Synote: development of a Web-based tool for synchronized annotations

Yunjia Li\textsuperscript{a}, Mike Wald\textsuperscript{a}, Gary Wills\textsuperscript{a}, Shakeel Khoja\textsuperscript{b}, David Millard\textsuperscript{a}, Jiri Kajaba\textsuperscript{a}, Priyanka Singh\textsuperscript{a} & Lester Gilbert\textsuperscript{a}

\textsuperscript{a} Learning Societies Lab, School of Electronics and Computer Science, University of Southampton, Southampton, UK
\textsuperscript{b} Institute of Business Administration, Karachi, Pakistan

Available online: 24 Aug 2011
This paper discusses the development of a Web-based media annotation application named Synote, which addresses the important issue that while the whole of a multimedia resource on the Web can be easily bookmarked, searched, linked to and tagged, it is still difficult to search or associate notes or other resources with a certain part of a resource. Synote supports the creation of synchronized notes, bookmarks, tags, links, images and text captions. It is a freely available application that enables any user to make annotations in and search annotations to any fragment of a continuous multimedia resource in the most used browsers and operating systems. In the implementation, Synote categorized different media resources and synchronized them via time line. The presentation of synchronized resources makes full use of Web 2.0 AJAX technology to enrich interoperability for the user experience. Positive evaluation results about the performance, efficiency and effectiveness of Synote were returned when using it with students and teachers for a number of undergraduate courses.

Keywords: Synote; Synchronisation; Open hypermedia; Multimedia annotation; Tags

1. Introduction

This paper discusses the development of a Web-based media annotation application named Synote. With the advent of video storage sites such as YouTube and Blinkx, it has become extremely easy for Web users to upload and share their videos. However, there is still the issue that it is difficult to find or associate notes or other resources with a certain PART of a resource.

To handle these vast data archives, video storage sites usually provide linking, tagging, note-taking and bookmarking services, which make the resources easier to access and search for web site users. However, most of the services are provided in the level of the WHOLE media resource, and so it is still problematic when considering the efficiency and accuracy of the
annotation. For example, on YouTube, users can easily find a whole video clip by searching the tags provided by the uploaders, but on many occasions, users are not interested in the whole content, but a particular segment within the video. From the view of user experience, if you search for videos by entering a certain word or phrase, then a click on a search result will open a new recording and it will be played from the very beginning, not from the time the word or phrase was spoken or refers to. This means that users have to listen to the whole recording to find the fragments that match the word or phrase they search. As an analogy, a textbook with no content page, index or page numbers will clearly be difficult to use. The problem described above can be summarized as follows:

*On the Web, a certain PART of a continuous multimedia resource is difficult to be bookmarked, linked to, tagged and searched.*

The “Video in the Web” activity has been trying to make the continuous multimedia “first class citizen” of the Web. Synote implements some of the requirements (Troncy and Mannens 2009) in that it enables people to create, navigate, search and share video or audio fragments through annotation content.

This paper explores application development aspects of the idea of making synchronized annotations for continuous multimedia resources. System requirements are listed in Section 2. Section 3 of this paper reviews some old and new applications that address similar problems as does Synote. Section 4 introduces how the Synote Hypermedia Model (SHM) is developed at the conceptual level of Synote. Section 5 discusses the design and implementation of Synote and is followed by a summary of unique features for Synote in Section 6. Section 7 presents the evaluation of the usage of Synote in the education environment.

2. Summary of requirements

A summary of the main Synote specifications to meet the user requirements is as follows:

- **R1**: There are several types of resources users require: continuous multimedia resource (audio and video), transcript text (caption), tag, note (can contain URL or even valid HTML code), bookmark and presentation slide.
- **R2**: Non-time-based resources (such as transcript, tags, notes, bookmarks and presentation slides) should annotate time-based resources (audios and videos), but not the other way round.
- **R3**: All the annotations in R2 should be synchronized.
- **R4**: When the audios and videos are replayed, all the other synchronized resources should be highlighted in some manner.
R5: All the synchronized resources can be “selected”. When a resource is selected, all the other resources synchronized with this resource should be “selected” too.

R6: Users should be able to create, edit, delete a transcript, tag, note, bookmark, or presentation slide when audios and videos are playing.

R7: Text-based resources (titles of the multimedia resources, tags, notes, transcript, etc.) should be indexed in order to search the multimedia resources they annotate.

R8: If a search result (a tag for example) in R7 is synchronized with an audio or video, users should be able to view the audio or video directly from that synchronization point (rather than only from the very beginning) when they select the search result.

