Watch-and-Comment as a Paradigm toward Ubiquitous Interactive Video Editing

RENAN G. CATTELAN
Universidade de São Paulo
CESAR TEIXEIRA
Universidade Federal de São Carlos
and
RUDINEI GOULARTE and MARIA DA GRAÇA C. PIMENTEL
Universidade de São Paulo

The literature reports research efforts allowing the editing of interactive TV multimedia documents by end-users. In this article we propose complementary contributions relative to end-user generated interactive video, video tagging, and collaboration. In earlier work we proposed the *watch-and-comment* (WaC) paradigm as the seamless capture of an individual's comments so that corresponding annotated interactive videos be automatically generated. As a proof of concept, we implemented a prototype application, the WaCTool, that supports the capture of digital ink and voice comments over individual frames and segments of the video, producing a declarative document that specifies both: different media stream structure and synchronization.

In this article, we extend the WaC paradigm in two ways. First, user-video interactions are associated with edit commands and digital ink operations. Second, focusing on collaboration and distribution issues, we employ annotations as simple containers for context information by using them as tags in order to organize, store and distribute information in a P2P-based multimedia capture platform. We highlight the design principles of the watch-and-comment paradigm, and demonstrate related results including the current version of the WaCTool and its architecture. We also illustrate how an interactive video produced by the WaCTool can be rendered in an interactive video environment, the Ginga-NCL player, and include results from a preliminary evaluation.

Categories and Subject Descriptors: H.5.1 [**Multimedia Information Systems**]

General Terms: Human Factors, Experimentation

Additional Key Words and Phrases: Annotation, interactive digital video, P2P collaboration, Ginga-NCL

ACM Reference Format:

Cattelan, R. G., Teixeira, C., Goularte, R., and Pimentel, M. da G. C. 2008. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. ACM Trans. Multimedia Comput. Commun. Appl. 4, 4, Article 28 (October 2008), 24 pages. DOI = 10.1145/1412196.1412201 http://doi.acm.org/10.1145/1412196.1412201

1. INTRODUCTION

Although capturing digital information is as easy as ever, for instance for sharing cell phones captured images with an absent person [Kindberg et al. 2005], combining digital artifacts in a homemade video

The authors thank the following organizations for their support: FINEP, FAPESP, CAPES, and CNPq. R. G. Cattelan is a PhD candidate supported by FAPESP (03/13930-4).

demands the use of specialized authoring tools which a typical digital camera user may be not acquainted with—particularly when interactive video is supposed to be produced.

Researchers have investigated the problem of allowing the editing of interactive TV multimedia documents on the client side [César et al. 2006b] as well as on the server side [de Resende Costa et al. 2006]. On the server side, live editing is demanded in situations in which not all information has been defined at authoring time—that is, before broadcast time on the server side [de Resende Costa et al. 2006]. On the client side, motivations include to allow a user to share that content with one or more members of a peer group [César et al. 2006b] or, in a more general approach, to exploit a secondary screen in order to help users during the control, enrichment, sharing, and transfer of interactive television content [César et al. 2008]. Considering related research work, in this paper we propose complementary contributions relative to end-user-generated interactive video, video tagging and collaboration, and social TV.

In the area of ubiquitous computing, which investigates alternatives for providing services to users in a transparent way [Weiser 1991], the term *capture and access* refers to the task of "preserving a record of some live experience that is then reviewed at some point in the future" [Abowd et al. 2002].

Exploiting the capture and access concept, watch-and-comment sessions can be automatically captured. Therefore, a corresponding annotated interactive video can be generated automatically. As a result, when the video is played back, the annotations can be made available along with the video. Such an approach takes advantage of the fact that capture devices are available in most computer-based platforms. Microphones may be used to capture audio in most computers, PDAs and smart phones. Similarly, hardware or software keyboards are always available.

The watch-and-comment approach can be also exploited in Interactive TV settings where next-generation remote controls are available—as in the settings investigated by Tsekleves et al. [2007] and by Bulterman et al. [2006]. In the first case, authors investigate prototypes for accessing and controlling the TV. In the second case, authors investigate a novel architecture for viewer-side enrichment of content, also considering the availability of a secondary screen [César et al. 2008].

We have previously explored such ideas by defining [Pimentel et al. 2007] and demonstrating [Pimentel et al. 2008] the watch-and-comment (WaC) paradigm. The original prototype application WaCTool supports the capture of digital ink and voice comments over individual video frames and segments, producing a Ginga-NCL document that synchronizes the different media streams. Ginga-NCL[1] is the Brazilian Digital TV Standard for declarative interactive programs. An important feature of this approach, from the digital rights perspective, is that edits and annotations are kept separate from the original media, which means that they can be distributed independently from the video stream.

In this paper, we extend our original proposal for the WaC paradigm in a number of ways. First, user-video interactions are associated with edit commands (loop, seek, skip and slow motion)—the aim is to demonstrate the opportunity for users to seamless author interactive video considering those conventional editing options. Second, focusing on collaboration and distribution issues, we employ annotations as simple containers for context information by using them as tags in order to organize, store and distribute information in our P2P-based multimedia capture platform [Cattelan and Pimentel 2008]. We also detail digital ink operators which can be used to explicitly expand and filter annotated contents. Finally, we include preliminary evaluation results.

Relative to the related work cited above, we argue that our approach brings the following contributions:

—The support for the user to personalize linear media, producing interactive video;[2]

[1]http://www.ncl.org.br

[2]In this text we use *video* to refer to digital (noninteractive) video, and *interactive video* to refer to digital interactive video.

—The support to remote communication among users beyond colocated settings enables broader social aspects of video watching;
—The use of video identifiers and tagging as context containers, which organize information and enable collaboration among active participants via P2P groups;
—The support to complementary digital ink manipulation operation, including ink filters and ink expanders;
—The use of tags to index specific portions of the video timeline, allowing search in the shared annotations;
—The provision of personalized, tag-oriented, context-based content search and retrieval of related media.

In the remaining of this article, we discuss design principles of the watch-and-comment paradigm in Section 2; we present the prototype we build to validate our proposal in Section 3, and detail its architecture in Section 4. In Section 5, we include a description of how our prototype allows the capture of digital ink and voice comments during video playback, as well as the available interactive edit commands and the P2P-based collaboration and content sharing mechanism. The resulting interactive video is presented in Section 6, illustrating how it can be used in the SBTVD platform. Samples of the declarative code generated as a result of the interaction are shown in Section 7; the presentation includes excerpts of the corresponding Ginga-NCL code produced. Results from an evaluation of the prototype by means of inspection techniques are presented in Section 8. In Section 9, we discuss how our research compares to others in the literature. We present our final remarks and pointers to future work in Section 10.

2. DESIGN PRINCIPLES FOR THE WATCH-AND-COMMENT PARADIGM

Considering that watching and commenting a video with someone else is a practice many people enjoy and feel comfortable with, the main premise underlying the WaC paradigm is that, while a user watches a video, any natural user-video interaction (such as a voice comment) can be captured and reported in an interactive video specified by means of a declarative document (e.g., one described in SMIL or NCL). We call the period while this interaction occurs a *watch-and-comment session*. The approach is a general one, according to the following principles.

—*There is no restriction with respect to the source of the video*. the media can be obtained live from a camera, from TV broadcasting, or played back from some storage device in the computer, in the set-top box or a media player;
—*There is no restriction with respect to the type of the video*. For instance, a video stream can be generated from a set of images converted into a video stream;
—*There is no restriction with respect to language of the resulting document*. For instance, Ginga-NCL or SMIL or other declarative language can be used;
—*The session can be collaborative, distributed, and synchronous*. More than one user (remote or collocated) can collaborate in a *watch-and-comment session* of the same source video at the same time;
—*The declarative document generated keeps annotations separate from the original media*. This means that the annotations and edits can be distributed independently from the video stream—which is an important feature as far as digital rights are concerned;
—*There is no restriction with respect to the media used for commenting as long as the media can be captured*. The capture can be transparent from the user's perspective (e.g., voice captured by a microphone, electronic ink from pen-based devices, gestures captured by sensors such as accelerometer

and compass as those present in the Wii[3] remote control or the iTouch[4] player). The capture can be also explicit, such as words typed in a keyboard and actions performed to produce some particular result (e.g., a tap on an image to indicate the start or end of a video segment);

—*There is no restriction with respect to how the capture interaction is to be used*. This means that applications can innovate in terms of what to do with the captured interaction. On the one hand, a particular user interaction could be associated to any command in the resulting interactive video (e.g., *skip*). In this case, when the resulting interactive video is later watched, the corresponding command would be offered as an option for the user (that is, the user would choose whether to skip at that particular moment);

On the other hand, the very fact that the resulting video is interactive is also an option: all interactions captured during a watch-and-comment session could be interpreted as commands associated with mandatory edits—and the result would be that a new non-interactive video is produced;

—*There is no restriction with respect to how the resulting declarative document is distributed*. This means that the interactive video could be stored and played back only on the device it has been captured (say, the user's own next generation remote control), or there may exist an integration of the user's environment with a Web repository (such as YouTube,[5] or AsterPix[6] for instance);

—*A watch-and-comment session could be initiated with a declarative document*. In other words, users can watch-and-comment an interactive video as well a linear one. This would demand that the annotation tool includes parsers for the corresponding format—as it is the case for the Ambulant Annotator [César et al. 2006a] and the NCL Composer [Guimarães et al. 2008].

