IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 October 2022, accepted 22 November 2022, date of publication 1 December 2022, date of current version 8 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226367

RESEARCH ARTICLE

Supporting Targeted Advertising in Integrated Broadcast-Broadband Systems With Automatic Media Content Preparation

MARINA IVANOV P. JOSUÉ[1], MARCELO F. MORENO[2], AND DÉBORA C. MUCHALUAT-SAADE[1]

[1]MídiaCom Laboratory, Universidade Federal Fluminense, Niterói, Rio de Janeiro 24210-240, Brazil
[2]Department of Computer Science, Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais 36036-900, Brazil

Corresponding author: Marina Ivanov P. Josué (marinaivanov@midiacom.uff.br)

This work was supported in part by the Coordination for the Improvement of Higher Education Personnel (CAPES), CAPES PRINT, National Council for Scientific and Technological Development (CNPq), and in part by the Carlos Chagas Filho Foundation for Research Support in the State of Rio de Janeiro (FAPERJ).

ABSTRACT Integrated Broadcast-Broadband (IBB) Systems enable a range of new services for viewers, including content personalization, such as targeted advertising. Custom advertising is delivered using the broadband connection synchronized with the broadcast content, demanding high-control synchronization among both communication channels. As the broadband channel is shared among different applications, delays may occur during the transmission of targeted advertisements, which can cause synchronization faults. Aiming at reducing these faults and truly supporting targeted advertising in IBB systems, this work proposes the automatic preparation of media content before its presentation. In order to provide automatic preparation, a preparation plan to orchestrate content loading in advance is also proposed. Our proposal is implemented in the Ginga-NCL middleware, which is a standard for the Brazilian Digital TV System. Our performance evaluation shows that the average switching time from broadcast to broadband is around 40 ms, if automatic content preparation is used, which is an outstanding result.

INDEX TERMS Content preparation, integrated broadcast-broadband systems, DTV applications, multimedia synchronization, targeted advertising, ginga middleware.

I. INTRODUCTION

Some digital television (DTV) systems [1], [2], [3] allow content delivery using both broadcast and broadband connections. These systems are named Integrated Broadcast-Broadband (IBB) Systems [4] and enable a range of new services for viewers, including personalized advertising content. In targeted-advertising IBB applications, a personalized content should be presented to the viewer during broadcasted commercial breaks, precisely replacing one or more of the broadcasted advertisements. Customized advertising is delivered using a broadband connection and synchronized with the broadcast content. Therefore, this type of application demands high synchronization control between such different communication channels.

In general, IBB-enabled DTV receivers contain a component responsible for some tasks as multimedia application loading and execution, user interaction control, management of media players, and switching between broadcast and broadband transmissions. In Brazilian digital television system, those tasks are performed by a multimedia formatter (presentation engine) present in the Ginga [5] middleware. Ginga is the standard middleware of the Brazilian Digital Terrestrial TV System (SBTVD) and ITU-T Recommendation for IPTV (Internet Protocol Television) [6] and IBB systems [4]. In its current version [7], the Ginga middleware supports the execution of multimedia applications specified using the Nested Context Language (NCL) [8] and HTML5 [9].

Synchronization among media objects in a multimedia application can be specified through several approaches, such as event-based and timeline paradigms [10]. In the

The associate editor coordinating the review of this manuscript and approving it for publication was Usama Mir.

timeline paradigm, presentation moments of media objects that compose the application are disposed along a timeline, using absolute time values. On the other hand, in the event-based synchronization, media object presentation moments are not explicitly defined, but defined as set of inter-media relationships. NCL is an example of language that specifies media synchronization using the event-based paradigm. NCL supports different event types, such as the media presentation event, the attribution event that occurs when a value is assigned to a variable, and user interaction events.

Different from broadcast transmission where the content is sent using a dedicated, unidirectional channel, broadband transmission is a two-way communication channel shared among different applications. Furthermore, broadband data delivery is subject to the best-effort approach of the Internet. Thus, during the transmission of targeted advertising, data congestion and unpredictable delays may occur. Those delays cause synchronization faults in the content presentation. Aiming at reducing these faults, it is necessary to use a mechanism to prepare the targeted content in advance, before it shall be presented to the viewer.