R9: There should be a permission control over all resources and annotations.

R10: User interface needs such as resizing frame and auto-scrolling.

R11: Support uploading transcript from other resources in other formats.

R12: Analyse unstructured tags (folksonomy) to help create structured tags (ontology).

R13: Use speech recognition to generate transcript.

3. Related work

Annotation in hypermedia is a means of marking up the objects and scenes in audio–video streams in order to facilitate the interpretation and understanding of its content. In the last two decades, many systems started to treat readers as annotator or authors of resources as well. Currently, the use of annotation is divided into two main categories: metadata association and content enrichment. Metadata association methods use specific metadata models and content-based retrieval (CBR) data to build a semantic structure, which supports features such as content search. Examples of such systems are EXIF (Japan Electronics 2002), ID3⁴ and MPEG-7 (Martinez 2002).

In order to gather metadata from users through collaboration in Web 2.0 era, many Web-based applications emerge as collaborative annotating tools. Mimicry system (Bouvin and Schade 1999) is one of the efforts that allow authors and readers to link to and from temporal media (video and audio) on the Web. Annotea (Kahan et al. 2002) is another example of a hypermedia system to create annotations on Web documents. It uses an Resource Description Framework (RDF) schema for defining annotations. One of the examples of Annotea implementation is Vannotea (Schroeter 2006). Web applications such as Overstream⁵, Tegrity⁶ and Ponopto⁷ enable users to enter text synchronized with media on the Web. YouTube has also introduced the facility of annotating videos, along with writing comments and star rating the videos⁸. Everyzing⁹ uses speech recognition to create a searchable transcript from podcasts or multimedia recordings. However, Everyzing does not display fully synchronized captions.
Video in the Web activity aims to improve multimedia content discovery, searching and indexing on the Web. The groups in this activity (Timed Text Working Group, Media Fragments Working Group and Media Annotations Working Group) have published ontologies\(^{10}\) for media annotations. They have also published a client-side API\(^ {11}\) to access metadata information.

4. Synote Hypermedia Model

In this section, the conceptual level model, Synote Hypermedia Model (SHM), is presented. SHM follows the Fundamental Open Hypertext Model (FOHM (Millard et al. 2000), see Figure 1), which defines a common data model for different hypertext domains (navigational hypertext, spatial hypertext and taxonomy hypertext). The main structure that SHM tries to explore is the navigational structure. The entities on the conceptual level are defined with inspiration from FOHM. In the presentation level, emphasis is given to presenting a user-friendly interface that better reflects the entities in the conceptual level.

4.1. From user requirements to the conceptual model

Figure 1 shows an example of FOHM navigational structure. In this picture:

- **Resources**: The “Node” entity is referred to as multimedia resource (R1). All resources have a globally unique identifier and are external to the model.
- **References**: They are anchors pointing to a node or part of a node. The location specification (Loc Spec in Figure 1) indicates which part of the node this reference is pointing to.
- **Associations**: They are structures that represent relationships between nodes.
- **Bindings**: Bindings specify the attributes of the connection between Association and Node through Reference. The bindings are directional,
and the nodes are bound with a “source”, “destination” or “bi-directional” value.

If the presentations of these entities are not considered, the navigational behaviour is easy to describe: if Reference1 is activated, then Reference2 will be activated following Association1 and the “[src]” and “[dest]” bindings. Then, for the same reason, Reference3 is followed. This navigational structure is rather flexible in that Associations can share References and References can share Nodes. So the bi-directional and multi-directional links are easy to achieve in this structure.

The environment SHM used is different from that used by FOHM, especially when more presentation level requirements are involved.

Firstly, R2 indicates that every annotation has a direction; that is audios and videos are always the resources to be annotated. This means the Reference, which the destination binding links to (destination Reference for short), must refer to a continuous multimedia resource (referred to as “target resource”), while the source Reference must refer to a resource other than a continuous multimedia resource (referred to as “source resource”). In this case, every association has only two bindings. But it is possible that there are many associations between two nodes.

Secondly, if destination Reference must point to a continuous multimedia resource, the Loc Spec for destination Resource must describe a span in time line. The Loc Specs for source References must be varied according to the nature of the nodes they refer to. So in terms of semantics, every annotation is annotating the time line. But users of Synote may not realize this fact because time line is an abstract concept and visualization of time line is the time in multimedia resources. R4 and R5 describe the navigational behaviour of “following a link”, and they imply that every association should be bi-directional.

