Helping users to recover from interruptions in audiovisual active reading: An interface for reflexive use of traces.

Bertrand RICHARD
LIRIS UMR 5205 CNRS
Universit de Lyon
43 Boulevard du 11 Novembre 1918
69622 Villeurbanne Cedex
France
bertrand.richard@liris.cnrs.fr

Yannick PRIE
LIRIS UMR 5205 CNRS
Universit de Lyon
43 Boulevard du 11 Novembre 1918
69622 Villeurbanne Cedex
France
yannick.prie@liris.cnrs.fr

Sylvie CALABRETTO
LIRIS UMR 5205 CNRS
Universit de Lyon
43 Boulevard du 11 Novembre 1918
69622 Villeurbanne Cedex
France
sylvie.calabretto@liris.cnrs.fr

ABSTRACT
Active reading is an activity commonly practised in the research, education and media. It is a creative process in which a reader manually describes, critics, and builds an analysis on the object of his reading. Active reading of audiovisual documents is a challenge because of the inherent temporality of the video: it notably implies managing attentional conflicts between the temporality of the activity and that of the document. Such conflicts often lead to interruptions of the tasks the user was carrying. In this paper, we present our ongoing work on active reading of audiovisual documents towards activity recovery and our interface for visualization and reflexive use of traces. Our approach is based on active reading data and activity modeling so as to present the user with real time traces of his activity, and means to interact with them.

Author Keywords
Timeline, traces, interactions, interruptions

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: Graphical user interfaces (GUI)

INTRODUCTION
More and more digital audiovisual documents are available on personal computers and content servers such as YouTube. This massive diffusion implies more and more personal activities based on these documents, such as active reading. Active reading is a complex activity ranging from video annotation to hypermedia authoring that needs to be correctly supported by software applications. Some tools exist for audiovisual documents annotation and indexing, others allow to browse video interactively or to design such interactivity (hypervideo building), such as Elan [26], Anvil [14], Diver [21] or Annodex [22].

Such tools generally provide the user with a lot of functionalities related to metadata and fine grained video document management and navigation, which require the whole user attention. But the video medium possess its own temporality, which often interferes with the temporality of the active reading activity, and users have difficulties managing their active reading tasks with as little interruptions as possible. In our approach for answering this problem we propose to provide the user with reflexive activity traces to manage his own activity. We accordingly define an interface to visualize these activity traces and to offer means to interact with them in order to recover from an interruption.

In this article, we introduce our work on this interface for recovering from interruptions in audiovisual active reading activity. The first part of the article is dedicated to the presentation of the context of active reading and the research problem we want to tackle. The second part deals with related works on interruptions, reflexive traces and interfaces for activity managing in Human Computer Interaction. The third part presents our own trace-based approach, the implementation in the Advene software and preliminary validation studies.

CONTEXT OF AUDIOVISUAL ACTIVE READING
Active reading is a complex activity in which the reader reads a document in order to produce some related document or to criticize it [25]. 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 while reading it.

Applied to audiovisual documents, active reading is an ac-

Submitted for review to IUI 2010.

tivity in which a user interacts with a movie and produces new content about it while visualizing it [2]. Annotations are the most common forms of production on an audiovisual document, but an abstract of the movie or an interactive hypervideo are other possible forms. The process of active reading of audiovisual documents is important and frequently performed, either by researchers, critics or film enthusiasts. Generally, during an active reading session, the reader annotates his audiovisual document. Annotations acts as notes, and indexes for future use. Relations between annotations can be a source for video hyperlinking. These annotations and relations may be organized according to a model (schema), which can define categories of annotations and relations. An active reader can also build presentations based on his annotations, such as a summary of the content of the audiovisual document or a subtitled presentation. An important remark is that if of course active reading can take advantage of automated annotation proposals (based on feature extraction for instance), we focus in this paper on *manual* annotation and human interaction with annotations and movies.

Here is an example of an actual active reading session: Martine prepares a film analysis to present it to her students. She visualizes the movie and begins her analysis based on previous annotations built by a friend, containing the description of all the shots of the film. Her first task consists in enriching this first structure by annotating the sequences of the film. Once this technical part is done, she underlines the different locations in the movie (street, nature, city, school) and the characters (the main character Billy, his family...). At the end, she decides to build some presentations to show the different aspects of the movie in an interactive way. For example, she builds a presentation summary for navigating through the different sequences of the film, and an other one to loop through the appearances of Billy in the movie. Doing so, she notices that she can improve her work by modifying the description of the characters, as some of them only appear in precise locations. She then decides to reoganize her description schemas, re-classifying her “character” annotations in three distinct categories: “Billy’s family”, “school acquaintances” and “other characters”.

Lots of active reading tools concentrate only on annotating with predefined schemas or with only one annotation type, and then visualizing the result. However some systems, like Anvil [http://www.dfki.de/ kipp/anvil/], Elan, or Advene aims to adress the entire process of audiovisual active reading. More particularly, the ongoing Advene project [3] at LIRIS is a project which mainly addresses the problem of annotating audiovisual material for creative distribution on the internet [http://liris.cnrs.fr/advene]. 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.

RESEARCH GOALS

The activity of audiovisual active reading we just presented is indeed very complex and requires a lot of attentionnal resources for the user to keep in mind his goals. Indeed, the user has to read and navigate through his document, annotate what he finds interresting in it, change his categories according to new ideas, modify annotations accordingly, define presentations of the annotations such as hypervideos, etc.

Audiovisual active reading session is then not a straightforward process, rather a process that encounters lots of interruptions related 1/ to the creative nature of active reading (the user has the right to change his mind), 2/ to the special nature of audiovisual documents, which have their own temporality. Indeed, such documents possess their own time flow, which interferes with that of the user task and activity.