The principles outlined stress how general the overall approach is—the prototype presented in the next section illustrates a few of many possibilities one may be able to envision by applying the WaC paradigm in the context of interactive TV in general, and end-user authoring in particular.

3. THE WACTOOL PROTOTYPE

As a proof of concept of the application of WaC paradigm, we have designed the WACTOOL. The current version of the WACTOOL prototype which includes, besides the multimodal annotation features demonstrated elsewhere [Pimentel et al. 2008], support to edit commands and a new set of P2P capabilities that allow user collaboration and content and metadata sharing.

The WACTOOL is designed for use with network-capable personal devices such as tablet PCs but can also be used with iDTV remote controls with touch screen and Bluetooth, for instance. The examples shown in this article illustrate the use of the prototype running on a tablet PC with pen-based interaction and audio capture capabilities.

The tool has options to open an existing video file in several formats—we use Java Media Framework[7] and its supported formats. In other words, the tool supports the authoring of interactive video from noninteractive media.

When the user executes the tool and selects a video file, the WACTOOL presents four panels as illustrated in Figure 1 (clockwise from the top left): the *playback window* containing a panel for video playback, the usual buttons for *play/pause/stop*, as well as buttons for recording *text and audio notes*; the *ink window* containing a panel for pen-based annotation (among other resources); the *shared content*

[3]http://wii.nintendo.com/controller.jsp
[4]http://www.apple.com/ipodtouch/
[5]http://www.youtube.com
[6]http://www.asterpix.com/
[7]http://java.sun.com/products/java-media/jmf/

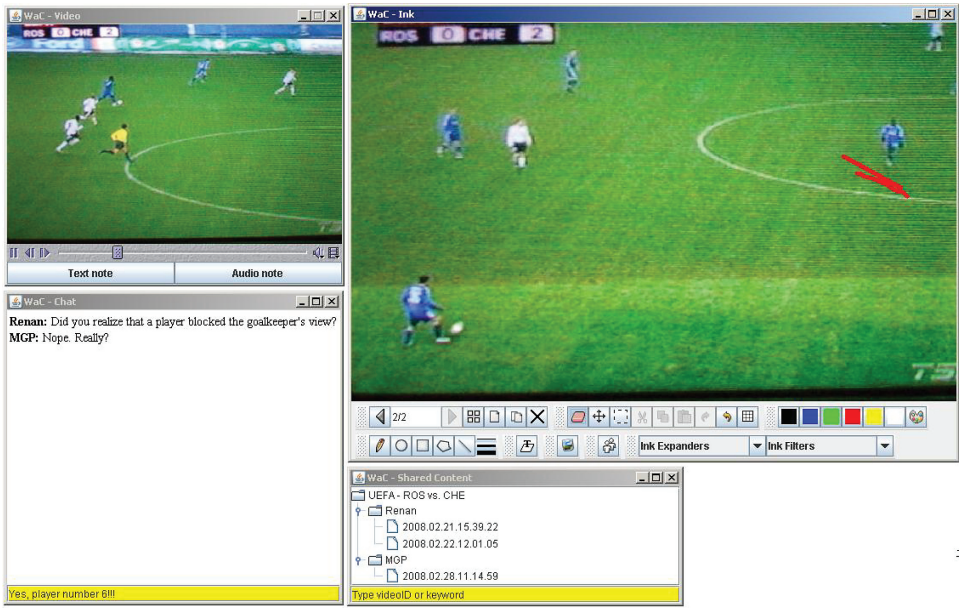

Fig. 1. WaCTool (clockwise from the top left): video *playback window*; *ink window* for pen-based annotation on top of a video frame grabbed from the playback window; *shared content window*; and the *chat window*.

window presenting content (video and annotations) related to the video being watched or the result of user search queries (based on the video ID and the text annotations/tags); and the *chat window* for collaboration via text chat among users currently active in the system.

4. THE WACTOOL ARCHITECTURE

An architecture supporting the WaC paradigm should contain components for annotation and collaboration, as illustrated in the two main blocks in Figure 2: *Annotation* and *Collaboration*.

4.1 Annotation Components

The Annotation components, shown on the left portion of Figure 2, are as follows:

—The *Video source* module is responsible for accessing the video stream and making it available to the *Video player* module. Our current implementation uses Java Media Framework and it's supported video formats. Observing that many home-made videos available (on YouTube, for instance) are built from a set of pictures, we have also specified that the video source module should allow a user to annotate a set of pictures so as to produce the interactive video. In this case, the video source is considered a set of pictures from a folder pointed to by the user. This facility is part of our work in progress and has not been made available in the current version;

—The *Video player* renders the video in the video panel, controlled by the *Interactive commands* module. The starting point of our work has been the production of interactive video from noninteractive media. This means that the *Video player* module does not include a DOM parser to process NCL or SMIL documents;

—The *Interactive commands* module allows the user to interact with the video media. Regarding video control, the usual options for the video playback (with the traditional play, pause and sliders widgets) are supported. Regarding edit commands, the options are skip, loop, and slow motion;

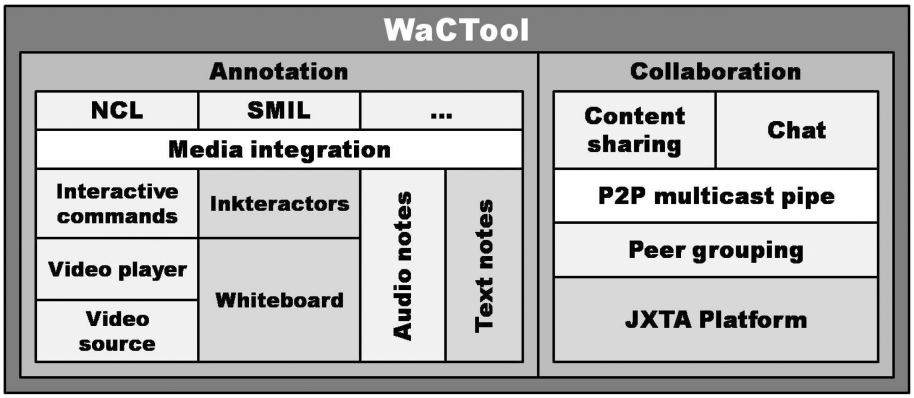

Fig. 2. The WaCTool architecture with two higher level blocks for *Annotation* (left) and *Collaboration* (right).

—The *Whiteboard component* allows the user to add pen-based annotations to the video frame grabbed by the user when interacting with the video: the frame is passed from the *Interactive commands* module to *Whiteboard* module via the *Media integration* and the *Inkteractors* modules. The current implementation allows a rich set of operations involving the ink (select ink color and pen thickness), the pen (free-form or geometric drawings), the frame (duplicate, delete), and object editing operations such as select/move/delete, undo/redo, copy/paste. Regarding annotation, a useful feature is the inclusion of images provided by the user. To support collaboration, it is possible to list the users who are annotating the same video;

—The *Inkteractors* module supports the simple registering of the ink annotations or more elaborate operations (such as generating one frame for each different ink color picked by the user, or for all the ink strokes used by a single user in a collaborative multiuser session);

—The *Audio notes* module is responsible for capturing audio comments from the user and comprise controls for start/stop audio recording;

—The *Text notes* module is responsible for presenting a dialog window in which the user can enter words via a software keyboard, for instance;

—The *Media integration* module captures all the multimodal interactions performed by the user while watching the video: this information is used to generate a NCL (or SMIL) document containing the user's comments and edits—the code is generated by the *NCL* component (or the *SMIL*) component shown in Figure 2.

Details of the supported multimodal interaction are provided by the individual media components (Text notes, Audio notes, and Whiteboard) presented in Section 5.1. The edit commands, as supported by the current version of the components, are detailed in Section 5.2.