Thirdly, R1, R4, R5 and R6 suggest that we should choose clear and understandable presentations for each type of resource. Our solution is to design high-level metadata storage as Synmark (Synchronized bookmark). Synmark has a unique ID, title, notes that can contain valid HTML code, tags separated by spaces and a link to the next Synmarked part of the recording.

Four types of media resources have been identified: Multimedia resource, Synmark resource, Transcript resource and Presentation (Slide) resources. So Loc Spec needs to be defined for each type of resource. For Multimedia resource, the Loc Spec can refer to the time span in milliseconds. Synmark is a compound resource without any notion of time, and so a start and end time can be added as the Loc Spec. The word count and index number can be used for transcript and presentation resource, respectively. After tailoring the FOHM with our identified user requirements, the conceptual model can be described as the following (Figure 2).

One example of using this model is shown in Figure 3. In this example, if TR2 is activated, following Association2, MR2 will be activated. As the
bindings for Association3 and Association4 are bi-directional, SR1 and PR1 can be activated. If MR2 is firstly activated, TR2, SR1 and PR1 can be followed.

4.2. Realization of Synote Hypermedia Model

This stage considers how to simplify the design and provide a clear presentation for users.

In Figure 3, it is easy to notice that one TR, SR and PR can be related to only one Association, but MR can be linked to many associations. The

Figure 2. Synote conceptual model.

Figure 3. Synote conceptual model example.
conceptual model does not restrict that one TR, SR and PR cannot be linked to many associations. However, in the presentation, it does not make sense if two MRs (e.g. MR2 and MR4) are activated or played at the same time.

Another interesting question is: is it possible to reuse MR? It is unreasonable to expect two users to annotate exactly the same period of time as time itself is not a discrete concept. It should not be expected that MR is “reuseable”. So it is reasonable to create an MR every time a new annotation is made. However, this will break the navigation behaviour as one MR cannot link to multiple associations. The solution Synote adopts is a controller that decides which link(s) should be followed (Figure 4).

In Figure 4, two new concepts are introduced. Every Association has two bi-directional bindings, and so it is not necessary to explicitly keep Association and Bindings as separate entities. Synote introduces Synpoint (synchronization point) as the combination of two References. It stores the start and end time for the target resource and the start and end index for the source resource. ResourceAnnotation defines the relationship between two resources as one resource annotates the other. In the real use of Synote, Multimedia Resource usually is the one to be annotated, but the conceptual model leaves the possibility that other types of resources can also be annotated. One ResourceAnnotation can have one or more Synpoints. It can be argued that Synpoint means the “Annotation” as our common understanding, but Synote treats “Annotation” as a collection of synchronized points.

Figure 4. Synote conceptual model realization example.
When TR2 is activated, the start and end time of MR2-1 is passed to the MultimediaController. Then, the controller set the time position for M1 to 2:10. As the position of M1 changes, MultimediaController will decide which Reference should be followed. 2:10 is within the time span of MR2-3, but not MR2-2. So MR2-3 and PR1 should be activated. When the time passes to 2:12, MR2-2 and SR1 will be activated. MultimediaController makes the semantic of following “multi-directional links” clear in Synote. Even though all the links (associations) are bi-directional when created by users, one link can activate many other links via MultimediaController in this way.

5. Synote design and implementation

This section will focus on the design from the object model (database) to the presentation level.

5.1. Synote object model design

Figure 5a and 5b shows the basic object model of Synote, in which Annotation and Resource are the main classes.

The hierarchy of Resource classes represents different Resource entities in SHM. A Single Resource can be regarded as “meta resource”. A Compound Resource will contain multiple occurrences of single resources. Both Compound Resource and Single Resource are direct subclasses of Resource. This structure can benefit the search of content (R7) and provide a better basis for presentation specification design.

Annotation and Resource tables are bundled and every ResourceAnnotation contains one or more Synpoints. In the design, any media is allowed to annotate any other media. For example, a Synmark can annotate a video clip, or vice versa. But in the real use, audios and videos are usually the resources to be annotated.

5.2. User interface design

In order to display multimedia resources and links in a synchronized manner and allow user interactions within permission control (R4, R5, R6 and R9), we designed a heavy Javascript user agent, named Synote Player (Figure 6), to implement “following a link” behaviour. Synote Player user interface comprises four parts: player module, transcript module, Synmark module and presentation module.