For instance, suddenly deciding to split a ”characters” category into three categories can occur at any time during the viewing and the annotation of the movie, leading to focus on the splitting task rather than on the annotating task that was carried when the decision occurred. Another example is an annotation task on ”characters” of film (watching the movie, annotating the characters) during which the user would suddenly notice an interesting reference to another movie, a reference he would have to annotate as soon as possible no to loose this important idea. But creating an annotation to describe the movie reference, if it took more that a minute would certainly make him forget what he was doing at first, while the movie would have continued its playing for some time, unwatched.

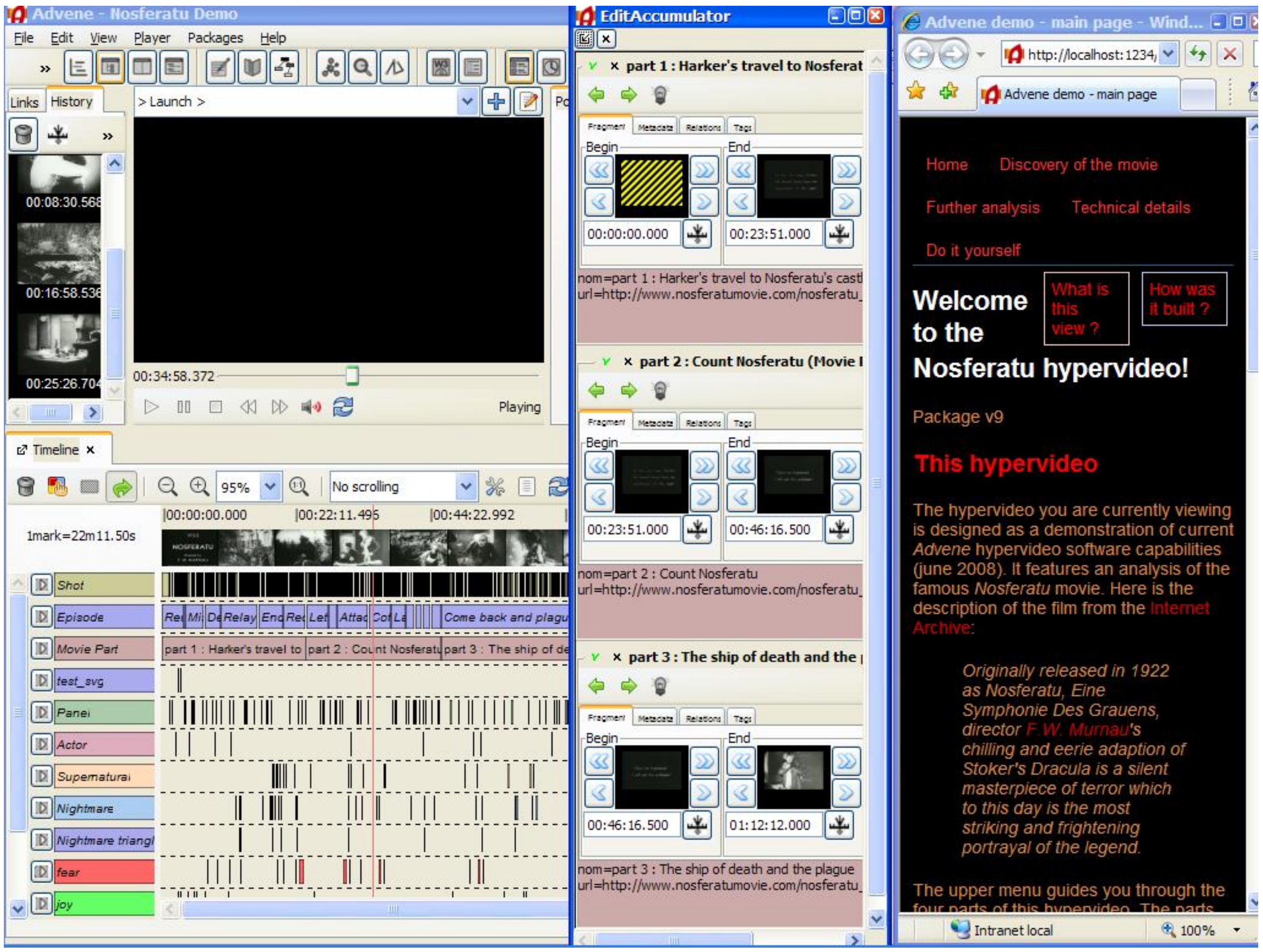
So, many flows such as activity flows, audiovisual stream flows can sometimes enter in a conflicting state, leading to a problem for the active reader: can he follow two flows at the same time, or should he follow one, interrupting the other flow and the related task he was performing? How can he resume a task he has lost attention to?

The user has indeed to concentrate on many different flows to complete his work, but as a human, he only has one attention. So he needs to share his attentionnal resources between his document and his task(s) and interruptions occur inevitably. Active reading tools should help him carrying his active reading activity 1/ by reducing the risk of being interrupted with careful application design, 2/ by facilitating attention and task recovery after interruptions.

So our main concern in this article will concentrate on the possibility to recover from interruptions through a dedicated interface and the use of reflexive traces.