In a previous work [11], we proposed a new multimedia event type named *preparation*, to be used as part of the application logic, in the authoring phase. The multimedia author may use a preparation event to specify when a media content should start to be loaded and then monitor if the preparation succeeds. The media preparation proposed in [11] finalizes when the specific media player instantiation and the media content preloading are completed. The content preloading considers the buffer space available at the DTV receiver, which may be a determination of each DTV standard.[1]

Content preload is also proposed in the literature [12], [13] through prefetching mechanisms. Different from the preparation mechanism, content prefetching only finalizes when all media content is downloaded in the receiver device. Besides that, the prefetching mechanism does not consider the player instantiation. Therefore, prefetching demands a high storage capacity, that is a limited resource in some devices as DTV receivers.

The media preparation proposed in [11] only occurs when the application author specifies such event manipulation in the multimedia document. If the multimedia author does not specify this preparation for a specific media, the media content loading will only occur when the media presentation event starts. In this case, the media presentation will certainly suffer a delay. A disadvantage of the preparation being specified in the authoring phase is that the author needs to know or infer the time needed to prepare a particular media content and handle the preparation event explicitly. Anyway, thinking of viewer's quality of experience, the author may also specify that a targeted content shall not be presented if its preparation fails.

As an alternative to preparation specification during the application authoring phase, this paper proposes the automatic preparation of media objects at runtime. The automatic preparation is triggered by the multimedia formatter instead of the application itself. The support for automatic preparation reduces or avoids delays during broadband content transmission. To accomplish this goal, our work proposes a preparation plan (coexisting with the presentation plan) that orchestrates the content buffering and media player instantiation in advance. The preparation plan is a data structure dynamically built based on the presentation moments of media objects that compose a DTV application, besides the current networking conditions.

During the multimedia document parsing, the Ginga-NCL formatter builds a presentation plan that describes the application's temporal behavior. Different from a typical Ginga-NCL implementation, in this work, the presentation plan derives from the Hypermedia Temporal Graph (HTG) proposed by Costa et al. [14]. The presentation plan, together with information on network transmission conditions, allows the multimedia formatter to predict each media content's preparation time and build the proposed preparation plan.

It is important to notice that the automatic preparation proposed in this work does not interfere with the broadcast content playback. Furthermore, the multimedia application author can define that broadband content presentation will only replace the broadcast content if it is prepared to be presented. Therefore, in case the targeted content is not prepared at the time of the commercial break to be replaced, the viewer will watch the broadcasted advertisement.

Targeted advertising is an interesting approach from both the viewers' and the broadcaster's viewpoints. TV viewers can consume the most relevant ads because their profile is considered when the advertisement is requested, and TV broadcasters can reach their target audience more objectively. Moreover, the use of broadband channel allows DTV applications to notify broadcasters if the viewer watched the targeted advertising successfully. This feature allows broadcasters to estimate the targeted content range, assisting in the advertising billing method and audience profiling.

The main contribution of this work is the automatic preparation of media objects to support targeted advertising. Aiming at validating our proposal, the mechanism to automatically prepare media content was implemented in the Ginga-NCL middleware and performance tests were done to evaluate switching delay between broadcast and broadband streams. Our contributions are summarized as follows:

- We propose a preparation plan to support the automatic preparation of broadband content. This preparation plan is based on the temporal behavior of the multimedia application and the network conditions (e.g., transmission delay).
- We incorporate the automatic preparation in Ginga-NCL middleware, which is the presentation machine of the Brazilian Digital TV System.

[1] For example, SBTVD recommends a preparation buffer large enough to hold 30 seconds of Full HD H.264 video and AAC audio [1].

- We have conducted experiments to evaluate the performance of the automatic preparation mechanism for supporting targeted advertising using the Ginga-NCL middleware.

The remainder of this paper is structured as follows. Section II presents related work regarding targeted advertising transmission in IBB systems. Section III describes the preparation event employed to ensure the delivery of targeted content. Section IV presents the automatic preparation mechanism proposed in this paper. Section V presents a general architecture to support the targeted advertising service. As proof of concept, Section VI describes the implementation of automatic preparation in the Ginga-NCL middleware. Section VII presents the proposal evaluation considering different DTV processing capacity and broadband channel configuration scenarios. Section VIII discusses important aspects of our proposal, highlighting strong points and current limitations. Finally, Section IX concludes this paper and presents future work.