4.2 Collaboration Components

The Collaboration components are built on top of the *JXTA P2P platform* and organized as follows:

—The *Chat component* is responsible for creating chat rooms associated with the videos being watched by the users, and allows the exchange of textual messages among users participating in the same chat room (and watching the same video);

—The *Content sharing component* allows users to view and share common resources;

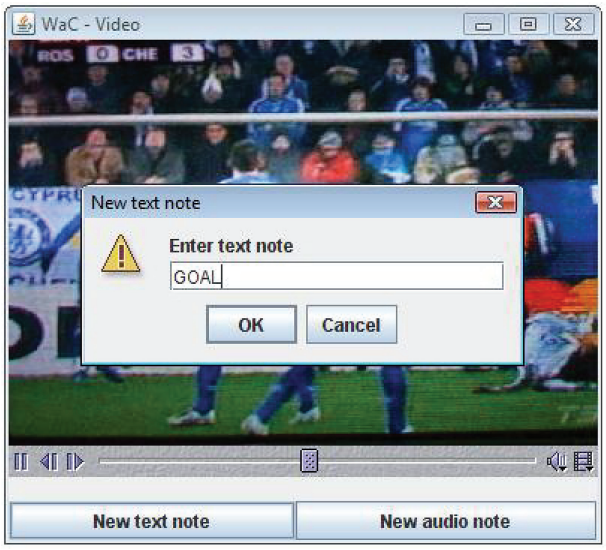

Fig. 3. Dialog window for text annotation input.

—The *P2P multicast pipe* component is responsible for delivering messages at the same time to several users;

—The *Peer grouping* component is responsible for coordinating groups of related peers.

A detailed discussion of the services provided by the Collaboration components is provided in Section 5.3.

5. THE WACTOOL IN USE

This session considers a scenario where one or more users have selected the same video file using the WACTOOL as illustrated in Figure 1. Once the video is loaded, the user starts a watch-and-comment session by pressing the *play button*, which causes the video to start in the *playback window*. At any moment during playback of the video the user may capture text, audio and/or ink notes, or add interactive edit commands related to the video, as detailed in the following sections.

5.1 Multimodal Annotations

The WACTOOL allows the capture of text, voice and digital ink annotations in a watch-and-comment session.

5.1.1 *Text.* The tool allows the creation of text annotations, by clicking the text note button located under the video playback panel in the playback window: this causes a dialog window to pop up, giving the user a text box which accepts textual input (as illustrated in Figure 3). Text input can be done via a physical or virtual (software) keyboard, as it is many times the case for tablet PCs. An extension could be added to support handwriting recognition (as it is the case with M4NOTE [Goularte et al. 2004]).

Single-word entries of text annotation are used for video tagging and allow the creation of navigation menus in the beginning of the video stream and the search for related resources (video files and annotations). Both processes are later detailed in Sections 5.2 and 5.3, respectively.

5.1.2 *Audio.* To record an audio comment, the user clicks the audio note button under the video panel on the video playback window (as shown again at the bottom of Figure 3): the button label changes

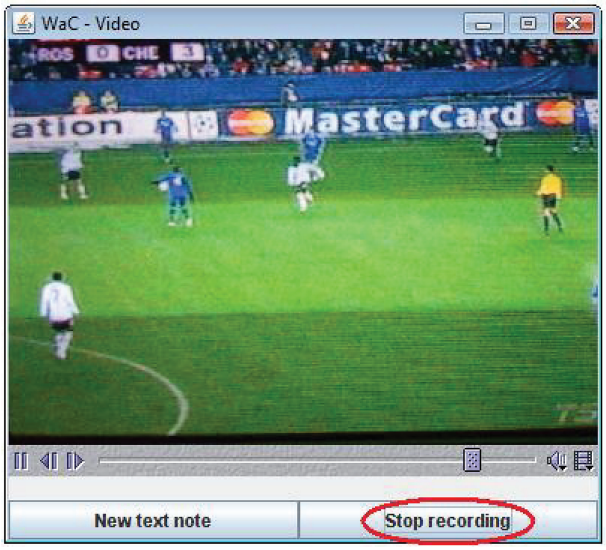

Fig. 4. Recording audio note: clicking on the *audio note button* under the video playback activates the recording of audio comments and changes the button label to *stop recording*; a second click on the button stops the recording and the button label changes back to *audio note*.

Fig. 5. Left: A user tapping on the *playback window* causes a copy of the current frame to be presented on the *ink window*; in this example, the video continues its playback (which explains why the frame in the *ink window* is different from the frame in the *playback window*). Right: Once a frame is available in the *ink window*, the user can make many types of annotations.

to stop recording (as shown at the bottom of Figure 4) to indicate that the audio is being recorded; a second click on the stop recording button stops the recording and the button label changes back to audio note.

5.1.3 *Digital Ink.* To create an ink comment, the user taps on the playback window which causes a copy of the corresponding video frame to be copied into the ink window, as shown in the example in Figure 5 (left).

Whenever a frame has been grabbed and is presented in the *ink window*, the user can make pen-based annotations such as a free-form drawing as illustrated in Figure 5 (right), or a handwritten note such as the name of someone in the frame; it is also possible to include typed text or other images (such as a photo).

Moreover, eleven inkteractors can be used to further enhance the annotation (as the Ink Expanders and Ink Filter options shown on the bottom-right portion of the ink window in Figure 1). The inkteractors available can be divided into four categories: Time-based, Attribute-based, Action-based, and Position-based. Considering that each drawn stroke has a timestamp relative to the start of the annotation session, time enables the definition of two simple but interesting time-based operators: TimeSlice(), which considers the timeline of inking activity to periodically generate derived versions of the annotations, and IdleTime(), which generates derived versions of annotations immediately before idle periods in the interaction.

Given that many attributes of pen strokes are collected during capture, available attribute-based operators are ChangeOnAttributes() and FilterByAttribute(). The fact that the user can perform several actions using the pen strokes (drawing, erasing, changing color, etc), action-based operators include ChangeOnAuthor() and FilterByAuthor(). Finally, four position-based operations are available including ChangeOnArea() and FilterByArea().

These *inkteractors* are available to users as an alternative to automatically generate derived versions of the annotations. Their use can lead to the availability of a large number of rich annotations build upon small user-interactions.

Independent of its type, one important issue is for how long an annotation will impact the playback of the original video. In all cases, we define a include time duration corresponding to how long it took to capture the annotation. In case of voice comments, this allows the comment to be played back as captured. In the case of a text comment, the default duration corresponds to the time it would take for text to be typed again. Regarding digital ink, the duration would allow the ink to be played back again on top of the image. When inkteractors are used for expansion, for instance, the duration time is repeated considering the duration of the original annotation. Moreover, in all cases it is possible to have an option in which a user can remove annotation and resume the video playback using the remote control.

5.2 Interactive Edit Commands

The WACTOOL allows the user to create interactive video commands while interacting via pen-based input with the playback window. The previous version of the tool included only a skip function [Pimentel et al. 2008]. In the current version, three new commands have been introduced: seek, slow motion, and loop.

Seek. An implicit seek function is implemented as follows: every time a single-word text note is entered by the user (as in Figure 3), an entry is created in the index at the beginning of the video, allowing the user to jump straight to the point of interest in the video timeline when that text note (tag) was made. This works as a navigation menu or table of contents.

Skip. The skip operation, as available in the current implementation, allows a user to indicate a portion of video that can be skipped when the interactive video is later watched. Using our tool, when the user identifies the start of such a portion, she taps on the left bottom corner of the playback window; when the end of the video portion to be skipped is reached, the user taps on the right bottom—both tap operations are indicated in Figure 6 (bottom left and bottom right).

It is relevant to observe that, in the current version, we have decided to shift by 3 seconds the skip time indicated by the user: the idea is to acknowledge that, when the user taps on the playback window to indicate the skipping, the intended start and end times have been presented, in fact, a few seconds earlier.

Loop. Figure 6 (center left and center right) illustrates the regions Start loop and End loop in which a user may tap to define the start and ending points of a video segment to be played in a loop in the interactive video. Again, we shift the start and ending times by 3 seconds to reflect the natural user delay when performing the operation.