RELATED WORKS

Traces are frequently used in our common computer activities, such as the navigation history of our internet browsers, or the list of recent documents used in our operating system. They do not support our activity at every step, but when we can not remember which file we were working on or what website we were visiting, these traces are there to help us.



part 1 : Harker's travel to Nosferat

00:00:00.000 00:23:51.000

nom=part 1 : Harker's travel to Nosferatu's castl
url=http://www.nosferatumovie.com/nosferatu_

part 2 : Count Nosferatu (Movie I

00:23:51.000 00:46:16.500

nom=part 2 : Count Nosferatu
url=http://www.nosferatumovie.com/nosferatu_

part 3 : The ship of death and the

00:46:16.500 01:12:12.000

nom=part 3 : The ship of death and the plague
url=http://www.nosferatumovie.com/nosferatu_

part 1 : Harker's travel to | part 2 : Count Nosferatu | part 3 : The ship of de

Come back and plagu

Welcome to the Nosferatu hypervideo!

Package v9

This hypervideo

The hypervideo you are currently viewing is designed as a demonstration of current *Advene* hypervideo software capabilities (june 2008). It features an analysis of the famous *Nosferatu* movie. Here is the description of the film from the Internet Archive:

Originally released in 1922 as Nosferatu, Eine Symphonie Des Grauens, director F. W. Murnau's chilling and eerie adaption of Stoker's Dracula is a silent masterpiece of terror which to this day is the most striking and frightening portrayal of the legend.

The upper menu guides you through the four parts of this hypervideo. The parts

Figure 1. Example of the Advene software for audiovisual active reading: right next to the video player, three annotations opened in edition in the "editAccumulator". At the bottom of the screen, a timeline, a classical presentation view, 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 welcome view of the hypervideo build on the movie.

Our aim being to support readers during their activity of audiovisual active reading, we find it promising to show them their activity traces, so as to help them understand and remember what they have done. Presenting a user with a trace allows him to have a reflexive approach of his activity, and has the advantage of not being intrusive.

Traces are currently being worked on in different research teams, as for example in [19]. In this paper, the authors present an application independant system to display the history of operations executed by a user, to help him to visualize his activity and to support him during it. In [16], the author presents the principles of trace-based system, based on trace modeling, collecting, transformations and presentation. Traces of interactions are also used in to create automated assistants that help users in their task [8].

Attentional resources of an active reader must be shared between what is happening in the audiovisual document and the actions carried on, which may lead to a loss of attention. Different works try to determine the appropriate moments to interrupt a task, depending on the current activity, on the interrupting task and on the cognitive load of the subject [11, 12]. Others rely on a modeling of the activity to save its context to be able to resume the ongoing work after a break [5]. The analysis of users' activity organization [4] is a good way to understand how we can support task switching [6, 9] and interruptions.

Interruptions during an activity are a current topic in the Human-Computer Interactions community, in works such as the studies of McFarlane [18] to compare interruption methods in HCI. Other studies [1, 11] try to determine what is the best moment towards the beginning, middle, or end of a task to interrupt the user when it is necessary.

Such studies all rely in one way or another on *task or activity modeling*, so as to be able to describe what the user is doing. There are actually three categories of tasks models that can be used for describing interactions and tasks in an activity. We can classify them as following: behavioral models, cognitive models and activity theory. *Behavioral models* of describe the interaction in a "behaviorist" way, and are mainly concentrated on graphical interactions on the task level and operation level. These tasks models are logical descriptions of actions needed to achieve a goal. They are functional descriptions of the activities. Examples of these models are UAN [10] and KLM [7], which describe interactions between a user and a system through its tasks. *Cognitive models* describe cognitive processes at a very fine level. "Goals, Operators, Methods, Selection Rules"[13] and "Tasks-Actions Grammars" [20] belong to this type of models. *Activity theory-based models* focus on the Leontjev activity theory [17]. These models are purely descriptive and generaly at high level. But different works based on it tend to be less descriptive and more functionnal, as the hierarchical decomposition of an activity presented by Kuuti [15] and the theory of instrumented activity [23] which defines a mediation of the activity through an instrument. We relied on those works to build our own modeling of the activity.

THE ACTIVITY MODEL

In order to build an activity trace of active reading, based on the previous context and related works, we need to consider and create a model of our activity at two distinct levels. 1/ We establish an object model of the data manipulated during the activity, so as to offer a structure for supporting the production of the activity, and 2/ we build a model of the activity, which will be a model of the traces we will collect. The models we present below do not aspire to universality, but as our approach and our software are really generic, they cover a great majority of current practices of active reading.

Data model for audiovisual active reading

What we want to do is to instrument active reading software to support the user activity. That is why we work on different parts of this activity, from data modeling to trace-based systems. The first step in our proposal has been to choose a global active reading oriented data model.

When dealing with annotations, lots of tools are based on knowledge models such as ontologies to structure the information created. Some other tools are build on standards such as MPEG-7 [27] and MPEG-21 formats specially designed to describe audiovisual documents and their content. These models are great to annotate videos, but not to build presentations based on these annotations. To take this dimension into account, the presentation part needs to be integrated in the model used during the activity of active reading. The model we choose to use, adapted to audiovisual active reading and presented in [24] is based on three main parts: generic schemas, specific objects and presentation (as shown in Figure 2).

The first part of the model is the *annotation structure* and represents concrete information a user manipulates during his activity. It is composed of annotations and relations. Annotations possess a temporal fragment, which is basically composed of two temporal marks (begin and end), and a content. Relations possess a content and a list of annotations they link. Moreover, relations and annotations always are instances of relation or annotation types.

The second part of the model is composed by *schemas*. Each schema regroups a set of annotation types and relation types. An annotation type defines a class of annotations, a set of attributes it can have and constraints on the value of these attributes. A relation type defines a class of relations, as for annotation types. Moreover, it specifies which kind of annotations can be linked by the relations.

The third part of the model corresponds to the presentation and contains *views*. A view is a presentation of a set of annotations, relations, schemas, types and elements external to the model according to a view definition. For example, it can be a HTML page presenting a summary of a movie, based on annotations of a "scene" type.

