Technical Disclosure Commons

Defensive Publications Series

February 27, 2019

Turning static media streams into interactive experiences

Tiruvilwamalai Raman

Turning static media streams into interactive experiences

ABSTRACT

Media such as podcasts or other audio, videos, etc. are linear content streams, e.g., the only navigation generally available to users is the ability to rewind or fast-forward along the time axis. This is in contrast to documents on the visual web, where hypertext languages enable flexible and interactive navigation. This disclosure describes techniques to create lightweight annotations around media content to enable an interactive user experience over such content.

KEYWORDS

- hypermedia
- annotation
- podcast
- smart speaker
- smart display
- linear content
- linear media
- interactive media
- media stream

BACKGROUND

Media such as podcast episodes, radio programs, video podcasts, video news briefs, etc. are linear content streams, e.g., the only navigation generally available to users is the ability to rewind or fast-forward along the time axis. This is in contrast to documents on the visual web, where hypertext languages enable flexible and interactive navigation.

DESCRIPTION

This disclosure describes techniques to create lightweight annotation structures around media content to enable an interactive user experience over such content. The techniques can be used, e.g., by media publishers, content providers, etc., such that media consumers can navigate the content by asking search-style queries, e.g.,

- What did the radio program have to say about the weather?
- Go to sections of the podcast that talk about coffee.
- What did this series of podcasts have to say about health and coffee?

Per the techniques, a given media stream, e.g., in a format such as .mp3, .m4a, .mp4, is annotated with an underlying navigational structure referred to as hypermedia. The hypermedia structure can be conceptually visualized as a collection of hyperlinks analogous to a list of HTML anchor elements. The hypermedia enables navigation to portions of an audio or video stream based on spoken input provided by a user. A software application such as a conversational assistant facilitates user interaction based on hypermedia metadata.

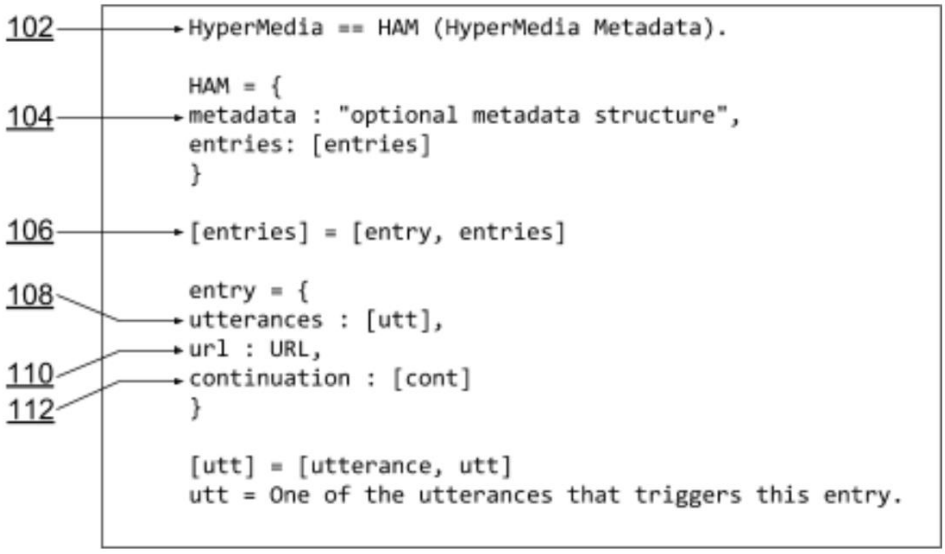

Fig. 1: An example hypermedia structure defined in terms of a JSON serialization

Fig. 1 illustrates an example hypermedia (.ham) structure defined in terms of a JSON serialization. The hypermedia structure (102) comprises optional metadata (104) for generating the initial introductory prompt, followed by a list of entries (106). An entry in the hypermedia structure defines the following:

- a list of utterances that trigger the entry (108);
- a URL that addresses the content to be played upon triggering (110), where a URL can address an entire (self-contained) piece of media, e.g., podcast episode, or part thereof;
- an optional continuation specification that defines the action to be performed after playing the specified content (112);
- a portion of an episode by using start and end timestamps encoded as URL parameters;
- any other URL parameters as decided by the publisher to segment the content, e.g., a radio station may have various archived programs as segments; etc.

Although the example of Fig. 1 illustrates a hypermedia structure in terms of a JSON serialization, the hypermedia design is itself serialization agnostic. Consuming the hypermedia structure results in an interactive experience where the user can move around the various pieces of audio addressed by the hypermedia structure using a dynamically generated conversational interface. The prompts listed in an entry of the hypermedia structure, together with the state of the dialog so far, are used to generate relevant prompts and grammars at each turn in the dialog.

