotations and transformations, have been implemented into the Advene software. The current version of Advene supports only a lightened version of the model, as the constraints and attributes are only partially managed. Types of attribute do not actually exist in Advene model, and thus the structure of content is not linked to the abstract structure. However, the main structure (schemas, types, annotations, relations, views) is currently functional, and nearly all transformations regarding these objects too. For example, it is currently possible to use a transformation wich automatically creates an html view presenting an annotation type. The Advene software is currently used by different kind of active reader, such as researcher, film enthousiasts, critics

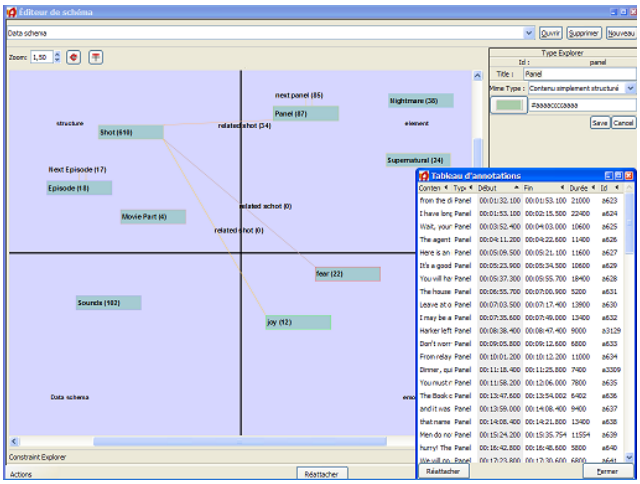

Fig. 3. Schema editor in the Advene software, displaying four schemas with their annotation and relation types. In the bottom right of the screen, a view of the annotations corresponding to the "Panel" type. In the top right of the editor, an explorer displaying details about the selected "Panel" type.

and teachers, proving that this model is able to support this activity.

To manipulate the abstract part of the model, we built a schema editor and integrated it in Advene. This editor presents schemas in a graphical way for a better representation of the manipulated structure. It allows the user to build a schema and to populate it with annotations and relations types. It offers the possibility to explore a type to find the corresponding instances, and allows schema comparison by displaying multiple schemas. Moreover, it implements some of the transformations linked to the abstract elements, as for example moving a type from a schema to another, or creating a presentation view based on a type. This schema editor has already been tested by two different volunteers, during two active reading sessions of two hours long. The contribution of this tool to the understanding of the manipulated structure and to the ease of use of the software has been highlighted by these users, which have really appreciated using this tool.

To test the ease of use and the flexibility of our model, we currently plan to organize an active reading session of one hour with five to ten participants, who will have to annotate and present a short movie of five minutes long. A first schema of description will be proposed to them, and they will be autorized to modifie or complete it to achieve their goal. A tool tracing their activity will help us to determine if they frequently used the implemented transformations, and they will have to answer a questionnaire at the end of the session to give their feedback.

VIII. Conclusion

We presented in this article our unified model of active reading, which allows to handle and support each process part of the audiovisual active reading activity. We first presented some works related to active reading and structure models used in this kind of activity. Then we presented our analysis

of active reading as an activity made of five processes, acting at different levels and different parts of the enriched docuemnt. We proposed four types of structure to support these different processes, and then we presented our unified model taking these structures into account. This model aims to be enough clear, transparent and easy to use to support active reading, but nontheless complete and strong enough to support each phase of the active reading activity, from annotations to hyperdocument building. Moreover, we studied every possible transformations on the model. A lightened version of this model and some of the transformations are currently implemented and used in the Advene software, as well as an interface to manipulate schemas and abstract types in a graphical way. Full support of this model will be made in version 2.0 of the Advene model, and the other transformations will be implemented as well.

As our research work aims at supporting an active reader during his activity, we also concentrate on the use of reflexive traces to help the user. These traces rely on the unified model presented in this article as well as on our study on the active reading activity, and are currently a work in progress.

REFERENCES

[1] B. Schilit, G. Golovchinsky, and M. Price, "Beyond paper: supporting active reading with free form digital ink annotations." in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1998.

[2] O. Aubert and Y. Prié, "Advene: active reading through hypervideo," in *ACM Hypertext'05*, Salzburg, Austria, Sep 2005, pp. 235–244. [Online]. Available: http://portal.acm.org/citation.cfm?id=1083405

[3] M. Dean and G. Schreiber, "OWL Web Ontology Language," W3C Recommendation, http://www.w3.org/TR/owl-ref/, Feb. 2004.

[4] C. Marlow, M. Naaman, D. Boyd, and M. Davis, "Ht06, tagging paper, taxonomy, flickr, academic article, to read," in *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*. New York, NY, USA: ACM Press, 2006, pp. 31–40.

[5] Y. Nizon and F. Trichet, "Une plateforme de dveloppement d'espaces webs smantiques communautaires ddis au partage de ressources multi-mdia," in *Ingnierie des Connaissances 2007, Grenoble, July 2007*, Jul. 2007, pp. 209–224.

[6] F. Nack and A. T. Lindsay, "Everything you wanted to know about mpeg-7: Part 1," *IEEE MultiMedia*, vol. 06, no. 3, pp. 65–77, 1999.

[7] I. Burnett, R. V. de Walle, K. Hill, J. Bormans, and F. Pereira, "Mpeg-21: Goals and achievements," *IEEE MultiMedia*, vol. 10, no. 4, pp. 60–70, 2003.

[8] J. van Ossenbruggen, F. Nack, and L. Hardman, "That obscure object of desire: Multimedia metadata on the web, part 1," *IEEE MultiMedia*, vol. 11, no. 4, pp. 38–48, 2004.

[9] N. N. Sawhney, D. Balcom, and I. E. Smith, "HyperCafe: Narrative and Aesthetic Properties of Hypervideo," in *UK Conference on Hypertext*, Bethesda, Maryland, United States, 1996, pp. 1–10. [Online]. Available: citeseer.nj.nec.com/53234.html

[10] P. Bottoni, R. Civica, S. Levialdi, L. Orso, E. Panizzi, and R. Trinchese, "Madcow: a multimedia digital annotation system," in *AVI '04: Proceedings of the working conference on Advanced visual interfaces*. New York, NY, USA: ACM, 2004, pp. 55–62.

[11] F. Shipman, A. Girgensohn, and L. Wilcox, "Combining spatial and navigational structure in the hyper-hitchcock hypervideo editor," in *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, 2003, pp. 124–125.

[12] K. Kuutti, "Activity theory as a potential framework for human-computer interaction research," pp. 17–44, 1995.