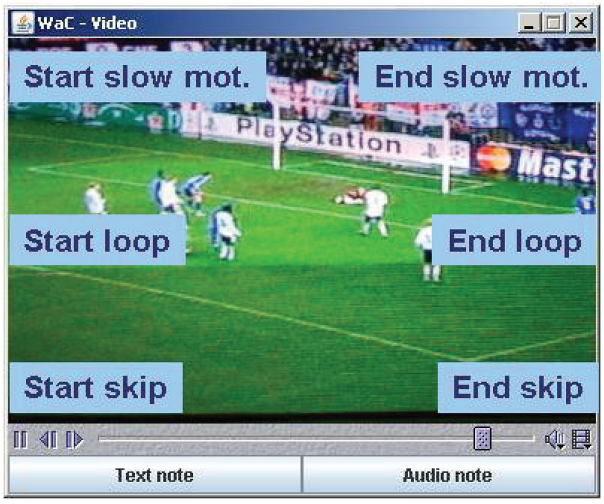

Fig. 6. Interactive edit commands can be activated by tapping the borders of the video playback panel: a tap on the left border, indicates the start of the command (slow motion, loop, or skip); a tap on the right, its end. When the user moves the mouse over the corresponding region, a hint indicating the respective command is presented.

Slow motion. The slow motion operation enables a user to indicate a portion of video that should be played back in slow motion when the interactive video is reviewed. Figure 6 (upper left and upper right) illustrates the corresponding regions.

5.3 User Collaboration and Content Sharing

The new scenario brought by the dissemination of personal computing devices make room for the incorporation of software applications that exploit the installed computational capabilities to support human activities, leveraging new forms of interaction among the involved participants and allowing the recording of events, the generation of associated documents and the retrieval of captured artifacts.

Multimedia capture in mobile computing environments requires software communication infrastructures capable of handling intermittent connectivity and providing collaboration capabilities in a decentralized arrangement. In this sense, P2P computing comes as a natural choice: sharing and grouping capabilities enable collaboration among peers; dynamic network connections and intermittent digital presence of peers are supported from design; replication of resources provides better overall performance and content availability; single points of failure and bottlenecks are eliminated due to the nonexistence of central entities; and the burden of storage, processing, and network bandwidth is distributed among peers, potentially allowing the development of scalable solutions at a low cost.

P2P networks have become popular in the last years as a simple and efficient way of sharing resources among a large number of interconnected computers, incorporating heterogeneous devices and promoting interoperability among them.

Considering such scenario, we used JXTA[8] to extend our original WACTOOL with P2P-based services to support collaboration and content sharing during watch-and-comment sessions, thus promoting WaC into an emerging "social" paradigm.

[8]http://www.jxta.org

Fig. 7. Chat window for exchanging text messages.

In essence, our system provides two basic services: text chat and content (video and annotation files) sharing, both JXTA-enabled.

5.3.1 *Communication Service.* Users can communicate via textual messages using P2P chat (Figure 7). Chat rooms group users and are created according to the video being annotated in a watch-and-comment session, that is, users commenting the same video stream are put in the same chat room and can exchange messages among them freely.

Chat messages flow through a multicast pipe associated to a peer group created specifically for the ongoing watch-and-comment session. A pipe is an asynchronous and unidirectional message transfer mechanism used for service communication. Multicast pipes are able to deliver messages simultaneously to multiple peers. A multicast pipes is bound to a pipe advertisement and to a peer group. Peer groups are collections of somehow related peers that have agreed upon a common set of services. Peers self-organize into peer groups, each identified by a unique peer group ID. In our case, we use the SHA-1 hash of the video file in order to create the watch-and-comment session's peer group ID.

The resource sharing service allows the user to have a collaborative view of content (video and annotation files) provided by remote users. Shared resources are presented using a tree hierarchy (Figure 8). Root nodes are video files, leaf nodes are annotations, an intermediate level identify annotation authors. Annotated sessions are identified according to the date they were created. Video streams and annotation files can be downloaded by right-clicking the corresponding entry and choosing *download* on a pop-up menu.

5.3.2 *Content Sharing Service.* Shared content and corresponding advertisements are listed on a common multicast pipe, bound to a global peer group. Advertisements are XML metadata structures describing specific details about the service provided (type, ID, etc.). Each peer publishes its content sharing service using a unique ID derived from the peer and user names. To request contents posted on the common multicast pipe, a peer queries its discovery service for the corresponding remote advertisement. If the peer who posted the contents is alive, the response to the query will be successful and

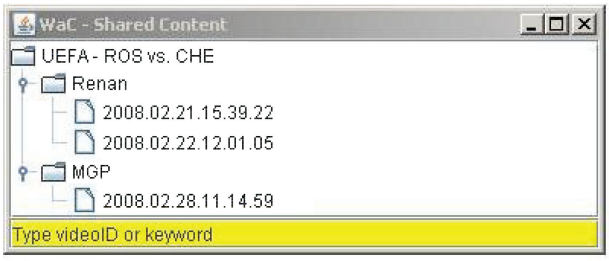

Fig. 8. Shared resources presented as a tree view: root nodes are video files, leaf nodes are annotations, and an intermediate level identify annotation authors. Annotated sessions are identified according to the date they were created.

the peers engage in the process of data exchange. Content requests and data transfers occur directly between the involved peers.

A search mechanism allows queries for video ID (based on the media's filename, initially loaded into the shared content window) and/or tags (targeting single-word text notes). By keeping track of the input tags entered as text annotations, we allow personalized content search and retrieval, extending our focus on the user current context for searching related content. Considering our relatively limited scope of home media networks, we employ the textual annotations fed into the system to filter video content returned from a search query.

Content is stored at the peers local file system and replicated passively as users exchange files (i.e., once a user download a given file, she may start to redistribute it). Replication of content is simplified since the captured information is considered immutable (after finished, if a captured watch-and-comment session needs to be modified or extended, a new session is generated and the original session is preserved). No data consistency mechanism is required.

6. THE RESULTING INTERACTIVE VIDEO

The annotated video may be watched as an interactive video in the tool itself. However, for exchange or publishing purposes, the tool generates an interactive digital TV specification in the form of a Ginga-NCL document—chosen as the standard for the Brazilian Digital TV Terrestrial System. Figure 9 illustrates a watch-and-comment session being presented with Ginga-NCL Player used in combination with a typical remote control. Such process is achieved as follows:

—At each occurrence of an ink comment, a miniature icon of the corresponding annotated frame is presented on the bottom right corner of the video window (as illustrated in Figure 10 (left)) and associated with the blue (square) button on the remote control: if the user presses that button, the video is paused and the annotated frame is presented (as illustrated in Figure 10(right)) until the yellow (triangle) button is pressed to indicate that the video playback should be resumed;

—At each occurrence of an audio note, an audio icon is presented on the upper right corner of the video (as illustrated in Figure 11(left)) and associated with the green (diamond) button on the remote control: if the user presses that button, the volume of the original audio is lowered and the audio commentary is played back—the original audio is resumed at the end;

—At each occurrence of a text note, a text icon is presented on the upper right corner of the video (as illustrated in Figure 11(right)) and associated with the green (diamond) button on the remote control: if the user presses that button, the text commentary is presented as a subtitle;

—At each occurrence of a video to be skipped, a skip icon is presented on the bottom left corner of the video (as illustrated in Figure 12(left)) and associated with the red (circle) button on the

Fig. 9. Watch-and-comment session presented with Ginga-NCL Player.

Fig. 10. A miniature image of the annotated frame is presented (left) to indicate that an ink note is available and is presented (right) if the user selects the blue (square) button on the Digital TV remote control; the yellow (triangle) button resumes the video playback.

remote control: if the user presses that button, the portion of video indicated by the user is skipped;

—At each occurrence of a loop command, a loop icon is presented on the bottom left corner of the video (as illustrated in Figure 12(middle)) and associated with the red (circle) button on the remote control: if the user presses that button, the portion of video is replayed once;

—At each occurrence of a slow motion command, a slow motion icon is presented on the bottom left corner of the video (as illustrated in Figure 12(right)) and associated with the red (circle) button on the remote control: if the user presses that button, the portion of video indicated by the user is presented in slow motion.

Although much processing can be done to extract data from the video (and audio) streams to provide useful information for a viewer of the interactive digital TV, the watch-and-comment paradigm, as

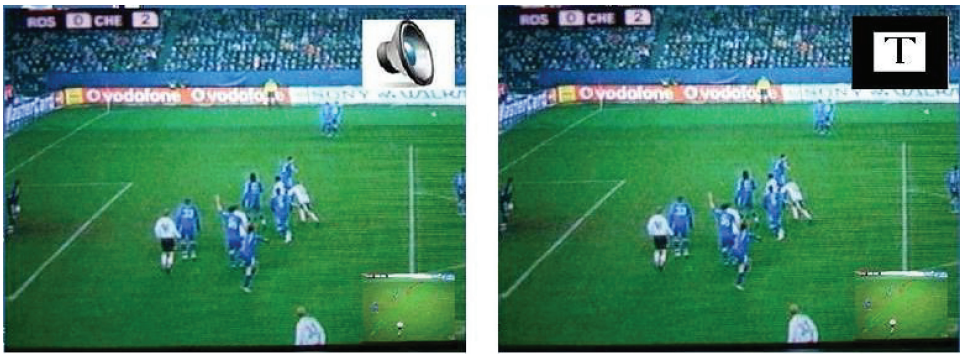

Fig. 11. Left: an audio icon indicates that an audio comment is available and is played back when the user selects the green (diamond) button on the remote control. Right: a text icon indicates that a text comment is available and is presented as subtitle when the user selects the green (diamond) button on the remote control.