A .ham structure can be published on the web at a given URL and referred to from an RSS or atom feed. A conforming implementation e.g., a virtual assistant action that is hypermedia-aware, can retrieve the annotation for a given media, e.g., podcast episode. Upon retrieval, the action consumes the metadata to do one or more of the following.

- Generate an initial summary or introductory prompt, e.g., play a short jingle; utter "welcome to today's edition of the podcast, where you'll hear about <list-of-topics>"; etc.
- If user requests one of the topics from the list of topics stated at the introduction, play the content at the URL in the matching entry. Here, a match occurs when a user-specified topic appears in the list of utterances.
- Play selected segment to completion, or until interrupted by user. A user can interrupt a segment by uttering a stop-word, e.g., "stop."
- Play a new or continuation prompt constructed from the continuation element in the current entry and from the set of remaining segments.
- Play a short jingle.
- If user doesn't provide input, play the remaining segments in sequence.

The hypermedia structure is tailored to information access via voice command, as enabled for example by virtual assistants, smart speakers, etc. The hypermedia structure can also be used to generate navigational affordances that are not limited to voice interaction, e.g., on mobile devices, wearables, on multimodal surfaces such as smart displays, etc. The hypermedia structure can be used to navigate the audio-web, which ranges from dynamically assembled audio feeds to machine-generated spoken information. The hypermedia structure relieves the user from having to consume a media item in its entirety, instead tailoring the media to the amount of time available to the user or topics of interest to the user. Alternative to hypermedia, a conversational dialog can be implemented that is specific to a given collection of media content.

Use-case example: Dynamic assembly and delivery of news shows

Today, several radio stations have websites that provide the broadcast of the day as well as archived content dating back years. These are typically accessed, e.g., by a REST API. A

given radio program of a radio station is often divided into segments, each of which has a stable URL. This form of decomposition, along with the associated metadata, makes it possible to dynamically generate a playlist as a response to example queries of the form:

- What does today's radio broadcast have to say about topic `<topic>`?
- What did this radio station say about topic `<topic>` through the month of December?
- What did the radio station say about topic `<topic>` across its various programs?

For each of the above, one or more custom playlists can be constructed, and a conversational interface can be provided that enables users to navigate the content.

Use-case example: Creation of dynamic audio content

By creating stable URLs to various segments of a given audio program, content creators can make their audio content ready for dynamic publishing, and create multiple audio views of the content they publish. An example is as follows. A broadcaster creates a media item that includes a one-hour interview with a famous personality, where the interview comprises of twenty questions. Individual questions and answers are assigned stable URLs. Longer responses are themselves segmented, with individual segments receiving their own URLs.

A content editor creates different snapshots of the same interview that range in length from two minutes to twenty minutes. The publisher enables searching over the metadata for these segments (which might include full transcripts). Users can obtain custom snapshots of the one-hour interview by performing search queries that operate on this metadata; e.g., what did the interviewee say about a particular topic.

Use-case example: Generation of custom audio views of existing web content

With the hypermedia structure described herein, custom views of audio content can be generated. In a manner similar to the visual web, which allows a user to reflow content on the

space axis, the audio web allows a user to reflow content on the time-axis. Thus, various views, e.g., a 5-minute summary-view, a 30-minute commute-specific view, etc., of a one-hour celebrity interview can be created, based on user needs.

Use-case example: Searching and exploring the audio web using content semantics

With the hypermedia structure described herein, structured overlays can be created over audio content, e.g., a lightweight tagged markup that annotates a time-linear audio stream with metadata. Such overlays and annotations can be created by different authors, e.g., a publisher, a speech-to-text transcriber, etc., and can be combined. Since the various annotation streams are synchronized over a single time-linear audio stream, it is possible to move between these annotations. As an example, a search can be performed over the results of speech-to-text transcription for a key-phrase to find the associated timestamp and utilize the metadata annotations provided by the publisher to go to the beginning of the segment that contains the key-phrase.

In this manner, the techniques of this disclosure turn time-linear static media streams into interactive experiences that enable users to move among various segments of the media using voice or other means of interaction.

CONCLUSION

Media such as podcasts or other audio, videos, etc. are linear content streams, e.g., the only navigation generally available to users is the ability to rewind or fast-forward along the time axis. This is in contrast to documents on the visual web, where hypertext languages enable flexible and interactive navigation. This disclosure describes techniques to create lightweight annotations around media content to enable an interactive user experience over such content.