Activity modeling of audiovisual active reading

To support the active reading activity with traces we had to build an active reading activity model, based on the manipu-

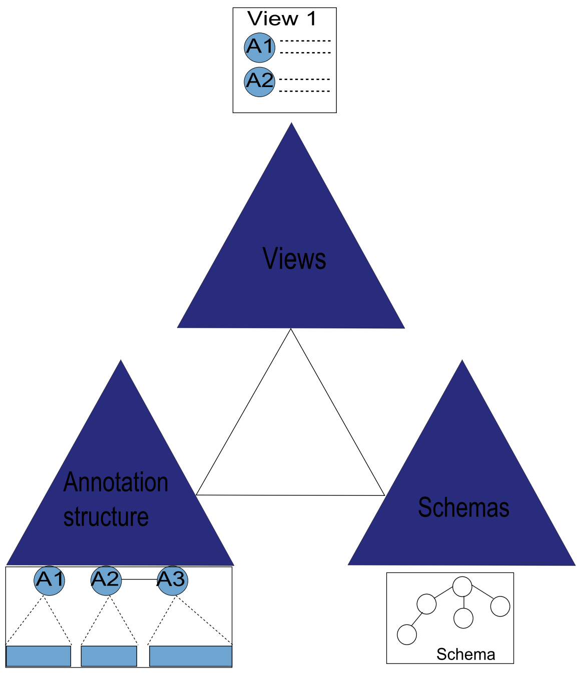


Figure 2. Global view of the active reading data model, presenting "Annotation Strucutre", "Schemas" and "Views".

lation of the previous data model.

According to the theory of activity, an activity can be decomposed into actions and sub-actions, and actions can be further decomposed into operations. *Actions* are directed towards a goal. *Operation* are acts performed reliably and quickly in an almost unconscious way. Operation is an elementary unit needed to achieve the goal of an action.

Our activity model of active reading is decomposed into the following five actions. *Navigation* gathers every act relative to the movie player. *Annotation* consists in adding new annotations to the document. *Classification* corresponds to the creation and manipulation of the schemas and annotation and relation types. *Restructuration* includes relation related operations and annotation modifications. Finally *View building* consists in creating and modifying presentation views.

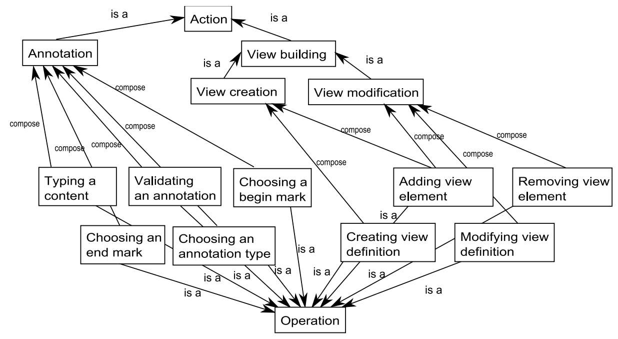


Figure 3. Excerpt of the active reading activity model. "Annotation" and "View building" actions, linked to their respective operations.

Each of these actions can be decomposed into operations, which are described by *the object of the model it applies to* and *what kind of link to the document it requires to be performed*. We built an ontology as the model of the activity, with action and operation concept classes. Each action is linked to operations by a relation of composition, and each operation possesses the following attributes: duration, object and attentionnal link. This attentional link is then used to determined if two actions can happen simultaneously or not, and may also help to determine if the moment currently playing in the movie is important or not to the user. We thus obtained an ontology of active reading with seventy-four concepts, partly presented in Figure 3.

This ontology allows us to consider the entire activity of active reading in a more formal way, which is going to allow us to build a consistant trace during this activity.

PRINCIPLES FOR RESUMING

In the global case

Resuming an activity is an action during which the user retrieves the context of his interrupted activity, to get back in a state close to the one preceeding the interruption. Recovering from an interruption mainly consists in restoring the activity as it was before the interruption, or in the worst case, in providing ways to remember its context.

The *context* of an activity can be represented by the state of a system composed by the activity, the user and the object of the activity. We can differentiate two kinds of context :

- the global context of the activity, which specifies what the user was doing in relation to the overall activity.
- the local context of the activity, which specifies, in relation to the ongoing action in the activity, the operations previously carried out and the objects involved in these operations. A local context takes sense in relation to a global context, which allows to situate it in the flow of the activity.

This distinction between contexts is necessary because of the variety of possible interruptions for a given activity. Thus, it is necessary to differentiate the mechanisms of resuming an activity depending on the type of interruption that occurred during it.

Moreover, these interruptions can be classified according to their duration and the fact that the interruption is voluntary or not:

- Very long stop: several months, voluntary (vls)
- Long stop: between the end of a working session and the beginning of a following, voluntary (ls)
- Short stop: coffee break, external interrupt, may be voluntary or involuntary (ss)
- Problem during the activity: memory problem of the subject, involuntary (ap)

At last, when practicing an activity, the subjet often creates and uses behavioral *patterns*. These patterns represent the way a user is doing a particular task. For example, a film critic may often begin his work by annotating the different scenes of the movie, and then analyses each of them separatly. Detecting these patterns may help to determine what the user was planning to do before his interruption.

The basis for resuming an activity is constituted of the *entries* that the subject can create during it. The problem of resuming an activity is therefore the problem of how to find, manage and use these entries to enable the subject to remember his context of work. These entries are constitued by each operations made by the user and each objects he creates or manipulates.