Fig. 12. Left: a skip icon indicates that a portion of the video can be skipped if the user selects the red (circle) button on the remote control. Middle: a loop icon indicates that a portion of the video can be replayed if the user selects the red (circle) button on the remote control. Right: a slow motion icon indicates that a portion of the video can be played back in slow motion if the user selects the red (circle) button on the remote control.

illustrated, can be also exploited to enrich the user interaction. INKTERACTORS, for instance, have already been exploited in the skip example above. For the sake of illustration, other operators that are under construction in the current prototype include:

—FilterByAuthor() and FilterByAuthorRole(): used to show icons indicating annotations of a particular user or class (role) of users, this applies when more than one person has contributed with annotations to the same original video;

—FilterByArea(): used to indicate annotations associated with specific portions of the screen.

7. GENERATED NCL DOCUMENTS

When a user decides to stop and terminate an ongoing watch-and-comment session, an annotation file is saved with markup for the annotations made (e.g., ink coordinates and time information), as well as references to the text and audio files containing all the recorded comments. Annotations are stored in an XML format (Ginga-NCL or SMIL) and packed with the corresponding medias (digital ink markup, JPEG images of the whiteboard frames, text note entries and audio note streams captured from the tablet PC built-in microphone) in a single, compressed file.

The interactive video automatically generated must specify references to original contents as well as the contents generated by the user's comments. The examples in this section illustrate portions of an NCL document corresponding to the session detailed in the previous sections.

```
01 <media descriptor="dVideo"
02       src="file:/C:/Users/rgclan/Videos/soccer.avi"
03       type="video/x-msvideo" id="video">
04 <area begin="0.0s" end="121.0s" id="aVisit1"/>
05 <area begin="121.0s" end="145.0s" id="aVisit8"/>
06 <area begin="108.0s" end="116.0s" id="aText5"/>
07 <area begin="108.0s" id="aText5D"/>
08 <area begin="77.0s" end="83.0s" id="aSkip2"/>
09 <area begin="83.0s" id="aSkip2D"/>
10 <area begin="96.0s" end="99.0s" dur="6.0s" id="aSlow4"/>
11 <area begin="88.0s" end="91.0s" id="aLoop3"/>
12 <area begin="91.0s" id="aLoop3D"/>
13 </media>
```

Fig. 13. NCL anchors for ink, text, seek, skip, slow motion and loop.

```
01 <media descriptor="dVisit1Thumb" src="medias/slide0.jpg"
02        type="image/jpeg" id="visit1Thumb"/>
03 <media descriptor="dVisit1" src="medias/slide0.jpg"
04        type="image/jpeg" id="visit1"/>
05 <media descriptor="dVisit8Thumb" src="medias/slide7.jpg"
06        type="image/jpeg" id="visit8Thumb"/>
07 <media descriptor="dVisit8" src="medias/slide7.jpg"
08        type="image/jpeg" id="visit8"/>
09 <media type="application/x-ginga-settings" id="nodeSettings">
10        <property name="menuOption"/> </media>
11 <media descriptor="dText5Thumb" src="../text.jpg"
12         type="image/jpeg" id="text5Thumb"/>
13 <media descriptor="dText5" src="medias/textnotes.html#t5"
14        type="text/html" id="text5"/>
15 <media id="menuOption5" src="medias/textnotes.html#t5"
16        type="text/html" descriptor="dMenu5"/>
17 <media descriptor="skip2" src="../skip.jpg"
18        type="image/jpeg" id="skip2"/>
19 <media descriptor="slow4" src="../slow.jpg"
20        type="image/jpeg" id="slow4"/>
21 <media descriptor="loop3" src="../loop.jpg"
22        type="image/jpeg" id="loop3"/>
```

Fig. 14. NCL descriptors for ink, text, seek, skip, slow motion and loop.

The code in Figure 13 includes a reference to the original video (lines 01–03) and specifies the points of interaction for each additional media created (the start/end times and references to elements that specify their interactive behavior). As an example, the frame grabbed in the interaction described in Figure 5 is referenced in line 05: it was captured 121 seconds after the start of the video and the interaction lasted for 24 seconds. The figure includes (lines 06–07) the specification for the annotation made via a text note (shown in Figure 3), as well the skip (lines 08–09), slow motion (line 10) and loop (lines 11–12) editions achieved by tapping on the video as illustrated in Figure 6.

Depending on the type of interaction, additional media is created and must be referenced in NCL code. Figure 14 presents the NCL code, which describes additional media created in the examples.

For each video frame grabbed by the user and annotated using the whiteboard, two images are created (a thumbnail version to indicate that the annotation is available, and a large one to be presented on top of the video)—their use has been described in Figure 10.

As an example of annotation operations allowed, an excerpt of the code for NCL anchors allowing reviewing a miniature slide annotated with ink is shown in Figure 15. As examples of the code associated with the edit operations, the code allowing a user to perform a seek interaction is shown in Figure 16. Figure 17 presents an excerpt of the code for controlling a skip operation.

```
01 <link xconnector="onBegin1Start1" id="l1Video8">
02 <bind role="onBegin" interface="aVisit8" component="video"/>
03 <bind role="start" component="visit8Thumb"/>
04 </link>
05 <link xconnector="onEnd1Stop1" id="l2Video8">
06 <bind role="onEnd" interface="aVisit8" component="video"/>
07 <bind role="stop" component="visit8Thumb"/>
08 </link>
09 <link xconnector="onKeySelection1PauseNStartN" id="l3Visit8">
10 <bind role="onSelection" component="visit8Thumb">
11 <bindParam value="9" name="keyCode"/>
12 </bind>
13 <bind role="pause" component="video"/>
14 <bind role="start" component="visit8"/>
15 </link>
16 <link xconnector="onKeySelection1ResumeNStopN" id="l4Visit8">
17 <bind role="onSelection" component="visit8Thumb">
18 <bindParam value="5" name="keyCode"/>
19 </bind>
20 <bind role="resume" component="video"/>
21 <bind role="stop" component="visit8"/>
22 </link>
```

Fig. 15. NCL anchors for the interaction corresponding to a miniature slide annotated with ink.

```
01 <switch id="switchOpcao">
02 <bindRule rule="r5" constituent="menuSelection1"/>
03 </switch>
04 <link xconnector="onKeySelection1PauseNStartN" id="showMenu">
05 <bind role="onSelection" component="video">
06 <bindParam value="BLUE" name="keyCode"/>
07 </bind>
08 <bind role="pause" component="video"/>
09 <bind role="start" component="menuOption5" />
10 </link>
11 <link xconnector="onSelection1SetNStopNStartN" id="menu5Selected">
12 <bind component="menuOption5" role="onSelection"/>
13 <bind component="nodeSettings" interface="menuOption" role="set">
14 <bindParam name="var" value="1"/>
15 </bind>
16 <bind component="switchOpcao" role="start"/>
17 <bind component="video" role="stop"/>
18 <bind component="menuOption5" role="stop"/>
19 <bind component="video" interface="aText5D" role="start"/>
20 </link>
```

Fig. 16. NCL code: menu for seek.

8. EVALUATING THE DESIGN

We exploited inspection methods which can be used in early development stages to provide input to our iterative design process.

We first carried out a think aloud evaluation [Wright and Monk 1991] with an arts student acquainted with the production of interactive videos for the SBTVD using the NCL composer. During the final interview, his main remark was that the idea is easy to grasp. He also observed that the slow motion option should be very useful for sports program, that the loop operation should be useful in a more varied set of programs including education and news, and that the *skip* option is probably the less important from the author's point of view. He valued strongly high the possibility of inserting audio and text comments. He also observed that the missing option is the opportunity to make links to other videos as follows: pause the current video, playback the related video, and resume the original video playback at the end.

```
01 <link xconnector="onBegin1Start1" id="l1Video2">
02 <bind role="onBegin" interface="aSkip2" component="video"/>
03 <bind role="start" component="skip2"/>
04 </link>
05 <link xconnector="onEnd1Stop1" id="l2Video2">
06 <bind role="onEnd" interface="aSkip2" component="video"/>
07 <bind role="stop" component="skip2"/>
08 </link>
09 <link xconnector="onKeySelectionStartNStopN" id="l3Skip2">
10 <bind role="onSelection" component="skip2">
11 <bindParam value="4" name="keyCode"/>
12 </bind>
13 <bind role="stop" component="skip2"/>
14 <bind role="stop" component="video"/>
15 <bind role="start" interface="aSkip2D" component="video"/>
16 </link>
```

Fig. 17. NCL code: skip interaction.

