

Volume 15, Number 4, October 2002

ISSN 0958-8221
CODEN CALLEE

CALL

Computer Assisted Language Learning

WE ARE
ONLINE

SWETS & ZEITLINGER
PUBLISHERS



Video-Based Language Learning Environment Using an Online Video-Editing System

Yoshiaki Hada, Hiroaki Ogata, and Yoneo Yano

Department of Information Science and Intelligent Systems, Faculty of Engineering,
Tokushima University 2-1, Tokushima, Japan

ABSTRACT

Unlike most videoconferencing systems, the new videoconferencing system we have developed allows sessions between a native English teacher and a non-native second language learner to be recorded and edited for later reuse for educational instruction purposes as a teaching aid thus enhancing explanations and comments about pronunciation and intonation. To develop easy-to-use tools for adding such enhancements, a new paradigm called video correction is proposed implementing as a video correction system so that the learner can understand the teacher's comments as if they were corrections on paper. The prototype system, called *Viclle* (Video-based Communicative Language Learning System), supports both a teacher (a native English speaker) and a learner (non-native speaker) with video correction and communication. A new XML schema, called *VCML* (Video-based Correction Markup Language), is exploited to represent pedagogical correction essentially in a compact text format so that both teacher and learner can communicate through e-mail without sending video files back and forth. In addition, *VCML* relates the teacher's corrections to the video with links, and it includes the learner's feedback. Therefore, the teacher and the learner can communicate with each other by using the *VCML* document only.

1. INTRODUCTION

Traditional foreign language learning in Japan is based on systematic instruction and focuses on writing. It pays little attention to other communicative skills like pronunciation, intonation and gestures. In addition, it is difficult for a teacher to coach well in a large class. In order to acquire skills to communicate in a foreign language, the learner needs to hear the target language

Address correspondence to: Yoshiaki Hada, Department of Information Science and Intelligent Systems, Faculty of Engineering, Tokushima University 2-1, Minami-Josanjima, Tokushima 770-8506, Japan. E-mail: {hada, ogata, yano}@is.tokushima-u.ac.jp, URL: <http://www-yano.is.tokushima-u.ac.jp/>

with his/her own ears. This approach is known as the communicative approach (Johnson & Morrow, 1981). This is a foreign language learning method that primarily emphasizes communication. Grammatical rules and vocabularies have a secondary importance. Recently, the approach has attracted much interest in the Computer Assisted Language Learning (CALL) field.

High-speed broadband networks have realized richer, smoother synchronous communication and collaboration using not only text but also audio and video equipment. Moreover, cameras are being loaded into cellular phones and PDAs, enabling flexible and mobile communications. Thus today, a user can not only communicate with a distant person but also learn with a native English speaker who lives in an overseas country, all in real time. In a school of conversational English in Japan, for instance, Japanese learners can communicate with a distant teacher via videoconference. The interactive video makes learning more effective (Dalton, 1990). The learner can solve questions in the class by asking the teacher directly. The teacher can correct pronunciation and intonation. In addition, the teacher can intervene in real time even with non-verbal information like gestures. The learner can attempt to master the language using actions and non-verbal information.

What seems to be lacking, however, is reuse of the conversational video for language learning by reviewing and editing the videos of past lessons. A teacher may miss a student's error in the class with a videoconferencing system. And if the learner wants to review a lesson, the learner has to take copious notes about important points. When the class is finished, a learner may have difficulty memorizing questions that occurred to him/her during the learning time. However, if the videoconferencing session is recorded, both teacher and learner can grasp the situation more easily by watching the learning session.

It takes a lot of time to play videos, however, therefore, supporting functions are required to make recorded videoconferencing sessions more useful. If a teacher could add an explanation or edit errors in the video, for instance, it would be easy for a learner to understand his mistake. And if a learner could ask about the details of errors, he could grasp the error points more easily. Traditional video techniques are unsuited to this task, since neither the teacher nor the learner are usually trained in these techniques. In addition, data is typically lost as a result of video editing, which makes doing "before and after" comparisons difficult. Thus, new supporting functions must be devised to make recorded learning sessions more useful.

When a teacher alters the document of a learner, the method called "correction" is used in the document. This shows the wrong part with a red

pen on the same paper. It is easy for the learner to grasp the teacher's intent. In this method, the learner can compare the original document with the corrected one. Therefore, the learner can understand errors easily. According to such a traditional correction manner, Ogata, Feng et al. (2000) and Ogata, Hada, and Yano (2002) have developed online markup-based collaborative writing support systems, called Communicative Correction Assisting System (CoCoA) and CoCoA for Java (CoCoA-J), which are very similar to a real correction environment with pen and paper. The systems can automatically generate an original text and a revised one from a corrected text. In addition, the corrected data can be sent to a learner through e-mail.

This same concept of the correction can be applied to video. We call it video correction. Using video correction, it is easy to get the original video, the corrected one, and the corrected parts with correction marks. Video playback can be customized easily by using working information and parameters. Thus, the video becomes a much more valuable teaching aid.

This paper proposes a video correction system by supplementing what the teacher wants to revise and comment on. The prototype system, called Video-based Communicative Language Learning System (Viclle), extends a normal video-conferencing system to a language learning system which can be used to support video correction on pedagogical videos.