II. RELATED WORK

Integrating broadband communication into traditional television systems enables offering services as interactivity and content personalization. Aiming at supporting the delivery and synchronized consumption of related hybrid (broadcast/broadband) media contents, Boronat et al. [15] propose an end-to-end platform compatible with the Hybrid Broadcast Broadband TV (HbbTV) standard (version 2.0.1) [2]. The platform proposed in [15] allows the delivery of media content using broadcast technologies, such as Digital Video Broadcasting (DVB), and via broadband, using Dynamic adaptive streaming over HTTP (DASH) [16].

Boronat et al. [15] propose synchronizing the broadcast and broadband content using a Controller/Agent scheme. This synchronization scheme uses a shared clock server to insert timestamps in all media streams and consists of three steps: (i) retrieving the Timed External Media Information (TEMI) timelines from the incoming MPEG2-TS, which give information about the generation instants of the video frames; (ii) tracking the video frames until the rendering elements (i.e., the audio/video sinks); and (iii) registering the estimated presentation absolute timestamps of those video frames.

After extracting the generation timestamp (TEMI timeline) of the currently processed content and its "estimated" presentation timestamp, the server sends this information to the application executing in the hybrid terminal. According to Boronat et al. [15], the inclusion of the "estimated" presentation timestamps allows achieving higher sync accuracy than reception time. This is because it allows overcoming variability issues regarding both network and end-system delays and jitter. The application server (Controller) should frequently report the timestamp information to the receivers (Agent) consuming their content. Upon receiving these reports, each receiver will be able to compare its playout timing with the server playout timing. Therefore, the receiver can calculate the playout time difference between them and perform the required playout adjustments, if needed. Unlike Boronat et al.'s proposal, which aims at ensuring synchronization between broadcast and broadband streams by inserting timestamps, our work solves this problem through broadband content preparation, following an event-based approach.

Yim et al. [17] propose some changes to be applied in the current ATSC 3.0 UHD system to support targeted advertisements. The ATSC 3.0 runtime environment supports personalized client-side advertisement insertion based on DASH technology. In Yim et al. [17] proposal, the switching from broadcast content to targeted content is done through a broadcast application. These applications signal in advance to the receiver the advertising section that should be replaced. And the receiver determines whether there is a personalized advertisement to be shown to the viewer within that section based on the viewing context.

Unlike our proposal, which downloads only part of the advertising content in advance, in Yim et al.'s [17] proposal, all targeted advertising content is predownloaded to the receiver, and in some cases, more than one advertisement may be stored. So, our proposal to deliver personalized ads uses less memory space in the receiver. As the content is already in the receiver, the proposal by Yim et al. [17] tries to ensure broadcast/broadband synchronization using timestamps. Initially, the transmission system and the receiving system are synchronized with the L1-D time [18]. Secondly, time information related to the media presentation is specified using Coordinated Universal Time (UTC) for absolute time-based media synchronization. Besides that, to minimize the time associated with different file formats in the content switching process, the proposed system uses a targeted advertising content format that is set to be the same as the live stream.

In DTV environments, multimedia applications may execute in different devices with specific constraints and features. In this context, Abdelli et al. [19] propose an algorithm to schedule the insertion of prefetching requests in multimedia documents considering the user context, as bandwidth and available memory. The prefetching mechanism is used to anticipate the content requests in SMIL (Synchronized Multimedia Integration Language) [20] applications, enabling the media player to retrieve content in time for its presentation.

According to Abdelli et al. [19], the prefetching mechanism may be static or dynamic. The static prefetching saves in the receiver's memory all media content that will be presented. In the dynamic prefetching, only media guaranteed to be presented are thus prefetched and saved into memory. In Abdelli et al. [19], a proxy server processes the multimedia document and collects the user profile and context information (e.g., resolution of the screen, buffering capacity of the player, and available bandwidth). Besides that, that proxy server builds a temporal graph of the multimedia application to decide which media objects should be scheduled to be prefetched, which ones should be planned to be real-time delivered, and which ones should be replaced. For each case, the algorithm proposed in [19] computes a set of parameters that allows for scheduling the start and the duration of each prefetching operation. Moreover, the algorithm assesses the bandwidth and memory occupation required for each media object. Finally, the algorithm assesses the importance of each object relative to the presentation by computing its weight. The object weight is used to advantage the media object whose execution is crucial for consistency of the presentation. If the content cannot be prefetched and cannot be delivered in real-time, it is replaced for other content to ensure presentation consistency.