Although the current tool allows the author to annotate the video being watched with images (it is one of the options of the *Whiteboard* component), the annotation via a link to other video (or part of it) has not been considered a requirement in the current design. Our starting premise was that making voice comments with respect to a video's content is a common practice to many people when watching video with someone else. If one relationship with some other video is observed by the user while watching a video, he would make a comment about this by means of an audio or a text note. However, linking the current video to another one is a natural demand, in particular for advanced users. Considering this feature as a new requirement would be important in scenarios focusing the producer's perspective.

We also asked a usability specialist to perform an heuristic evaluation on the tool according to the 10 classic heuristics proposed by Nielsen [1992]. The summary of the identified usability problems, according to their impact on the interaction, are as follows. Classified as cosmetic usability problems: the use of an abbreviation in one of the menus and the lack of default configuration with respect to the video to be watched. Classified as minor usability problems: lack of feedback via sound, lack of standard position where the windows appear, the same icon being used to represent media and other objects, terms used may be unfamiliar to users, lack of indication scale of values used (e.g., seconds). Classified as major usability problems: buttons for forward video and slides are not consistent with each other, the cursor does not change to indicate which tool is active, some icons do not change to indicate when they are selected, video window buttons do not include alternative text, lack of shortcut options, and lack of online help. Finally, classified as catastrophic usability problems: text tool on slides did not work, application did not close gracefully, the texts in the menu option were partially shown, and no feedback is provided in case of error in the interaction. We plan to tackle all these problems in the current version aiming at obtaining a prototype that is robust enough to perform formal user evaluations.

Considering the importance of having user feedback in the early stages of the design, we also collected the opinion of one video producer. She specializes on informal scenarios such as camera-and-microphone installations placed in public areas where pedestrians can have their performances captured and made available on the web. Her main positive remark was that the tool could be easily used in the scenarios she worked with. Her main criticism was with respect to the user interface: although easy to understand, the interface should be improved to a nicer look-and-feel—for instance with respect to the icons used on the resulting video to indicate the edit options (skip, loop, and seek). We totally agree with her remarks and plan to tackle this problem in the near future.

We summarize evaluation results in Table I.

Table I. Prototype Evaluation Summary

Evaluation technique	Main issues, findings and requests
Think aloud	— *Slow motion* option useful for sports program — *Loop operation* useful for education and news — *Skip* option less important from the author's point-of-view — Audio and text comments positively evaluated — Request for mechanism allowing the creation of links to other videos
Heuristic evaluation	—Cosmetic: abbreviation in one of the menus and the lack of default configuration —Minor: lack of feedback via sound, lack of standard position where the windows appear, the same icon being used to represent media and other objects, terms used may be unfamiliar to users, lack of indication on scale of values used —Major: buttons for advancing the video and advancing slides are not consistent, cursor does not change to indicate which tool is active, some icons do not change to indicate when they are selected, video window buttons do not include alternative text, lack of shortcut options and lack of online help —Catastrophic: text tool on slides did not work, application did not close gracefully, the texts in the menu option were partially shown, and no feedback is provided in case of error in the interaction
Expert opinion	—Tool could be used with easy in the scenarios expert worked with —Although easy to understand, the interface should be much improved to a nicer look-and-feel

9. RELATED WORK

Considering our own previous experience with a capture and access general platform [Pimentel et al. 2007], we have previously exploited the ubiquitous computing paradigm to associate user annotations with video streams in the design of the Multimedia Multimodal Annotation tool (M4Note), which supports annotations in two complementary methods: context-based metadata association and content enrichment [Goularte et al. 2004]. M4Note comprises a multimodal interface which allows both live video capture and annotations. Annotations can be made by means of pen-based digital ink or via voice recognition and are, in both cases, converted to text. At the end of the capture process, XML documents are generated as a composition of references to all captured media: video, audio, images, slides, ink strokes, and text. Our experience has given us the opportunity to observe situations where intermediary stages and derived representations of a document annotated with digital ink might be relevant to users. We then formalized Inkteractors: operators that can be applied to ink strokes so as to allow the generation of documents containing alternative views of the original interaction process [Cattelan et al. 2008].

Considering related research work, we discuss six trends: User-Generated Content, Interactive Video, Media at the Home, Video Tagging, and Collaboration and Social TV.

9.1 User-Generated Content

Typical video authoring tools may be considered not worth learning by home users because the tools have not been designed for the average home user: they assume trained users with complex authoring tasks at hand. By specialized authoring tools we mean, for noninteractive video, simple tools such as

Movie Maker[9] and iMovie,[10] as well more sophisticated ones such as Premiere[11] Pro CS3. Although the simple ones may be easy to use when compared with professional software, they still require some training by the typical home user (as illustrated by the video tutorial by Belmont [2006]). In fact, there are university level courses in which students learn the basics of communicating with video (a case in point is the long term experience reported by Ross and Ross [2005]).

While we exploit an ubiquitous capture and access approach, several authors report using explicit authoring techniques [Hua and Li 2006; Girgensohn et al. 2000; Hua et al. 2004]. Targeting at the home user, Hua and Li's LazyMedia aims at facilitating editing and sharing home videos using techniques such as content analysis in association with authoring composition and presentation templates [Hua and Li 2006].

In an earlier work, Girgensohn et al. [2000] process video contents, for instance in terms of the type and amount of camera motion, to select frames which are presented to the user, who can then use a drag-and-drop approach to position the desired video in a storyboard. Video and music content analysis are also used by Hua et al. [2004] to automatically collect and align video clips, photographs, music and lyrics to compose a Karaoke video.

Kirk et al. [2007] studied the use of video production by a varied user population, observing that their main aim is usually to share the video in the capture device itself (in the case of portable phones, for instance) and that they usually do not use the editing options embedded in the devices themselves. They also observed teenage users who opted by explicit capture (with digital cameras) and some level of edition (with computer software).

Our work is most related to efforts targeted at nonexpert users, as in the architecture proposed by César et al. providing an approach for end-user enrichment of video streams via many alternative end-user devices [César et al. 2007], in the use of a secondary screen to allow users to control, enrich, share, and transfer interactive television content [César et al. 2008]. Compared to their work, our architecture explores complementary opportunities including: support to remote communication among users beyond colocated settings; the use of video identifiers and tagging as context containers that organize information and enable collaboration among active participants via P2P groups; the support to complementary digital ink manipulation operation including ink filters and ink expanders; the use tags to index specific portions of the video timeline and to allow searching in the shared annotations; and the provision of personalized, tag-oriented, context-based content search and retrieval of related media.

9.2 Interactive Video

In the context of interactive video, authoring tools include VideoClix™[12], which allows users to segment and track objects in a video stream, as well as add annotations and metadata. VideoClix is an authoring tool based on annotation of video objects. The Ginga-NCL Composer [de Resende Costa et al. 2006] allows the authoring of declarative documents via alternative views (structural, temporal, layout, textual). In our work we exploit automated capture to provide the user with a transparent authoring service while he watches and comments on the video.

While our proposal aims at achieving authoring via transparent interaction, other authors opt by the explicit manipulation of tangible objects representing video artifacts. In the multiuser Tangible Video Editor reported by Zigelbaum et al. [2007], users edit video clips by manipulating active handheld

[9]http://www.microsoft.com/windowsxp/using/moviemaker
[10]http://www.apple.com/imovie/
[11]http://www.adobe.com/products/premiere/
[12]http://www.videoclix.com/technology/software/overview.html

computers embedded in plastic cases; in the mediaBlocks system children capture, edit and display media by manipulating passive wooden blocks [Ullmer et al. 1998].

9.3 Media at the Home

Many initiatives explore the field of audiovisual media technologies for home platforms. For example, VisNet [Sadka 2004] covers several research disciplines relating to audiovisual systems and home platforms with a particular focus on creating/coding audiovisual content, storage and transporting audiovisual information over heterogeneous networks, audiovisual communication techniques and security mechanisms. VisNet provides an end-to-end audiovisual framework that supports numerous applications and services based on trusted open interoperable multimedia user platforms and devices, notably for broadcasting and in-home platforms with full interactive capacity, enabling new media-rich and interactive audiovisual applications and services for home platforms and mobile users.