This paper is organized as follows. First, an online video-editing model is described. Then, video correction is proposed. Next, Section 3 illustrates the example of the system's usage. Section 4 outlines the kinds of teacher-learner interactions we aim to support by VCML. Our implementation of Viclle is illustrated in Section 5. Related research is introduced in Section 6. Finally, concluding remarks are given in Section 7.

2. ONLINE VIDEO EDITING MODEL AND VIDEO CORRECTION

In this section we classify an online video editing model and describe its features. Then we give details of our video correction paradigm.

2.1. Online Video Editing Model

In order to correct a document, there are several editing systems that enable teachers to review and correct learners' drafts. Farkas and Poltrock (1995)

describe an online text-editing model, including the silent-editing model, comment model, edit trace model, and traditional markup model. In the silent editing model, the teacher overwrites and corrects the learner's document. In the comment model, annotations are inserted in the original document. In the edit trace model, the teacher works in the manner of an author, for example, deleting, adding, and moving text as needed. The computer can compare the editor's new version with the original one, and allow learners to view the draft that contains the teacher's corrections. Microsoft Word supports this model. A traditional markup model employs correction marks to the computer screen, for example, deletion, insertion, and move. Along the lines of this classification, we propose an online video-editing model as follows: (See Fig. 1)

- (A) *The silent model*: After correcting the wrong parts of the original video with a general video-editing tool, the teacher creates a new video file. This is the simplest model but it is very difficult for the learner to check the teacher's work. This model is destructive because the teacher cannot readily recover the original video once he/she has made the revised video.
- (B) *The comment model*: This model employs pop-up notes, temporary footnotes, hidden texts, and special symbols placed within the original video. In most annotation systems, this model is applied. However, although it is

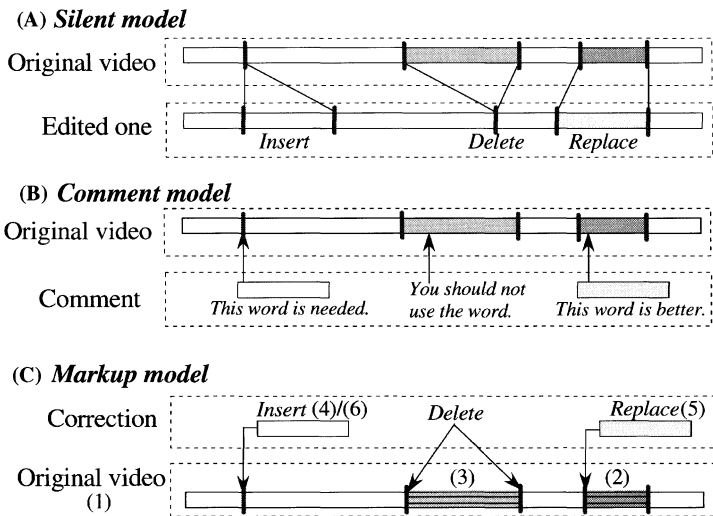


Fig. 1. Taxonomy of online video editing models.

easy to check an original video, a learner cannot get the revised video applied with the comments. For example, a system based on this model is developed with SMIL. It often takes considerable time for teachers and learners to obtain the revised video from comments in the original video.

- (C) *The mark-up model*: This adapts the traditional marks with pen and paper to video editing on the computer screen. The symbols are both familiar and intuitive for teachers and learners; for example, deletion, insertion, and move. For instance, MATE (Hardock, Kurtecbach, & Buxton, 1993) which is a text-based collaborative writing tool, allows teacher to use both digital ink and voice commands toward pen and voice computing. In this model, both learners and teachers can interpret the teacher's modification with markings much more readily than in the other models. Moreover, they can easily derive both the original video and the revised one from a marked-up video. The contents which have been corrected by the teacher are independent of the original video. In addition, the learner can understand the teacher's work easily. The contents for playback are made in real time.

Many systems allow multiple users to mark-up an electronic document as if they were marking up a printed copy of the document. However, no studies have ever tried to adapt a markup model into a video-editing system. Therefore, this paper proposes a video markup model for supporting language learning.

2.2. Video Correction

In a video editing system, a "scene" is the basic atomic unit of editing. The scene has one thing to tell the audience, and it is composed of frames. An editor creates a video by combining scenes. Hitchcock (Girgensohn et al., 2000), for example, allows users to easily create custom videos from raw video shot with a standard video camera. Hitchcock uses automatic analysis to determine the suitability of portions of the raw video. In order to create a custom video, the user drags key-frames corresponding to the desired clips into a storyboard.

In contrast, for language learning, a teacher revises the pronunciation and the action included in a video. Thus, the video needs to be treated with a smaller unit than the "scene." We therefore focus on the frame. Figure 2 shows a comparison of text and video correction. The correction for text in (A) shows deletion and insertion of characters. The video correction in (B) shows them for frames. This shows that video frames can be treated like text characters.

The teacher creates a lot of information in correcting a video. A video correction system displays corrections to the learner by using this information.