So, the first step to resume an interrupted activity is to recover the context of this activity. For that, a user needs to visualize what he was doing right before the interruption, and may need to recontextualize these last actions in the more global context of the whole activity he was conducting so far.

The second step is to continue the interrupted process. This can be helped in many ways, giving means to the user to get back to his work from the interface used to visualize the trace.

Applied to audiovisual active reading

The requirements for resuming an activity of active reading are related to the contexts previously defined. Thus, we need to find information about the context among the more or less recently (depending on the type of interruption) created entries.

In the case of short interruptions (ss / ap), it is necessary to include in this context:

- the last operations
- the last access to the audiovisual document
- the latest objects manipulated
- the recent changes to these objects

In the event of a resumption after a longer period (ls / vls), information regarding the conduct of the activity since its beginning is necessary in order to define the global context before the interruption, in addition to the information previously cited - to define the local context of the activity.

To enable the subject remembering the context of his activity, it is necessary in the case of a long interruption, to show him information about the global context of its activity, so that he could remember the pattern he eventually used during his practice. Once this pattern recalled from the various steps in the global context, it will be easier for him to resume his work by studying the local context of his past activity.

In the case of active reading, the context of the activity includes:

- the state of the audiovisual document at a time t,
- the operations as defined in the model of the activity of active reading,
- and the manipulated objects as defined in this model.

These different information can be found in the user entries, it is therefore necessary to trace them to keep a complete and detailed history, and to be able to display these information when the user need them.

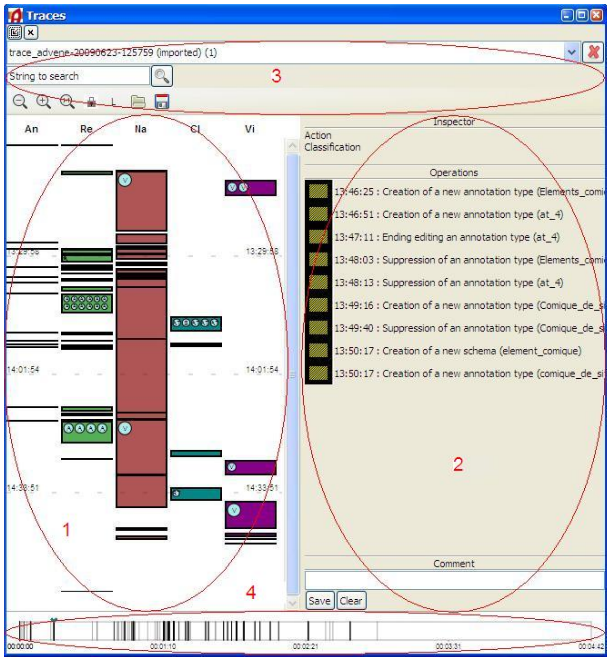
OUR INTERFACE TO RECOVER FROM INTERRUPTION

The interface that we are going to present in this section can be divided in four parts (see Figure 4):

- the activity timeline, displaying actions of the activity according to the time of this activity
- the inspector, displaying either the operations composing a selected action, or the operations executed on the selected item
- the document radar, displaying the moment in the audiovisual document when the operation was carried on

The aim of this interface is not to be able to manipulate traces, but rather to offer a way to visualize traces, and to act on the document and the activity from these traces.

We are now going to describe and explain the role of each of these componants.



```
13:46:25 : Creation of a new annotation type (Elements_comi
13:46:51 : Creation of a new annotation type (at_4)
13:47:11 : Ending editing an annotation type (at_4)
13:48:03 : Suppression of an annotation type (Elements_comi
13:48:13 : Suppression of an annotation type (at_4)
13:49:16 : Creation of a new annotation type (Comique_de_s
13:49:40 : Suppression of an annotation type (Comique_de_s
13:50:17 : Creation of a new schema (element_comique)
13:50:17 : Creation of a new annotation type (comique_de_si
```

Figure 4. The global view of our interface: in the left part, an activity timeline (1); in the right part, an inspector (2), and a radar (4) in the bottom. The last part (3) is a classical toolbox, offering zooming, loading, exporting and searching features.

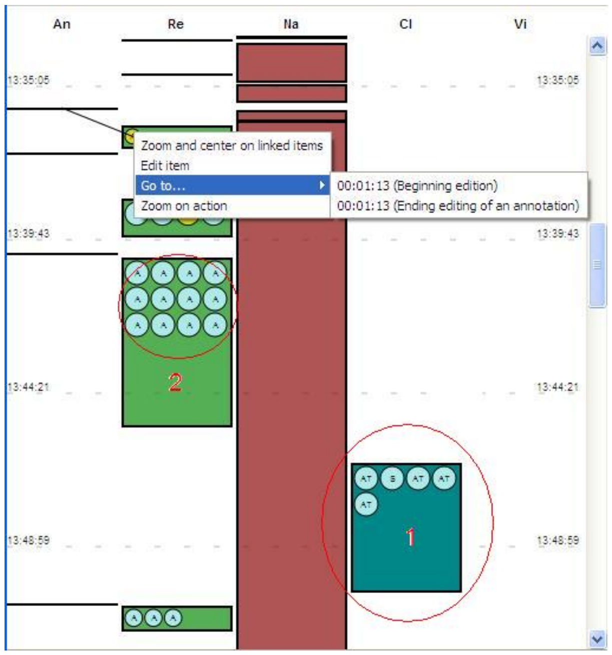

An
Re
Na
Cl
Vi
13:35:05
13:39:43
13:44:21
13:48:59
Zoom and center on linked items
Edit item
Go to...
Zoom on action
00:01:13 (Beginning edition)
00:01:13 (Ending editing of an annotation)
2
1

Figure 5. The trace timeline component: actions (1) are displayed according to the activity time and the activity model, and contain objects (2) which have been manipulated through them.