On the commercial arena, with the Windows Home Server[13] platform, Microsoft is one among many other software vendors recognizing the demand for a unified solution to organize and share families' digital memories and captured content.

9.4 Video Tagging

Ramos and Balakrishnan [2003] present a system that allows users to make annotations while watching a video: the textual annotation is made while the video is watched. The system lacks an explicit model in order to enable annotations to be used in other operations, like search.

The work developed by Shamma et al. [2007] illustrates an approach where some applications were designed in order to use contextual meta-data about how media content is being used in specific contexts. They argue that this is a shift from semantics, where applications try to understand the meaning of the media content, to pragmatics, where the knowledge of the context usage can be useful.

Commercial efforts experimenting with video tagging have been around for a while. Used initially by services such as YouTube[14] and Google Video[15] to improve search, video tagging is now also seen as a useful tool for indexing the (intra) video content. The key idea is to let users tag individual sections of a video so that others can skip straight to particular points of interest.

VeoTag[16] and Click.TV[17] were two pioneer endeavors which explored video tagging as a primary feature. With Veotag, one can tag videos in a clip library. Click.TV, on the other hand, provides a richer interaction experience for the casual user, when compared to Veotag. Rather than tags, the site is focused on the discussion of videos (sports coverage, presidential speeches, keynotes and documentaries) via user comments, allowing users to add a running commentary. They also provide a feature to post on blog and social networking profile (e.g., MySpace).

9.5 Collaboration and Social TV

User communication features have been previously integrated into live television broadcast systems. For example, Media Center Buddies [Regan and Todd 2004] and Telebuddies [Luyten et al. 2006] provide instant messaging and chat to allow viewer to communicate among themselves. AmigoTV allows voice chat and a shared viewing experience, bringing together buddies viewing the same live broadcast channel and allowing them to share their emotional reaction to the onscreen content through overlaid avatars [Coppens et al. 2004].

[13]http://www.microsoft.com/windows/products/winfamily/windowshomeserver/default.mspx
[14]http://www.youtube.com
[15]http://video.google.com
[16]http://www.veotag.com
[17]http://click.tv

A few authors have studied the impact of chat on user perception while watching video. Weisz et al. [2007] explore the potential to transform video watching from a passive, isolating experience into an active, socially engaging one. They empirically examined the activity of (text) chatting while watching video online and found that chat has a positive influence on social relationships and that people chat despite being distracted. They have also examined the experience of watching together and how that promoted conversation among strangers and affected evaluation of the video, pointing out that socializing around media is perhaps just as important as the media itself.

Geerts [2006] looked at two modes of communication for interactive TV: besides text chat, he has also considered voice chat and the advantages and disadvantages of each one. The study found that voice chat is considered more natural and direct, making it easier to keep on following the program. Both studies, however, stress the need for a balance between the fun of sharing and discussing the program with others versus the potentially negative impact of distraction on perceiving and processing the video content. Specific design solutions should minimize the distraction from watching the television program.

Shaw and Schmitz [2006] present the concept of community annotation and remix of multimedia archives. The main contribution is a platform where the design of paradigmatic human-centered computing applications gives attention to user experience and social dynamics. The platform provides users with a system for fun, creative exploration of media collections, allowing deploy with real communities of users outside of a lab. Also, they argue that this data can be used to develop emergent semantics for the media being explored and reused, having potential to improved media retrieval, browsing, and authoring applications.

Harboe et al. [2008] have conduced two studies on social television concepts. In the first one a social TV prototype was tested in the field, allowing groups of users watching television at home to talk to each other over an audio link. The results of the tests with patterns of use pointed out users did perceive the system as valuable. In the second study, groups were presented with several social TV concepts, and their responses were collected. They conclude that participants deal with potential conflicts between conversation and television audio without the need for additional technical support, and there is no indication that a video link would improve the experience.

9.6 Automated Capture and Ubiquitous Video

Digital inking systems are part of many ubiquitous computing platforms. The user's natural interaction via pen, gestures, audio, video or sensors, for instance, can be captured so as to transparently produce associated multimedia documents that can later be reviewed in an integrated and synchronous manner.

A few projects focus on helping users to develop an understanding of complex information by exploring the relationships underlying digital ink annotations, its representation and related medias. Bulterman [2003] elucidates the requirements for an environment supporting user-centered analysis of annotations. He worked with SMIL[18] tools, the Ambulant Player, and its companion Ambulant Annotator, aimed at allowing interactive multimedia SMIL documents. In work targeting interactive digital TV, Bulterman and his colleagues address the viewer-side enrichment of multimedia content [Bulterman et al. 2006; César et al. 2006a]. We move toward this goal by giving the user increased control for customizing the review of annotations, for instance by means of the *inkteractors* available using the *whiteboard* module.

Liu and Chen [2005] address explicit and implicit correlations among various media streams in a composite document. As in our work, their system considers temporal and spatial relationships to allow the replay of a video (corresponding to a captured lecture) as a synchronized multimedia document. The

[18]http://www.w3.org/AudioVideo

main difference is that their techniques, such as the adaptable handwriting that groups handwriting and text for example, are applied during the capture phase while we focus on both capture and reviewing phases.

Processing the digital video to support authoring has been extensively investigated. Truong and Venkatesh [2007] provide a survey on techniques used to process video information to generate video summaries. The use of metadata to support authoring video has been also reported [Madhwacharyula et al. 2006], in particular in the context of the TV-Anytime [Butkus and Petersen 2007]. User and context information has also been discussed [den Ende et al. 2007], including efforts to improve TV Based Communication [Hemmeryckx-Deleersnijder and Thorne 2007].

10. CONCLUSION

We have highlighted the design principles of the WaC paradigm, and demonstrated related features supported in the current version of our WACTOOL. The WaC paradigm enables nonexpert users to produce interactive video while enjoying the video. This is done by associating user-video interactions with video editing commands (like skip, loop and seek) and tagging users' comments with the video. Those commands and comments are translated with the video into a declarative document which can be rendered into an interactive video.

Users' comments (audio, text or digital ink) are explored as containers to contextual information. Tagging the video with contextual information makes it possible to search and retrieve personalized related media. This makes the WaC paradigm meet the Ubiquitous Computing vision—we are exploring information present in the user-video interactions in order to provide personalized services to users without taking their attention to their main task. In this way, by integrating concepts of ubiquitous computing and video annotation, the WaC paradigm provides transparent end-user authoring of interactive multimedia content. Moreover, by discussing opportunities to exploit digital ink for authoring multimedia content, the approach results in novel ways a user can seamlessly author interactive video streams.

Another advantage of this work is the remote communication. We have demonstrated how collaboration, organization and distribution issues can be tackled in the WaC paradigm, enabling the social experience of watching a video with others. Both collocated and remote P2P-based collaboration are supported as well as the sharing of annotations that extend some previous content.

As future work, our plans include having a version of the WACTOOL prototype robust enough to perform further evaluations. Also with respect to the WACTOOL, we are working on a new version with options for creating an interactive video file using a set of images or photos as a starting point. We are also studying approaches to use live video. Yet another current implementation effort is a version that allows a watch-and-comment session to start with an interactive video instead of a linear one—so one can edit and extend existing annotations. These efforts, related to important research opportunities discussed in the state of the art literature, are to be carried out taking into account complementary end-user roles and as well their feedback.

ACKNOWLEDGMENTS

We thank Dick Bulterman for great discussions on this topic. We thank Luis F. G. Soares for inspiring us with Ginga-NCL. We thank the users and specialists, in particular Erick Melo and Vivian Motti, who helped us with the preliminary evaluation, and the anonymous reviewers for the many important suggestions. We thank Felipe S. Santos for his collaboration in our previous work, and Bruno C. Furtado for providing SMIL capabilities to our tool.

REFERENCES

ABOWD, G. D., MYNATT, E. D., AND RODDEN, T. 2002. The human experience. *IEEE Pervasive Comput. 1*, 1, 48–57.

BELMONT, V. 2006. Create your own video blog. *CNet Reviews, http://reviews.cnet.com/Create_your_own_video_blog4660-10165_7-6634979.html*.

BULTERMAN, D. C. A. 2003. Using smil to encode interactive, peer-level multimedia annotations. In *Proceedings of the ACM Symposium on Document Engineering (DocEng'03)*. ACM, New York, 32–41.

BULTERMAN, D. C. A., CÉSAR, P., AND JANSEN, A. J. 2006. An architecture for viewer-side enrichment of tv content. In *Proceedings of the 14th Annual ACM International Conference on Multimedia (MULTIMEDIA'06)*. ACM, New York, 651–654.

BUTKUS, A. AND PETERSEN, M. 2007. Semantic modelling using tv-anytime genre metadata. In *Proceedings of the European Interactive Television Conference*. 226–234.