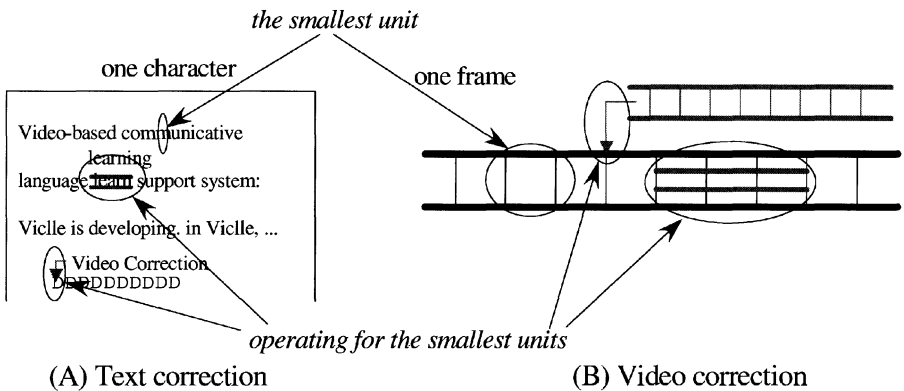


Fig. 2. Text-based and video-based correction.

In text correction, a learner can get the information, for example, delete, replace and insert, from marks and text written by a teacher directly on the paper document. However, in video correction, the corrected information needs to be shown to a learner by computer. Therefore, attaching semantics, for example, revision and delete, to the information corrected by a teacher is important. We model video correction as follows (see Fig. 1):

- (1) *Original video*: An original conversational video of both a teacher and a learner
- (2) *Error*: The error part that a teacher has selected in the original video
- (3) *Delete*: The deleted part which a teacher has selected in the original video
- (4) *Explanation*: The newly made information that a teacher has created for explanation
- (5) *Revision*: The newly made information that a teacher has created to replace the wrong part
- (6) *Comment*: The newly made information that a teacher has created to comment on, for example, giving advice
- (7) *Index*: The corrected contents of the original video

Semantics are attached to a correction using this information, and the system can classify the inserted contents by these categories. So, a system can show a learner the necessary information for studying when the corrected video is played.

2.3. Learning Flow by Video Correction

In the case of text correction, a learner writes a document to be revised by a teacher, then the teacher corrects it. In the case of video correction, a video is

created as the result of reading a videoconferencing session between teacher and learner, then the learner and teacher study the video with a video correction system asynchronously. The learning flow of Figure 3 will be described below.

- (1) *Recording with videoconferencing system:* The teacher gives a spoken language lesson. The videoconferencing system in Viclle records the video of both the teacher and the learner. The learner and the teacher share the video files. The video will be used as the primary artifact for later review.
- (2) *Correcting video:* The teacher corrects the recorded video with the Viclle-Editor. The instructor makes the data for correction and corrects the video by using the data. The teacher sends a VCML file (made by Viclle-Editor) to the learner after correcting. The VCML file contains only the corrections made by the teacher, but the video is shared via network, so the learner can use the VCML file to view the corrections in the video.
- (3) *Learning by correction:* The learner studies using the corrected video with the Viclle-Viewer. According to the VCML file and the learner's process, the Viclle-Viewer plays the contents. The student reviews and repeats trouble spots or places that were difficult (indicated by the teacher's corrections) to better understand the lesson.
- (4) *Entering questions:* If the learner has a problem or a question, s/he can describe it in the video. In addition, if there is a question from the teacher, s/he can answer it. In both cases, the Viclle-Viewer generates a VCML file

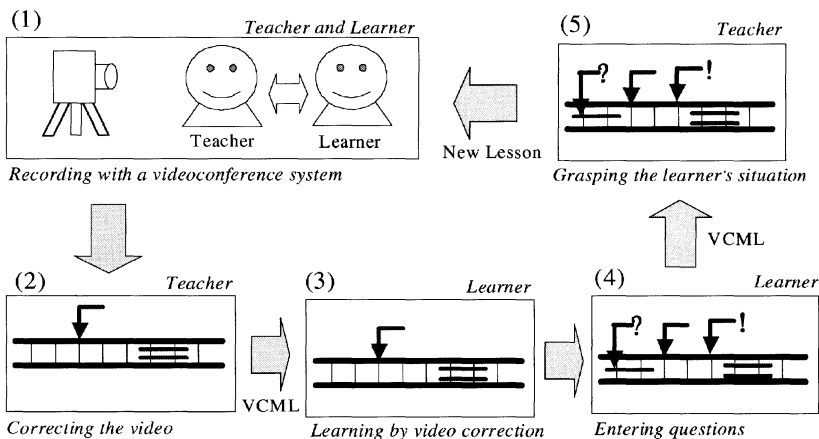


Fig. 3. Learning flow by video correction.

and sends it to the teacher. This VCML file contains the learner's question and feedback.

- (5) *Grasping learner's feedback*: The teacher grasps the learner's feedback from the VCML file with the Viclle-Editor and can use it to tailor subsequent lessons to the learner's specific needs. The Viclle-Editor shows the student's feedback for each correction, and new problems raised by the student are shown separately.

The learner and the teacher repeat the learning process outlined above and try to solve the learner's problems. In this way, the learner improves his/her communication skills.

3. EXAMPLE OF VICLLE'S USAGE

This section describes an example of interactions using the Viclle system. In this case, the learner is Japanese and he is playing the role of a tourist who has left his bag at a hotel. The teacher is a native English speaker who is playing the part of an information clerk in the hotel. The learner talks with the teacher using a videoconferencing system. The conversation scene between them is recorded into the their computers. A part of the conversation is as follows:

- (1) [L: Learner] I'm looking for my bag. (The learner's pronunciation is bad.)
- (2) [T: Teacher] How long ago?
- (3) [L] Just before I left my bag.
- (4) [T] Just a second ago? I understand.
- (5) [L] Do you know that?
- (6) [T] What bag is it?
- (7) [L] A shoulder bag (with a gesture).
- (8) [T] It's not a shoulder bag but the bag that the strap attached.
- (9) [L] You mean.