The activity timeline component

This is the main component of the interface. Its goal is to display the actions performed by the user according to the time of the activity. They are also classified using the type of this action, as defined in the activity model. We choose a timeline object to represent the activity flow because it allows the user to view the entire activity as well as to zoom on a particular moment of it, and also because it is a frequently used component within audiovisual softwares, generally to represent the document flow. So the component looks like a classical vertical timeline, in which each action is represented by a rectangle. Each of these rectangle can include circles, representing the objects used during the action. The first letter of the type of the object is also included, to allow a faster identification of the item, as shown in Figure 5. For example, if the user edits an annotation, a rectangle will be created in the *re* (restructuration) column, and will include a circle with an *A* representing the modified annotation.

When moving the mouse cursor over an action or an object, it automatically displays information in the two other component. Moreover, when moving the cursor over an object, it displays links between the different occurencies of this object in the timeline, thus allowing the user to quickly identify the different phases of his activity in which he manipulated this object. For the conveniency of the user, these links can be displayed starting from the pointed item to each other, or respecting the sequentiality of the activity.

The particularities of this interface takes place in the actions offered to the user. From this interface, he can:

- re-edit an already manipulated object
- re-create a deleted item
- visualize the different actions he performed on an object, and materialise links between the appearances of this item
- go to the moment in the audiovisual document which was playing when he performed a given operation

The aim of these different opportunities given to the user is first to allow him to get his context back, and then to give him easy ways to resume his activity. The links materialised between the different appearences of the selected item help the user to identify the different moment when he worked on it. The component offers two ways to build these links: the first one create links centered on the selected item, the second one displays the links in a temporally sequential way.

Looking at the global timeline of his activity, the user can easily identify what were the different steps of his work, to remember globally what he has done so far. Then, looking to the last actions details, he can find information about his *local context*, and ways to start again from where he stopped. These information are mainly represented by the objects he manipulated, and the detail of these manipulations which are available through the second component of the interface.

The inspector component

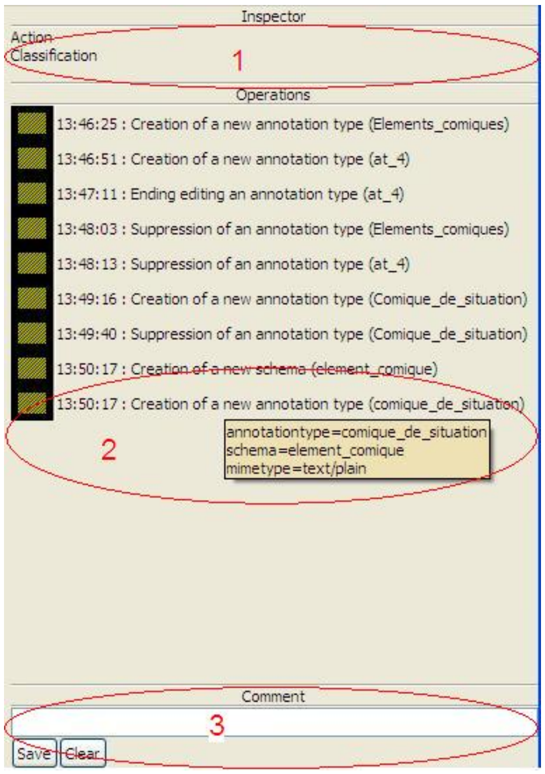
This component displays information about the object or action selected in the timeline. The information are displayed in two parts (see Figure 6):

- generic information about the object or actions
- information about the operations conducted on the object or within the action

In the first category, we find the id and the kind of object it is according to the data model (which is directly used and manipulated by the user during the activity, so this information is meaningful to him), and the object it is linked to; for example, for an annotation about characters in a movie, we would find that it is an annotation, with id *a1* for example, and of type *characters*.

For an action, we would find the same kind of information to identify what type of action it is according to the activity model (annotation, restructuration, classification, view buidling or navigation).

The second category of information displayed concerns the operations conducted on the selected object or included in the selected action. In this part, we find each operation described by the time at which it occured, a snapshot of the movie at this moment, its name, and the details of the operation in popup, in order to take less space on the screen. For example in Figure 6, we can see the details of a *classification* action. The different operations composing the action listed below are mainly creation and deletion of annotation types, and information concerning the objects edited are displayed in the popup window.



Inspector

Action
Classification

Operations

13:46:25 : Creation of a new annotation type (Elements_comiques)
13:46:51 : Creation of a new annotation type (at_4)
13:47:11 : Ending editing an annotation type (at_4)
13:48:03 : Suppression of an annotation type (Elements_comiques)
13:48:13 : Suppression of an annotation type (at_4)
13:49:16 : Creation of a new annotation type (Comique_de_situation)
13:49:40 : Suppression of an annotation type (Comique_de_situation)
13:50:17 : Creation of a new schema (element_comique)
13:50:17 : Creation of a new annotation type (comique_de_situation)

annotationtype=comique_de_situation
schema=element_comique
mimetype=text/plain

Comment

Save Clear

Figure 6. The inspector component, displaying general information (1) about the selected action or item, and details concerning this item (2). The user can let a comment on his own actions using the bottom field (3).

Thus, this component is principaly designed to remember the local context of the actions, offering details on the operations and the manipulated objects. The user can reedit an object from here, just by double-clicking on the corresponding line.

The document radar component

This is the third component and his goal is to rapidly show the moment of the movie which were displayed when an operation occured. The component is a basic rectangle representing the duration of the movie (some timecodes are indicated under the rectangle, as shown in Figure 7). In this rectangle are bars, representing the different times accessed during the activity. The contrast of the bars indicates the quantity of access made to the timecode (the darker is the bar, the more it has been accessed).