CATTELAN, R. AND PIMENTEL, M. 2008. Supporting multimedia capture in mobile computing environments through a peer-to-peer platform. In *Proceedings of the ACM Symposium on Applied Computing (SAC'08)*. ACM, New York, 1246–1251.

CATTELAN, R. G., TEIXEIRA, C., RIBAS, H., MUNSON, E., AND PIMENTEL, M. 2008. Inkteractors: interacting with digital ink. In *Proceedings of the ACM Symposium on Applied Computing (SAC'08)*. ACM, New York, 1246–1251.

CÉSAR, P., BULTERMAN, D. C. A., AND JANSEN, A. J. 2006a. The ambulant annotator: empowering viewer-side enrichment of multimedia content. In *Proceedings of the ACM Symposium on Document Engineering (DocEng'06)*. ACM, New York, 186–187.

CÉSAR, P., BULTERMAN, D. C. A., AND JANSEN, A. J. 2006b. Benefits of structured multimedia documents in idtv: the end-user enrichment system. In *Proceedings of the 2006 ACM Symposium on Document Engineering (DocEng'06)*. ACM, New York, 176–178.

CÉSAR, P., BULTERMAN, D. C. A., AND JANSEN, A. J. 2008. Usages of the secondary screen in an interactive television environment: Control, enrich, share, and transfer television content. In *Proceedings of the European Interactive Television Conference*. 168–177.

CÉSAR, P., BULTERMAN, D. C. A., OBRENOVIC, Z., DUCRET, J., AND CRUZ-LARA, S. 2007. An architecture for non-intrusive user interfaces for interactive digital television. In *Proceedings of the European Interactive Television Conference*. 11–20.

COPPENS, T., TRAPPENIERS, L., AND GODON, M. 2004. Amigotv: towards a social tv experience. In *Proceedings of the European Interactive Television Conference*.

DE RESENDE COSTA, R. M., MORENO, M. F., RODRIGUES, R. F., AND SOARES, L. F. G. 2006. Live editing of hypermedia documents. In *Proceedings of the ACM Symposium on Document Engineering (DocEng'06)*. ACM, New York, 165–172.

DEN ENDE, N. V., DE HESSELLE, H., AND MEESTERS, L. 2007. Towards content-aware coding: User study. In *Proceedings of the European Interactive Television Conference*. 185–194.

GEERTS, D. 2006. Comparing voice chat and text chat in a communication tool for interactive television. In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction (NordiCHI'06)*. ACM, New York, 461–464.

GIRGENSOHN, A., BORECZKY, J., CHIU, P., DOHERTY, J., FOOTE, J., GOLOVCHINSKY, G., UCHIHASHI, S., AND WILCOX, L. 2000. A semi-automatic approach to home video editing. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST'00)*. ACM, New York, 81–89.

GOULARTE, R., CATTELAN, R. G., CAMACHO-GUERRERO, J. A., VALTER R. INÁCIO, J., AND DA GRAÇA C. PIMENTEL, M. 2004. Interactive multimedia annotations: enriching and extending content. In *Proceedings of the ACM Symposium on Document Engineering (DocEng'04)*. ACM, New York, 84–86.

GUIMARÃES, R. L., COSTA, R. M. R., AND SOARES, L. F. G. 2008. Composer: Authoring tool for itv programs. In *Proceedings of the European Interactive Television Conference*. 61–71.

HARBOE, G., MASSEY, N., METCALF, C., WHEATLEY, D., AND ROMANO, G. 2008. The uses of social television. *Comput. Entertain. 6*, 1, 1–15.

HEMMERYCKX-DELEERSNIJDER, B. AND THORNE, J. M. 2007. Awareness and conversational context sharing to enrich tv based communication. In *Proceedings of the European Interactive Television Conference*. 1–10.

HUA, X.-S. AND LI, S. 2006. Interactive video authoring and sharing based on two-layer templates. In *Proceedings of the 1st ACM International Workshop on Human-Centered Multimedia (HCM'06)*. ACM, New York, 65–74.

HUA, X.-S., LU, L., AND ZHANG, H.-J. 2004. P-karaoke: personalized karaoke system. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MULTIMEDIA'04)*. ACM, New York, 172–173.

KINDBERG, T., SPASOJEVIC, M., FLECK, R., AND SELLEN, A. 2005. I saw this and thought of you: some social uses of camera phones. In *Proceedings of the Extended Abstracts Conference on Human Factors in Computing Systems (CHI'05)*. ACM, New York, 1545–1548.

KIRK, D., SELLEN, A., HARPER, R., AND WOOD, K. 2007. Understanding videowork. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*. ACM, New York, 61–70.

Liu, K.-Y. and Chen, H.-Y. 2005. Exploring media correlation and synchronization for navigated hypermedia documents. In *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA'05)*. ACM, New York, 61–70.

Luyten, K., Thys, K., Huypens, S., and Coninx, K. 2006. Telebuddies: social stitching with interactive television. In *Proceedings of the Extended Abstracts Conference on Human Factors in Computing Systems (CHI'06)*. ACM, New York, 1049–1054.

Madhwacharyula, C. L., Davis, M., Mulhem, P., and Kankanhalli, M. S. 2006. Metadata handling: A video perspective. *ACM Trans. Multimed. Comput. Comm. Appl. 2*, 4, 358–388.

Nielsen, J. 1992. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'92)*. ACM, New York, 373–380.

Pimentel, M., Baldochi, L., and Cattelan, R. 2007. Prototyping applications to document human experiences. *IEEE Pervasive Comput. 6*, 2, 93–100.

Pimentel, M., Goularte, R., Cattelan, R., Santos, F., and Teixeira, C. 2007. Enhancing multimodal annotations with pen-based information. In *Proceedings of the Workshop on New Techniques for Consuming, Managing, and Manipulating Interactive Digital Media at Home*. IEEE Computer Society, Los Alamitos, CA, 207–213.

Pimentel, M., Goularte, R., Cattelan, R., Santos, F., and Teixeira, C. 2008. Ubiquitous interactive video editing via multimodal annotations. In *Proceedings of the European Interactive Television Conference*. 72–81.

Ramos, G. and Balakrishnan, R. 2003. Fluid interaction techniques for the control and annotation of digital video. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST'03)*. ACM, New York, NY, 105–114.

Regan, T. and Todd, I. 2004. Media center buddies: instant messaging around a media center. In *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction (NordiCHI'04)*. ACM, New York, NY, 141–144.

Ross, J. M. and Ross, K. R. 2005. Developing web-based video training modules to aid students learning multimedia skills. *J. Comput. Small Coll. 21*, 2, 281–287.

Sadka, A. H. 2004. Visnet: Noe on networked audiovisual media technologies. In *Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services*.

Shamma, D. A., Shaw, R., Shafton, P. L., and Liu, Y. 2007. Watch what i watch: using community activity to understand content. In *Proceedings of the International Workshop on Multimedia Information Retrieval (MIR'07)*. ACM, New York, 275–284.

Shaw, R. and Schmitz, P. 2006. Community annotation and remix: a research platform and pilot deployment. In *Proceedings of the 1st ACM International Workshop on Human-Centered Multimedia (HCM'06)*. ACM, New York, 89–98.

Truong, B. T. and Venkatesh, S. 2007. Video abstraction: A systematic review and classification. *ACM Trans. Multimed. Comput. Comm. Appl. 3*, 1, 3.

Tsekleves, E., Cruickshank, L., Hill, A., Kondo, K., and Whitham, R. 2007. Interacting with digital media at home via a second screen. In *Proceedings of the Workshop on New Techniques for Consuming, Managing, and Manipulating Interactive Digital Media at Home*. IEEE Computer Society, Los Alamitos, CA, 201–206.

Ullmer, B., Ishii, H., and Glas, D. 1998. mediablocks: physical containers, transports, and controls for online media. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'98)*. ACM, New York, 379–386.

Weiser, M. 1991. The computer for the 21st century. *Scientific American 265*, 3, 94–104.

Weisz, J. D., Kiesler, S., Zhang, H., Ren, Y., Kraut, R. E., and Konstan, J. A. 2007. Watching together: integrating text chat with video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*. ACM, New York, 877–886.

Wright, P. C. and Monk, A. F. 1991. The use of think-aloud evaluation methods in design. *SIGCHI Bull. 23*, 1, 55–57.

Zigelbaum, J., Horn, M. S., Shaer, O., and Jacob, R. J. K. 2007. The tangible video editor: collaborative video editing with active tokens. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction (TEI'07)*. ACM, New York, 43–46.

Received August 2008; accepted August 2008