After the role play, the teacher corrects the video with the Viclle-Editor as follows: (Note: The brackets means [location of the above conversation, type of the correction mark, type of the play mode, level of the importance].)

- (a) [(1), Replace, compare, 2] I'm looking for my bag. (The teacher pronounces correctly.)
- (b) [after (3), Insert, normal, 3] We don't use "just before." Please say "just a second ago."

- (c) [after (3), Insert, normal, 1] I will show you another expression. “I noticed that my bag was missing.” or “I just realized that my bag was gone.”
- (d) [(5), Replace, normal, 2] Has anyone turned it in?
- (e) [(9), Overlap, question, 3] What do you mean? Did you mean, “you are mean”?

After the teacher sends his corrections to the learner, the learner views the corrected video with *Viclle*. If the learner selects the level 3 (the error must be corrected, in accordance with the level of importance assigned), the *Viclle-Viewer* plays (b) and (e). The learner can reply to the teacher’s question (e), and can insert some of his own questions into the video. In playing a correction, the original video pauses where there is a corrected error. Then, the process of the correction is played. Once completed, the original video is resumed. After the learner studies the corrected video, the *Viclle-Viewer* packages feedback which is sent to the teacher as a VCML file. The teacher receives the learner’s feedback and uses it to tune instruction in subsequent lessons. In this way, *Viclle* facilitates asynchronous interactive learning between a learner and a teacher.

4. FEEDBACK BETWEEN TEACHER AND LEARNER

In this section, we describe a feedback method between teacher and learner. First, the taxonomy of a teacher’s feedback to a learner is described. Then, correction marks needed are defined. After that, the taxonomy of a learner’s feedback to a teacher is described. The feedback marks needed are defined. We then define Video-based Correction Markup Language (VCML), which is used to encode these taxonomies.

4.1. Taxonomy of Teacher’s Feedback

In a lesson, a teacher asks questions to check a learner’s understanding. The teacher may guide lessons based on the feedback. For instance, if a learner could not understand a section, a teacher could rephrase what s/he said or s/he could return to a previous section. If a learner understands it, a teacher proceeds to the next section. Thus, in order to proceed with a lesson, feedback is important. A teacher’s feedback is classified into the following categories (Koike, 1994):

- (1) *Explicit feedback*: A teacher breaks into the conversation and corrects a learner’s wrong expression. An example is shown as follows:

(L: Learner) I goed to New York Yesterday.

(T: Teacher) No, you are not correct. Please say “I went to New York yesterday.”

- (2) *Confirmation check*: A teacher confirms a learner’s incorrectly spoken phrase by an interrogative sentence. The teacher does not explicitly give a correct expression. For example,

(L) I goed to New York Yesterday.
(T) You went yesterday?
- (3) *Clarification request*: A teacher asks a learner to restate the wrong expression, for example:

(T) Could you say it again?
- (4) *Implicit feedback*: A teacher corrects a learner’s statement. However, the teacher does not break the conversation. An example is:

(L) I goed to New York Yesterday.
(T) I went there yesterday, too.
- (5) *Indication of non-comprehension*: A teacher expresses that s/he cannot understand a learner’s statement. For example, a teacher says “what?”

Correction marks are defined for these activities in the next section. The marks are based on conventions for text correction, however text is static whereas video is dynamic. Thus, we need not only basic correction marks but also marks for asking questions allowing adequate time to compose the questions.

4.2. Correction Marks

Teachers do not teach with speech and gestures alone. They use paper or a blackboard as well. In addition, they repeat important parts, and they engage their students. In practicing pronunciation and intonation, for instance, a teacher may have a learner who imitates the teacher’s accent over and over. Moreover, a teacher will not only explain concepts unilaterally, but will also ask questions.

Thus the learning environment is highly interactive, and Viclle must support this richness. In particular, a learner using the Viclle-Viewer should see just what the teacher creates in the Viclle-Editor. In order to get the same output between a teacher and a learner, a set of correction marks is defined beforehand. In the system, a teacher can correct by using these marks, and a learner can see the same correction worked by the teacher.

According to the above classifications and discussion, correction marks for video should include the following mark types (Refer to the categories of Section 3.1):

- (a) *Insert*: This mark is needed for all feedback. This is needed to insert an explanation on the video or a missing piece of conversation, it is used. This can show the teacher's feedback directly.
- (b) *Overlap*: This mark is needed for all feedback. This can show the teacher's feedback indirectly. This composes the correction into frames of the original video.
- (c) *Delete*: This is needed for implicit feedback. This mark can show a learner unnecessary conversation or sentences. This deletes parts from the original video.
- (d) *Replace*: This is needed for confirmation checking and implicit feedback. This mark can be used to rephrase a learner's utterance in more suitable words. In short, this can replace an error with a correction. In addition, it can compare the error with the revision.
- (e) *Annotation*: This mark can be used to show advice and comments. The comments appear in another window, and are distinct from corrections.
- (f) *Repeat*: This mark is needed for clarification requests and Indications of non-comprehension. If a teacher wants a learner to practice intonation and pronunciation, this mark can be used.
- (g) *Pause*: This is needed for clarification requests and indication of non-comprehension. Essentially, "Pause" is used to ask whether a learner understands.