A limitation of Abdelli et al.'s [19] proposal is that it applies only to the SMIL language. Besides, the algorithm proposed in [19] demands a high processing capacity and requires a proxy server to schedule media prefetching. In contrast, the preparation mechanism proposed in our work can be executed on DTV receivers and can be adapted to different authoring languages and multimedia formatters.

Content prefetching is also described in the HbbTV standard [21] through the *preload* attribute of the HTML5 media element. In the HbbTV standard [21], when the *preload* attribute is set to ''none'', the terminal should not download audio/video content for that media element. When rendering a media format that uses manifest or playlist files (such as DASH [16] and HTTP Live Streaming [22]) the terminal may continue to download these files in case the *preload* attribute is set to ''none''. When the *preload* attribute is set to ''metadata'' the terminal should download audio/video content for this media element at a reduced rate, for example by downloading slower than real-time. And when the *preload* attribute is set to ''auto'' the terminal may choose any downloading strategy, including using as much bandwidth as available.

As the media content prefetching in the HbbTV standard [21] is defined by the *preload* attribute of the HTML5 language, the application author should specify which videos he/she thinks should be loaded in advance. Our solution, on the other hand, prepares the broadband content automatically, without requiring the author to specify this information in the application authoring phase.

Table 1 summarizes the characteristics of related studies that aim to ensure the synchronization between broadcast and broadband content in targeted advertising application. Analyzing them, it was possible to observe that not much research has been done to provide targeted advertisement in the Brazilian Digital Terrestrial TV System (SBTVD). In addition, those proposals do not consider the storage capacity of the receiver device, as is done in our current proposal.

III. THE PREPARATION EVENT

In a broadcast service, the broadcaster transmits a single content for all viewers simultaneously using a dedicated channel. In IBB systems, a broadcaster may send personalized content using a secondary communication channel, named broadband channel. However, in broadband communication, the transmission channel is a shared resource, and it is susceptible to network congestion problems. Congestion can cause delays in personalized content delivery and degradation of users' quality of experience (QoE).

In order to reduce the switching time between broadcast and broadband content and avoid delay to start the transmission of targeted content via broadband, this work proposes the use of a mechanism named content preparation. This mechanism was initially proposed in [11] to enable multimedia authors to control the preparation process of media content as a part of the application logic in the authoring phase. In [11], the content preparation is modeled as a new event type to be used in multimedia applications whose synchronization among media objects is based on events.

In a multimedia application, an event is an occurrence in time that can be instantaneous or have a measurable duration [23]. Besides that, an event may be predictable (like the end of a media content presentation with known duration and known beginning time) or unpredictable (like viewer interactions) [14]. Thus, the preparation event proposed in [11] is an event triggered by the application when a media object must have its content prepared before it should be presented. According to [11], the occurrence of a preparation event finalizes when (i) all content requested is loaded into the media player buffer, or (ii) the media player data buffer is full, considering the situation that occurs first. In addition to loading data, the preparation event also includes the media player instantiation task as part of the preparation process.

In our previous work [11], the advance content preparation only occurs if the application author explicitly defines this operation in the TV application. In case the author does not specify such preparation, the media content preloading occurs when the media presentation starts. In this scenario, the content will be presented with an inevitable delay. A disadvantage of our previous approach [24] is that the author must know or infer the time necessary to prepare a particular media object content, and always specify when it should be done explicitly in the TV application.

This paper proposes to automate the media content preparation to reduce or avoid delays during the execution of multimedia applications. As aforementioned, in IBB systems, it is possible to send personalized content via broadband connection synchronized with the broadcast content. Thus, this work also proposes applying content preparation to deliver targeted advertising in IBB systems, avoiding synchronization faults among broadband and broadcast content. For this purpose, we propose the creation of a preparation plan based on the exhibition moment of each television commercial. The broadcaster should provide an application (multimedia document) specifying the synchronization between the personalized and broadcasted content. This multimedia document is analyzed by the IBB middleware at the receiver side to obtain the presentation time of each targeted ad and automatically build a preparation plan to define media object preparation time instants.

The preparation duration for a broadband-delivered content depends on networking conditions, and in some cases, it may happen that the preparation cannot be completed on

TABLE 1. Comparison of related work.