Moreover, when selecting an object or an action in the main component, its *movie times* are stated by a little *v* mark above the bar, in the color of the action / object. This allows the user to quickly identify at what times of the movie he manipulated an object or conducted an action. This information may or may not be relevant in certain situations; indeed, a user could have created an annotation because his saw something interesting in the movie, or he may just have let the movie play and continue working on some annotations. In the first case, the information concerning the movie access time is relevant and can be helpful to recover information about the goal the user had in mind at this time, in the second case the information is no more than a noise.

At last, the user can access any time of the document just by clicking in the rectangle, thus allowing to quickly go back to some point of interest in the movie. This component mainly help to contextualize the actions according to the movie, in a global way.

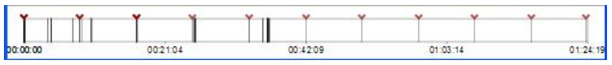


Figure 7. The radar component presents the temporality of the document. Each bar represents a moment in it, corresponding to an operation occured.

FIELD STUDY

Based on the models previously described, we implemented a tracking system in the Advene software (an opensource audiovisual active reading softawre), to record each of the user operations. The events are recorded and constitute a primary trace of his activity. We already tested this tracking system in a preliminary study with five voluntary users, which helped us validate our collecting model by verifying that we were able to recall each important action or operation the user had done during his session.

Goal and subjects

We implemented our activity timeline to visualize these traces and tested it with ten volunteers who had preliminary knowledge about the software. These subjects were given an individual active watching task of two hours long, on an already prepared video. Each subject had to complete annotations about comic scenes in Monty Pythons' ¡¡ Holy Grail ¿¿ film, and to build a presentation of there analysis.

So the task consisted in a classical active reading session, during which we interrupted the subjects. Duration of these interruptions varied between less than one minute to nearly ten minutes. Interruptions were based on tasks requiring the full attention of the subject, as for example, information retrieval on the internet about the Monthy Pythons. The interruptions used during the activity were:

- quizz about Monty Python
- little mathematical games
- coffee break

In one case, we introduced a longer interruption (lunch break) to experience the particular case of activity recovery at the begining of a work session.

The evaluation of the usefulness of our interface is based on the following main criterias:

- does the user understand what we display ?
- does it help him to recover after an interruption (it takes him less time than without this interface)
- does he need more information than we display ?

- does he need more active ways to get back in his activity ?

Results

The sessions were fully recorded with a screen recorder software for further analysis. At the end of each session, we had a talk with the participant to evaluate his use of our activity timeline for recovering from interruptions.

The results of this preliminary experiment are quite satisfactory. The main remark is that remembering previous work after short interruptions (lower than one minute) does not need any help in many cases, but the possibility we offer to move in the movie to the moment of the last operations is sometime very usefull.

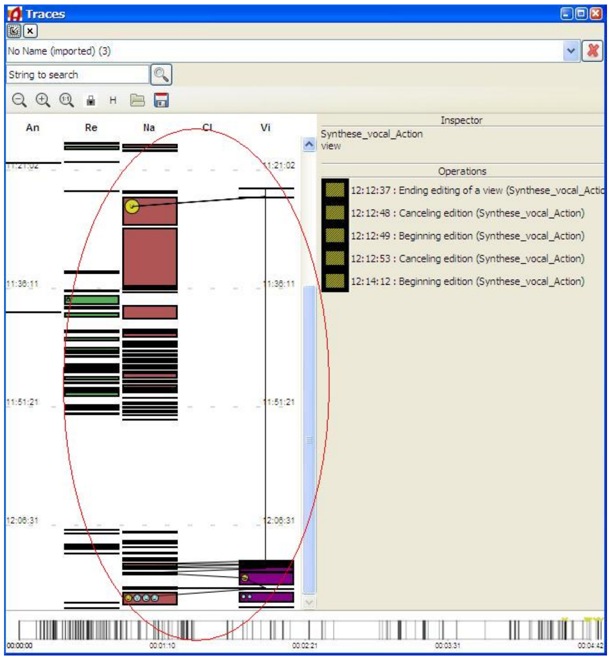
In cases of longer interruptions, the subjects find the interface much more useful. They mainly used it to remember in which part of their task they were. For example, they used it to find what were the latest annotations they were editing before the interruption, or until which instant in the movie they synchronized their subtitle annotations with the speach.

However, they did not really use the global context given by the visualization of the whole trace in the timeline, except in the case of the lunch break. This observation is rather consistant with the studies on attention and cognitive load, as a user does not need to refer to the global history of his activity since it has not disappeared from his mind.

We also clearly identified some patterns (as shown in Figure 8) through the observation of the traces, and plane to conduct other experiments over the long term to determine how we can use them to better help the user recovering from a very long interruption (from one session to another). In this case, the identified pattern is revealed by the different appearences of a *view* item. The user built a view to pronounce subtitle annotations during the playback of the movie (at 0:35). He then tested it (0:35), and edited lots of annotations to synchronize them (0:35 - 1:17). After that, he repeated two actions alternatively: he edited his view and used it to play the movie. This is a pattern of view building and testing, modifying the view according to the result. It can be interesting to identify such pattern, to help the user using his trace, but also to detect possible moments in the activity when the user needs help, for assistant creation for example.