Each of the correction marks above (except Annotation) implies some action in replaying the corrected video. These actions include:

- (1) *Normal*: The video is played normally. The video is played this way when the correction is an explanation or a revision.
- (2) *Comparison*: The revised video and the original video are played side-by-side. They may be played alternately, or simultaneously. The video is played this way when the correction is a replace mark.
- (3) *Question*: The video pauses in order to give the learner an opportunity to answer the teacher's question if, for instance, the teacher asks a question after making a correction.

In addition, the teacher can assign a level of importance to each correction s/he makes. This helps the learner to decide what to do with each of the teacher's corrections. They include:

- (*Level 1*) It does not need to be corrected. It shows another expression and so on.

(Level 2) It should be corrected (this is the default).

(Level 3) It has to be corrected.

4.3. Taxonomy of Learner-Teacher Feedback

If a teacher is available, it is natural that a learner will advance his/her learning by interacting with the teacher. However, in an asynchronous environment, the learner cannot ask questions directly, since the teacher is not present. We propose that the teacher can receive the learner's feedback via the same video correction system used for language instruction. In this case, however, communication marks are needed in place of correction marks. Here, we classify the types of feedback a learner may employ. According to the taxonomy and discussion, the feedback marks for a learner are defined.

- (1) *Clarification request*: If a learner cannot understand or hear, a learner requests a clearer statement.
(L: Learner) Could you please say that again? / Sorry, I didn't get that.
- (2) *Confirmation check*: To confirm whether the learner understands his/her partner's statement
(L) Did you say...? / Was that...?
- (3) *Comprehension check*: To confirm whether the partner understands the learner's statement
(L) OK? / Are you with me? / Are you following me?

In an actual face-to-face lesson, teacher gets learner's understanding by non-verbal communication. Thus, the above taxonomy is also used in Viclle.

4.4. Learner's Feedback

If a learner cannot understand a teacher's explanation, a learner may ask the teacher questions. In video correction, the learner studies the corrected video by him/herself asynchronously, and the teacher is not present. Therefore, it is important for a teacher to receive the learner's feedback via some other mechanism. According to the above taxonomy, feedback to the teacher is provided with the following communication marks (Refer to the categories of Section 3.3):

- (1) *Opinion*: This is needed for Clarification requests and Confirmation checks. Using this type of mark a learner can ask a question and give a comment to the teacher.

- (2) *Reply*: This is needed for all categories. Using this type of mark a learner can reply to the teacher's questions.
- (3) *Repeat*: This is needed for Confirmation checks. The system automatically records which part of the video to be repeated by the learner, and sends it to the teacher. Therefore, a teacher can get an idea of the student's level of understanding. The teacher can see where a learner has repeated and reviewed an explanation.
- (4) *View*: This is needed for Confirmation checks. The teacher can see whether a learner views the corrected part of a video.

In video correction, a system can store a learner's study history because a learner has to study the corrected contents by computer. After studying, the learner can send the above information to the teacher. Thus a teacher can grasp a learner's understanding of the situation.

4.5. VCML

According to the above discussion, marks for support of structured pedagogical communication between teacher and learner are defined in principle. In this section, we formally define each mark in our XML schema called VCML. The advantages of VCML are summarized below:

- (1) Links between the original video and the corrections show the corrected video.
- (2) It is possible to use files via a network.
- (3) It generates the revised video by applying VCML tags to an original video (thus the original video is preserved).
- (4) It is possible to describe the learner's feedback.
- (5) It is independent of the computer operation system and software.
- (6) It can be sent via E-mail because of its text-based format.

An example of a VCML file is shown in Appendix 1.

- (1) *Profile part*: It describes the profile of the learner and the teacher in the original video.
- (2) *Correction part*: The corrected information is described. Correction marks are shown in Table 1. Those marks can use the media shown as seen in Table 2.
- (3) *Status part*: The learner's feedback is described in this part. The tags are shown as seen in Table 3.

Table 1. VCML tags for video correction.

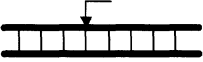


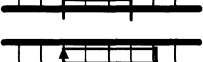
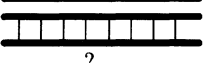
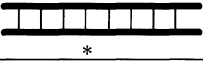

Correction	Status Mark	Contents
Insert		Inserting media
Overlap		Composing comments
Replace		Replacing with other media
Delete		Deleting the part
Repeat		Repeating the part
Pause		Pausing the point
Annotation		Inserting media as a comment

Table 2. Media used by VCML in correction part.

Correction	Text	Picture	Audio	Video	Web
Insert			✓	✓	✓
Overlap	✓	✓			
Replace			✓	✓	
Delete					
Repeat					
Pause	✓				✓
Annotation	✓	✓	✓	✓	✓

Table 3. VCML tags in status part.

Response	Contents
Opinion	Representing a learner's opinion
Question	Representing the answer from a teacher
Repeat	Recording a learner's repeating point

The VCML file consists of three parts, though the Status part is only added after a learner studies the teacher's corrections.