Work	Approach to ensure IBB synchronization	DTV Standard	Targeted content download	Automatic process
Boronat et al. [15]	Use timestamps to synchronize the broadcast and broadband content using a Controller/Agent scheme	HbbTV	N/A	Yes
Yim et al. [17]	Use an application to signal in advance the advertising section that should be replaced. Uses L1-D time to synchronize transmission and reception systems, and UTC for media synchronization.	ATSC 3.0	All targeted advertising content is downloaded in advance, and in some cases, more than one advertisement may be stored.	Yes
Abdelli et al. [19]	Use prefetching mechanism to anticipate the content requests in SMIL applications, enabling the media player to retrieve content in time for its presentation.	N/A	Static prefetching saves in the receiver all media content that will be presented. Dynamic prefetching save only media guaranteed to be presented.	Yes
HbbTV Standard [21]	Use the *preload* attribute of the HTML5 language to content prefetching. In this proposal, the application author should specify which videos he thinks should be loaded in advance.	HbbTV	Download all content of the audio/video in the receiver.	No
Our proposal	*Automatic preparation of broadband content considering the temporal behavior of application and the network conditions.*	*SBTVD*	*The content pre-loading considers the buffer space available at the DTV* receiver	*Yes*

time for a smooth presentation. For this sake, this paper also proposes an attribute associated with the preparation event to signal whether a preparation was successful or not. This attribute is named `prepared`, and its initial value is "false". When a content preparation is completed successfully by the IBB middleware, the `prepared` attribute value shall be modified to "true". Content preparation uses some resources of the receiver device such as the storage space, and in some cases, it is necessary to release this resource for another application. Thus, if the `prepared` attribute is "true" and the resources used in the preparation are released before media content presentation, the `prepared` attribute is set to "false". Thus, the application author can use the `prepared` attribute to verify if the media object is prepared before starting its presentation.

IV. AUTOMATIC PREPARATION PROPOSAL

This section presents the primary components used to carry out the automatic preparation of media objects that compose an application. The automatic preparation is carried out by the multimedia formatter and allows to reduce, or even avoid delays during the execution of multimedia applications delivered over the network. The execution of automatic preparation uses information of the execution environment, such as available storage space and transmission rate on the communication channel. Therefore, all the steps described below are performed at the receiver and do not require any processing on the broadcaster side.

A. MODELING TEMPORAL BEHAVIOR

In the authoring phase of a multimedia application, an author can specify a document that describes the media content presentation behavior. The multimedia document can be specified through declarative languages, such as NCL [6], SMIL [20] HTML [9], or using imperative languages, like JavaScript [25] and Lua [26], for example.

Furthermore, as previously mentioned, the synchronization relationships among content that compose the multimedia application can be specified using an event-based or timeline paradigm. In the timeline paradigm, the presentation moment of each content is defined by positioning media objects over the time axis. On the other hand, in multimedia languages that use event-based synchronization, the occurrence moments of the events are not explicit in the multimedia document. Thus, some studies published in the literature [14], [27], [28] propose to use a data structure to model the temporal behavior of a multimedia application.

In this paper, the temporal behavior of a multimedia application is modeled using a directed graph named Hypermedia Temporal Graph (HTG), which was proposed by Costa et al. [14]. It consists of vertices and directed edges, where vertices represent event transitions, and a directed edge represents a relationship between two events in a multimedia application. Besides, the hypermedia graph allows for representing specific conditions and priorities between edges that have the same source vertex.

In summary, the hypermedia temporal graph is defined by a tuple (V, A, C, N), where:

- V $= (v_0, v_1, v_2, \ldots, v_{n-1}, v_n)$ is a finite set of vertices;
- A $= (a_0, a_1, a_2, \ldots, a_{n-1}, a_n)$ is a finite set of edges, where each edge connects two different vertices v_i and v_j;
- C $= \{c_{ij}\}$ is a finite set of traverse conditions associated with the edges. Condition c_{ij} is associated with edge $(v_i, v_j) \in$ A. A condition must be satisfied in order to trigger the action specified in the edge's output vertex.
- N $= \{n_{ij}\}$ is a finite set of priorities associated with the edges. Priority n_{ij} is associated with each edge (v_i, v_j) $\in$ A and represents the order in which this edge must have its c_{ij} condition verified, compared to other edges starting from the same source vertex.