The player module main consists of an embedded player and the control buttons. The presentation of the link is the time line in the embedded player. The link can be followed by a time trigger when an audio or video is playing, or user selecting a certain time point. The transcript module maintains the synchronization of captions with media, and the spoken words are highlighted. Each word (or block of words) is synchronized with the time line. Clicking on a word (or word block) will lead to a change in current playing
Figure 5. Synote Object Model: (a) Resource. (b) Annotation.
time via the start time of the Synpoint. This facility also allows users to edit and re-synchronize the transcript text while playing the media. The Synmark module displays a list of Synmarks. Every item in this list is clickable, and clicking on one Synmark will reset the time to the point that Synmark

Figure 5. (Continued)

Figure 6. Synote interface design graph.
annotates. Users can use a “Synchain” to sequence Synmarks in a linear fashion in order to view only those marked parts of a recording. The slide module displays the related slides, all of which will have start and end times, so that the current slide will automatically be highlighted by the system. When a video or audio is playing, different parts in those modules will be highlighted via Synpoints, but no new page or presentation will be opened.

5.3. Backend of Synote Player

Synote Player is designed to have its own runtime model in order to perform like a powerful user agent to deal with user interactions. Whenever an event happens, the runtime model will notify other related resources to jump to synchronized points. Then the link will stay still until other events happen. The event here can be divided into two categories. One is user interaction event, which is performed when users click links. The other one is a time-triggered event, which is performed every certain time interval (one second for example) as the audio or video is playing.

Every clickable event is registered in MultimediaController. When an event is triggered, MultimediaController will set the time position of the embedded player and select the proper item to be highlighted in other models in Synote Player. It feels like a multi-directional link is followed when any head of this link is clicked, but actually, when users create the annotation, what they create are bi-directional links.

5.4. Implementation

Synote has been developed using Grails framework\textsuperscript{12}, and it is a typical Browser-Client Web application (Figure 7). Different from other pages,
Synote Player is an Ajax application written in Google Web Toolkit (GWT). Using GWT, some Javascript runtime classes are generated to present the player interface and realize different runtime events (Figure 8).

The database stores links of external media (e.g. video, audio and image slides) and Synmark, transcript resources and annotations. External media are stored in user-owned or specified online spaces. When replaying this media, the media stream will be read directly from this link by the embedded media player. The classes mentioned in Figure 5a and 5b are defined as domain classes in Groovy. These domain classes are mapped into MySQL database via Hibernate.

Synote Web server provides services and dynamic Web pages for Synote users to manage personal information, permissions and multimedia resources. GWT plugin for grails also requires the server exposes service interface for Synote Player. Figure 9 shows actual screen displays of Synote. Synote Player will be introduced in the next section.
Figure 9. Synote recording list screenshot.

Figure 10. Synote Player interface screenshot.
In the browser on the client side, end users can replay the recordings and all the synchronized resources in Synote Player (Figure 10). Considering that the heavy Javascript implementation of Synote Player may result in accessibility issues, a print-friendly version of the media and synchronized resources is also implemented.

In Synote Player, when the recording is played, the currently spoken words are shown highlighted in the transcript (Figure 10). Selecting a Synmark, transcript word or Slide/Image moves the recording to the corresponding time (R4, R5). Toggling Autoscroll can ensure that the current selection stays visible in the panel or allows user to scroll manually (R10). All the four widgets can be resized by users to fit their needs. The start time is automatically entered as the time in the recording corresponding to when the Synmark was created. Users can choose to display their own Synmarks, owner’s Synmarks or all Synmarks, but they can edit or delete only their own Synmarks. Users who are entitled to edit the transcript can edit the transcript manually or upload a.srt caption file within Synote Player (R6, R11). Figure 11 shows the transcript editing module. The time icons are used to display the synchronization points and moving the mouse over them will display the synchronized time. A mouse click on transcript text brings it to the edit box, where the user can edit the transcript text, whereas clicking at the spaces between words will create synchronization points in the transcript using the current media time.

In the print-friendly output, information for the whole time period of the recording or only part of it can be printed and the user can choose to print information for the whole of the recording within this time period or only the Synmarked parts of it and select whose Synmarked parts. The user can also choose whether to print the transcript, presentation slides/images and the Synmarks, as well as information in the Synmarks.

Another distinguished function of Synote is the searching services (R7, R8). All the content-based resources, that is subclasses of TextResource, are

![Figure 11. Transcript editing screenshot.](image-url)
indexed. Figure 12 shows the searching result for the term “guide”. Every searching result may consist of the recording title, Synmarks and transcript text, in which the hits are highlighted. Different from the search results provided by YouTube, clicking on a Synmark or a block of transcript text in the search results will open the Synote Player and the recording will start playing from the time the Synmark or text block annotate. Synote also provides a Google-like advanced search function. The search results are behind the permission control, and logged in users can search their own recordings, which are not publicly available.