5. IMPLEMENTATION

We have developed Viclle on Windows 98/NT/2000. The development language is Java SDK1.3.1. Java Media Framework (JMF) 2.1.1a as a framework to deal with media, Java API for XML Parsing (JAXP) as a XML Parser, Regular Language description for XML (RELAX) to analyze XML documents and Relaxer as a parser of RELAX are used. Viclle has the following features:

- (1) The videoconferencing system in Viclle can record the learning scene.
- (2) The teacher can correct the video recorded by the videoconferencing system.
- (3) The learner's errors in the original video are marked by correction marks with VCML.
- (4) The learner can see exactly what the teacher has corrected. This is based on What I See Is What You See (WISIWYS).
- (5) The learner can study by interacting with Viclle.
- (6) The revised/explained video is generated if VCML tags are applied to the original video.
- (7) The teacher and learner share the corrected video as a teaching and study aid.
- (8) Learner errors are highlighted as correction marks by Video-based Correction Markup Language (VCML) based on eXtensible Markup Language (XML).

Viclle consists of an editor for teachers and a viewer for learners. The snapshot of Viclle-Editor is shown in Figure 4. In the editor, the teacher edits an original video by correction. The editor interface contains the following features: The original video is shown in the Correction Editor (A). Frames composing the video are shown in the Timeline (D). The Timeline is controlled by the Timeline Controller (C), and the teacher can get an index of the video by operating the Timeline Controller. When correcting the video, the teacher can select any of the correction marks on the Correction Palette (B). As an example, the replace correction is inserted in this interface. When the replace button is pushed, the Correction Dialogue (E) appears and the

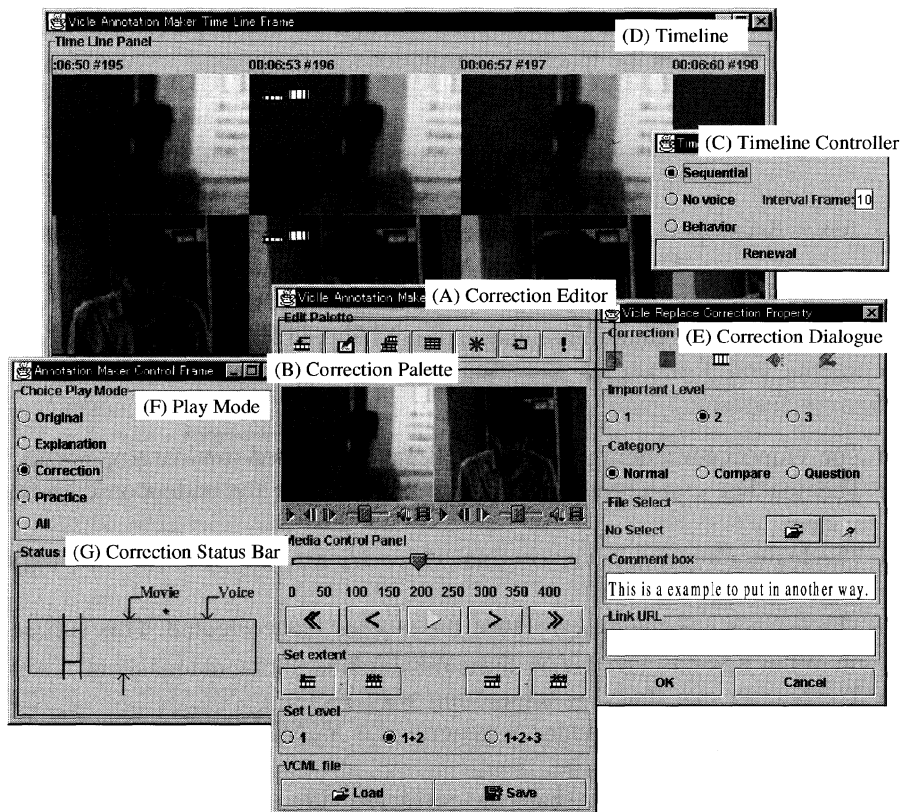


Fig. 4. Snapshots the Vicle-Editor.

correction is inputted. Similarly, other corrections are inserted. In order to view corrections, the Correction Status Bar (G) is used. In order to choose the play mode for corrections, the Play Mode (F) is used.

6. RELATED RESEARCH

There are some systems available that can insert comments into video. Lieberman et al. (1997) developed an annotation system for conventional knowledge acquisition. A domain expert interacts with a knowledge engineer, who interviews the expert, and codes knowledge about the domain objects and

procedures in a rule-based language, or other textual representation language. The research proposes a user interface that allows a domain expert who is not a programmer to construct representations of objects and procedures directly from a video of a human performing an example procedure. It can comment on the operation of the system but cannot represent implicit knowledge like language learning or non-verbal information because it is difficult for a teacher to express explanatory notes for pronunciation and intonation with audio.

A wide range of authoring systems support both text and media annotations. A representative example is the DIANE system (Benz, Bessler et al., 1997; and Benz, Fischer et al., 1997). The system uses the data recorded by an arbitrary application as the basic content of a multimedia document and allows it to be annotated with any media available to the user. In this system, the annotation framework is defined. It can cope with the ordinal video but cannot deal with two videos recorded by a system like a videoconferencing system. To date, there is no research into annotating videos from a videoconferencing system.

A multimedia annotation environment for young children using Java (Bouras et al., 2000) is used to provide children with the opportunity to reflect upon and annotate episodes from their everyday lives. This system facilitates learners' discussion through the inserted annotation. Moreover, it does not provide written instructions to aid learning.

The classroom 2000 (Abowd et al., 1996) project focuses on teaching materials. The project aims to create tools that will aid in the capture and subsequent access of classroom information, and will enhance both the learning and teaching experience. Therefore, the teacher can make teaching materials with various media and show it to the learner. The learner can take notes on the teaching material to enhance learning. However, the learner cannot study with a personal video.