CONCLUSION

In this article we presented our work on an interface designed for interruption recovery in active reading of audiovisual documents. We first presented a definition of video active reading activity and the problem of activity interruption recovery we want to tackle. After a study of related works in activity modeling, attention in HCI and traces, we proposed to build an interface to visualize reflexive traces for helping users getting back to the tasks they were doing before they where interrupted. For this, we presented the data model for active reading of audiovisual documents and the ontological model of the activity we want to trace and show to the user. We then presented our interface for activity recovery and his



Synthese_vocal_Action
view

12:12:37 : Ending editing of a view (Synthese_vocal_Actio
12:12:48 : Canceling edition (Synthese_vocal_Action)
12:12:49 : Beginning edition (Synthese_vocal_Action)
12:12:53 : Canceling edition (Synthese_vocal_Action)
12:14:12 : Beginning edition (Synthese_vocal_Action)

Figure 8. Discovery of a behavioural pattern in the trace.

three components: activity timeline, inspector and radar. We presented the aim of each of these components and the possibilities they offer to get the activity context back in a first time, and get back to work in a second time. We then presented the encouraging results of our preliminary study to test this interface.

There are currently two directions for our work. The first concerns improvements of our activity timeline and of our first interface for visualizing traces as an accumulation of events, with more experiments to test them. The second direction consists in studying more in details the identification of behavioural patterns during active reading activity and how we can use them to support more actively the active reader.

REFERENCES

1. P. D. Adamczyk and B. P. Bailey. If not now, when?: the effects of interruption at different moments within task execution. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 271–278, New York, NY, USA, 2004. ACM.

2. O. Aubert, P.-A. Champin, Y. Pri, and B. Richard. Active reading and hypervideo production. *Multimedia Systems Journal, Special Issue on Canonical Processes of Media Production*, 2008. 6 pp.

3. O. Aubert and Y. Prié. Advene: active reading through hypervideo. In *ACM Hypertext'05*, pages 235–244, Salzburg, Austria, Sep 2005.

4. L. Bannon, A. Cypher, S. Greenspan, and M. L. Monty. Evaluation and analysis of users' activity organization. In *CHI '83: Proceedings of the SIGCHI conference on*

Human Factors in Computing Systems, pages 54–57, New York, NY, USA, 1983. ACM.

5. J. Bardram, J. Bunde-Pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 211–220, New York, NY, USA, 2006. ACM.
6. S. K. Card and A. Henderson, Jr. A multiple, virtual-workspace interface to support user task switching. In *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, pages 53–59, New York, NY, USA, 1987. ACM.
7. S. K. Card, T. P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Commun. ACM*, 23(7):396–410, 1980.
8. D. Cram, B. Fuchs, Y. Prié, and A. Mille. An approach to user-centric context-aware assistance based on interaction traces, June 2008. MRC 2008, Modeling and Reasoning in Context.
9. A. N. Dragunov, T. G. Dietterich, K. Johnsrude, M. McLaughlin, L. Li, and J. L. Herlocker. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 75–82, New York, NY, USA, 2005. ACM.
10. H. R. Hartson, A. C. Siochi, and D. Hix. The uan: a user-oriented representation for direct manipulation interface designs. *ACM Trans. Inf. Syst.*, 8(3):181–203, 1990.
11. S. T. Iqbal and B. P. Bailey. Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1489–1492, New York, NY, USA, 2005. ACM.
12. S. T. Iqbal and E. Horvitz. Disruption and recovery of computing tasks: field study, analysis, and directions. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 677–686, New York, NY, USA, 2007. ACM.
13. B. E. John and D. E. Kieras. The goms family of user interface analysis techniques: comparison and contrast. *ACM Trans. Comput.-Hum. Interact.*, 3(4):320–351, 1996.
14. M. Kipp. Anvil - a generic annotation tool for multimodal dialogue. In *Proceedings of Eurospeech 2001*, pages 1367–1370, Aaborg, Sep 2001.
15. K. Kuutti. Activity theory as a potential framework for human-computer interaction research. pages 17–44, 1995.
16. J. Laflaquire, L. S. Settouti, Y. Pri, and A. Mille. A trace-based System Framework for Experience Management and Engineering. In *Second International Workshop on Experience Management and Engineering (EME 2006) in conjunction with KES2006*, Oct. 2006.
17. A. N. Leontjev. *Activity. Consciousness. Personality*. Englewood Cliffs, Prentice Hall, New York, 1978.
18. D. C. McFarlane. Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. *Human-Computer Interaction*, 17:63–139, 2002.
19. T. Nakamura and T. Igarashi. An application-independent system for visualizing user operation history. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 23–32, New York, NY, USA, 2008. ACM.
20. S. J. Payne and T. R. G. Green. Task-action grammars: A model of the mental representation of task languages. *SIGCHI Bull.*, 19(1):73, 1987.
21. R. Pea, M. Mills, J. Rosen, K. Dauber, E. W, and E. Hoffert. The Diver project: interactive digital video repurposing. *IEEE Multimedia*, 11(1), Mar 2004.
22. S. Pfeiffer, C. Parker, and C. Schremmer. Annodex: a simple architecture to enable hyperlinking, search and retrieval of time-continuous data on the web. In *5th ACM SIGMM International workshop on Multimedia information retrieval*, pages 87–93, 2003.
23. P. Rabardel and G. Bourmaud. From computer to instrument system: a developmental perspective. *Interacting with Computers*, 15(5):665–691, 2003.
24. B. Richard, Y. Prié, and S. Calabretto. Towards a unified model for audiovisual active reading. In *Tenth IEEE International Symposium on Multimedia.*, pages 673–678, Dec. 2008.
25. 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.
26. the M.I.P. Elan: A linguistic multimedia annotator, 2006.
27. T. Tran-Thuong and C. Roisin. Multimedia modeling using mpeg-7 for authoring multimedia integration. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, 2003.