Usability and accessibility issues are considered in every stage of design. Currently, Synote Player can play most media formats available on the Web, such as WAV, MP3, AVI, MPEG and flash videos, through the use of Windows Media Player, JW Player and QuickTime Player on both Windows and MAC OS platforms. Synote will choose a proper media player according to the format of the file and the client platform. Keyboard shortcuts are supported so that users can accomplish most of the interaction operations with Synote Player without touching the mouse. Users can use keyboard shortcuts to play/pause/stop/rewind media player, edit transcripts and create Synmarks. The display of Web pages and other interaction operations have been tested in Internet Explorer, Firefox, Google Chrome and Safari browsers. Synote also has a plan to support HTML5’s <video> and <audio> tag in the future.
There are also some extended functions for the convenience of resource generation. A tool has been developed to automatically create Synote presentations from narrated PowerPoint files and searchable Synmarks from the slide title, text, notes and timings. It can create a synchronized transcript if this is typed into the PowerPoint slide notes. Synote also works with IBM Hosted Transcription Service\(^19\) (for licensing reasons this function is not public outside of the University of Southampton). When users create a new recording, they have to provide the URL of the audio or video file. Then the transcribing service can take the file from the URL and automatically generate the transcript (R13). After the generation is finished, the transcript can be uploaded by the recording creator. To enable users taking live notes, Synote also integrates with Twitter\(^20\) through Twitter API. Users can send twitter messages while the live lecture is going on. After the lecture, the Twitter messages will be uploaded as Synmarks.

6. Discussion about the design and implementation

Synote is not just a research tool or demo. It is a real Web application that can be and has already been used by many users. At a conceptual level, sometimes, the flexibility of a structure leads to difficulty in its presentation and user experience. For example, multi-directional link is a common structure on the conceptual level, but issuers will be confused if you ask them to create a multi-directional link in the Web environment, because they have been used to the presentation of single-directional links in Web pages. Our solution is to insert a runtime layer to perform the navigational behaviour. Synote is a good attempt to bridge the conceptual level and the presentation level.

There are many applications that have some features similar to Synote, as has been mentioned in Section 3. Compared with commercial products, such as Tegrity, Ponopto and Echo360\(^21\), Synote is freely available for everyone to use immediately at www.synote.org. Other developers can customize and integrate it with existing and future systems such as Matterhorn/Opencast.\(^22\) Websites such as YouTube offer synchronized annotation services, but currently these annotations are not available for search.

7. Brief evaluation results

Synote has been developed and evaluated with the support of JISC\(^23\) and Net4Voice\(^24\) and with the involvement of students and teachers who have used Synote for learning and teaching in their existing courses rather than in artificial experiments. Very enthusiastic and positive feedback has been received that supports the identified benefits to users.

Generally, this evaluation showed that students liked using Synote and found the synchronized transcripts and note-taking facilities very useful and wanted more recordings and lectures to be available in this way. The design of Synote to provide synchronized slides, video, audio and transcripts was highly appreciated. Non-native speakers in particular commented how valuable Synote was...
for them as it was sometimes difficult to understand lecturers’ speech and note-taking in a foreign language was very difficult for them.

In the evaluation, students also suggested that it would be useful to be able to create Synmarks via mobile systems (e.g. laptops or phones) using a mobile version of Synote as well as by using Twitter. Through mobile devices, it is possible to extend Synote to m-Learning (mobile learning) and integrate it with some mobile learning toolkits, such as mPLAT (Pei et al. 2008). Synote now is also being used to annotate interviews for research purposes and monitoring students’ recorded progress in nursing or health care professions. However, it can be expected that Synote will be widely used in other areas and fields in the future.

8. Conclusion

This paper has described the Synote media annotation system. In Synote, users can create synchronized notes, bookmarks, tags, links, images and text captions on a fragment of video or audio, so that the searching results can link directly to the accurate time span of the resource. Ajax technology in Web 2.0 is widely used in the implementation of Synote Player, so that all the links can be followed without refreshing the player. Synote also works with various operating systems and browsers and provides many ways to deal with accessibility issues. Evaluations have shown that most users highly appreciate the functions Synote provides. Synote is available at http://www.synote.org for any author to contribute resources and annotations.

Notes

[8] http://uk.youtube.com/t/annotations_about
References