MRAS (Barger, Gupta, Grudin, Sanocki, & Li, 2001) is a system for on-demand training and education by multimedia contents. It focuses on multimedia lecture scenarios and asynchronous collaboration. In this system, a learner can annotate the teaching material in a video when s/he studies a teaching material video. The annotations are stored on an annotation server and can be sent via e-mail. Therefore, a teacher can receive the annotations from a student through email or by accessing the annotation server. Moreover, if the annotation server is put on a network, the teacher can access when and where s/he likes. However, a learner can only annotate with text or audio, and not with structured marks such as VCML provides. Therefore, structured video correction is unsupported.

A Visual Language (Davis, 1995) to express the meaning of a point on video by icon has been proposed. In order to enable the search and retrieval of a video from large archives, a representation of video content is needed. Although some aspects of video can be automatically parsed, a detailed representation requires that video be annotated. The prototype system, Media Streams, enables users to create multi-layered, iconic annotations of the video content. Within Media Streams, the organization and categories of a "Director's Workshop" allow users to browse and compound over 2500 iconic primitives by means of a cascading hierarchical structure which supports compounding icons across branches of the hierarchy. This idea is used to show the situation of video correction, for example, a kind of video correction, information and so on.

MPEG-4 is a standard for multimedia for the fixed and mobile web. It is aimed at the coding standard of comprehensive multimedia, for example, video, audio, still pictures, computer graphics pictures, Musical Instrument Data Interface (MIDI), and text. For authors, MPEG-4 enables the production of content that has far greater reusability, has greater flexibility than is possible today with individual technologies such as digital television, animated graphics, World Wide Web (WWW) pages and their extensions. For network service providers MPEG-4 offers transparent information. For end users, MPEG-4 brings higher levels of interaction with content, within the limits set by the author. However, MPEG-4 cannot record a learner's situation. Moreover, the system cannot attach the semantics of the inserted media when a teacher edits a video. Therefore, it is difficult to change the contents for playback by the system.

MPEG-7 is an ISO/IEC standard that is developed by Moving Picture Experts Group (MPEG), formally named "Multimedia Content Description Interface." The aim is to define a standard for describing the meaning of multimedia contents. The multimedia contents can be retrieved quickly and efficiently by the descriptions. However, it does not provide a way to recognize pictures and voices. In addition, it does not provide the search engine for the descriptions. However, MPEG-7 focuses on the scene and retrieval. It can only deal with meta-format. Therefore, it cannot synchronize both a learner's and a teacher's video.

Synchronized Multimedia Integration Language (SMIL) is an enabling technology allowing simple authoring of interactive audiovisual presentations. SMIL is typically used for "rich media"/multimedia presentations which integrate streaming audio and video with images, text or any other media type. Therefore, SMIL makes a linear presentation stream. VCML is different from SMIL because VCML can give a non-linear and hyperlinked presentation

stream. For example, the learner can see only the original video and compare it with a revised one. Moreover, SMIL links video with timestamps, whereas VCML does it at the frame level for an easy video editing as in a video-editing tool, for example, MediaStudio (Ulead Systems Inc.) and Premiere (Adobe Systems Inc.).

Video-editing tools, for example, Premiere and MediaStudio, can also edit and create videos for explanation. However, the purpose is to create a new video after editing. Such a system can effectively show only the corrections made by the instructor, and all of the learner's original errors are lost in the correction process. In addition, it is difficult to describe the learner's feedback and to communicate changes between the teacher and learner by e-mail because the file becomes too large in size.

7. CONCLUSION

This paper presents a prototype video correction system called Viclle, a video-based language-learning environment that uses VCML to exchange corrections and feedback via e-mail. In video correction, a teacher can edit a video easily by applying text correction-like marks to it. There are some educational systems that use videos as teaching material or a videoconferencing system. However, to our knowledge no studies have ever tried to apply text correction concepts to video. Since the broad-band networks that enable people to exchange video data will be widely spread, the feasibility of failure of Viclle is very high.

In future work, we will evaluate Viclle. In the evaluation, we discuss the effectiveness of our video correction model with correction and communication marks. In addition, we develop the analysis and the accumulation system of the corrections. In the system, we discuss the way of analyzing correction and communication marks. We analyze corrections with the data extracted by new technologies (image/voice recognition system) from corrections. The system will be able to support the correcting teacher and explain the learner's problem while self-learning by agent or a voice recognition system.

ACKNOWLEDGEMENT

We are most grateful to the anonymous referees and the editor for many helpful suggestions and corrections of English sentences. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for JSPS Fellows, No. 14004123, 2002 and Scientific Research (B)(2) No. 13480047 from Japan Society for the Promotion of Science.

REFERENCES

- Abowd, G.D., Atkeson, C., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N., & Tani, M. (1996). Teaching and learning as multimedia authoring: The classroom 2000 project. In *Proceedings of the ACM Multimedia '96 Conference* (pp. 187–198). New York: ACM Press.
- Adobe Premiere (Adobe systems Inc.). specification. Available: <http://www.adobe.com/products/premiere/>
- Bargeron, D., Gupta, A., Grudin, J., Sanocki, E., & Li, F. (2001). Asynchronous collaboration around multimedia and its application to on-demand training. In *Proceedings of the HICSS-34 Conference* (10 pp.) (CD-ROM). Island of Mani: IEEE Computer Society.
- Benz, H., Bessler, S., Fischer, S., Hager, M., & Mecklenburg, R. (1997). DIANE: A multimedia annotation system. Proceedings of the Second European Conference on Multimedia Applications, Services and Techniques (ECMAST '97) (Milan IT, May 1997). In S. Fdida & M. Morganti (Eds.), *Lecture notes in Computer Science 1242* (pp. 183–198). Berlin: Springer Verlag.
- Benz, H., Fischer, S., Mecklenburg, R., & Dermier, G. (1997). DIANE – Hypermedia documents in a distributed annotation environment. Proceedings of the Conference on Hypertext – Information Retrieval – Multimedia (HIM '97) (Dortmund DE). In F. Norbert, D. Gisbert, & K. Tochtermann (Eds.), *Schriften zur Informatik* (pp. 293–306). Konstanz: UVK Universitaetsverlag.
- Bouras, C., Kapoulas, V., Konidaris, A., Ramahlo, M., Sevasti A., & Van de Velde, W. (2000). Using multimedia to support reflection on past events for young children. In *ED-MEDIA2000 Proceedings* (pp. 105–110), Montreal, Canada.
- Dalton, D.W. (1990). The effects of cooperative learning strategies on achievement and attitudes during interactive video. *Journal of Computer Based Instruction*, 17, 8–16.
- Davis, M. (1995). Media streams: An iconic visual language for video representation. In R.M. Baecker, J. Grudin, W.A.S. Buxton, & S. Greenberg (Eds.), *Readings in human-computer interaction: Toward the year 2000* (2nd ed., pp. 854–866). San Francisco: Morgan Kaufmann. (Also Available: <http://www.interval.com/papers/mediastreams/index.html>)
- Farkas K.D., & Poltrock, E.S. (1995). Online editing mark-up models and the workspace lives of editors and writers. *IEEE Transaction on Professional Communication*, 38, 110–117.
- Girgensohn, A., Boreczky, J., Chiu, P., Doherty, J., Foote, J., Golovchinsky, G., Uchihashi, S., & Wilcox, L. (2000). A semi-automatic approach to home video editing. In *Proceedings of UIST'00* (pp. 81–89). San Diego, CA: ACM Press.
- Hardock, G., Kurtecbach, G., & Buxton, W. (1993). A marking based interface for collaborative writing. In *Proceedings of ACM Symposium on User Interface Software Technology* (pp. 259–266), Atlanta, Georgia.
- Johnson, K., & Morrow, K. (1981). *Communication in the classroom: Applications and methods for a communication approach*. Essex, UK: Longman Press.
- Koike, I. (Ed.). *The second language learning*. Tokyo: Daisyuukan publishing (in Japanese).
- Lieberman, H. (1997) *A user interface for knowledge representation from video*. Available: <http://lieber.www.media.mit.edu/people/lieber/Lieberary/Mondrian/Knowacq/Knowacq.html>

- Ogata, H., Hada, Y., & Yano, Y. (2000). CoCoAJ: Supporting online correction of hypermedia documents for CALL. *ICCE 2000* (Vol. 1, pp. 323–329). Taipei, Taiwan 2000.
- Ogata, H., Feng, C., Hada, Y., & Yano, Y.: *Online* markup based language learning environment. *International Journal of Computers & Education*, 34, 51–66.
- Java API for XML Parsing specification. Available: <http://java.sun.com/xml/jaxp/index.html>
- Java Media Framework (JMF) specification. Available: <http://java.sun.com/products/java-media/jmf/index.html>
- Java SDK specification. Available: <http://java.sun.com/j2se/1.3/>
- MediaStudio (Ulead Systems Inc.) specification. Available: <http://www.ulead.com/msp/runme.htm>
- MPEG4/7 specification. Available: <http://www.cseit.it/mpeg/>
- Relaxer specification. Available: <http://www.asahi-net.or.jp/~dp8t-asm/java/tools/Relaxer/>
- SMIL specification. Available: <http://www.w3.org/TR/smil20/>
- XML specification. Available: <http://www.w3.org/XML/>

APPENDIX 1: A PART OF VCML FILE

```

<?xml version = '1.0' encoding = 'ASCII' ?>
<!DOCTYPE VCML SYSTEM "VCML.dtd">
<VCML>
<Profile><Learner e-mail = "hada@it.is.me" name = "hada" url =
    "learner.avi"/>
    <Teacher e-mail = "ogata@it.is.me" name = "ogata" url =
    "teacher.avi"/>
    <Editor e-mail = "ogata@it.is.me" name = "Hiroaki Ogata"/>...
</Profile>
<Correction><Annotation media = "Charactor" category = "Revision"
    start = "90" level = "2" comment = "Your pronunciation is wrong."
    url = "annotation.avi"/>
    <Insert media = "Voice" category = "Advice" start = "100" level =
    "1" comment = "This is an example." url = "voice1. wav"/>
    <Replace media = "Movie" category = "Revision" start = "20"
    end = "40" level = "2" comment = "This is another expression."
    url = "replace.mov"/>
    <Overlap media = "Charactor" category = "Revision" start = "60"
    end = "70" level = "2" comment = ""
    px = "0" py = "50" string = "Don't use [the]."/>... </Correction>
</Status><Question frame = "120" string = "What do you mean?"/>
    <Repeat start = "110" end = "150"/>
    <Answer frame = "188" string = "The"/>...</Status>
</VCML>

```