An example of temporal graph composed of four vertices and three edges is presented in Figure 1. This graph represents

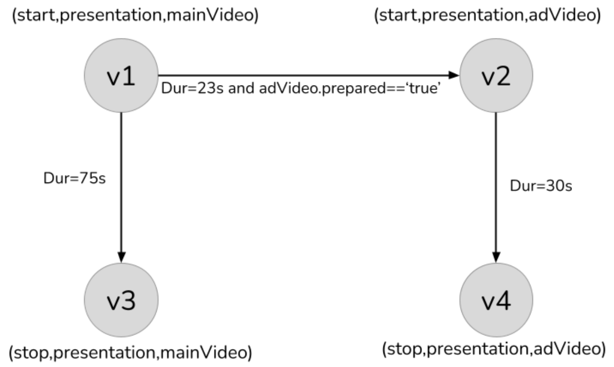
(start,presentation,mainVideo)
(start,presentation,adVideo)
v1
Dur=23s and adVideo.prepared=='true'
v2
Dur=75s
Dur=30s
v3
v4
(stop,presentation,mainVideo)
(stop,presentation,adVideo)

FIGURE 1. Temporal graph representing a targeted advertising application.

a targeted advertising application, starting with the presentation of a main video represented by the vertex v_1 in the temporal graph. The duration of main video presentation is equals to 75 seconds. This characteristic is represented by the edge connecting vertices v_1 and v_3, and has as condition the main video duration. The v_3 node represents the end of the main video presentation. After 23 seconds of the main video presentation, the targeted advertising (adVideo) presentation (vertex v_2) starts, and stops after 30 seconds (edge connecting v_2 and v_4).

The temporal graph is divided into other subgraphs – named temporal chains – composed of vertices representing predictable event transition sequences. Each temporal chain must be started by an unpredictable event. According to Costa et al. [14], dividing the HTG into temporal chains facilitates the calculation of the start times of events and the presentation control in multimedia applications.

When the HTG contains edges labeled by an unpredictable condition (e.g. user interaction, environment or user variable value changes), several auxiliary temporal chains must be created. However, if the graph contains only edges that have a temporal duration as an associated condition, it defines a single temporal chain.

The hypermedia temporal graph can be extended with other data structures, to assist the control of multimedia application presentation, as well as its transmission. On the receiver side, the HTG can be used to represent a presentation plan [14] that orchestrates media content presentations that make up a multimedia application. This structure can also be used to estimate the instantiation moment of media players. On the server side, the HTG may be specialized into a transmission plan [14], [29], which specifies when each media content must be transmitted.

Aiming at supporting the automatic preparation of media objects, this paper proposes to use the hypermedia temporal graph as the basis to build a preparation plan at the TV receiver side. The preparation plan provides the specific time instants that each media content must be requested from a remote server when the multimedia application is transmitted over a communication network. Besides the preparation plan, this paper also uses information from the presentation plan to control the automatic preparation of media objects.

TABLE 2. Presentation plan of the targeted advertising application.

Instant	Transition	Event	Object
0s	start	*presentation*	mainVideo
23s	start	*presentation*	adVideo
53s	stop	*presentation*	adVideo
75s	stop	*presentation*	mainVideo

B. PRESENTATION PLAN

In the authoring phase, a multimedia author specifies media object temporal behavior, their spatial position and the synchronization relationships among them. In order to maintain those relations during the execution phase, the multimedia formatter can use the presentation plan for supporting presentation control.

The presentation plan is a data structure derived from the hypermedia temporal graph. This plan contains the occurrence moments of each predictable presentation event specified in a multimedia application. Therefore, the presentation plan supplies the moments that the presentation of a media object must start and end in a multimedia application. Table 2 shows the presentation plan built based on the temporal graph specified in Figure 1.

According to Costa et al. [14], the presentation plan enables the multimedia formatter to position a multimedia presentation at any point in time, advancing, rewinding or resuming an application. Initially, the presentation plan comprises only the specifications associated with the main temporal chain of the hypermedia temporal graph. If the multimedia application contains unpredictable events, the presentation plan must be updated during the execution of the application. This should happen because, if an unpredictable event occurs, all temporal moments, where other events resulting from such unpredictable event should occur, must be updated.

C. PREPARATION PLAN

As aforementioned, the delivery of personalized content in IBB systems is usually performed through a broadband connection. The broadband connection allows for the transmission of content targeting at different users. However, broadband network transmission delays may be experienced. Taking into account this characteristic of broadband transmission, this paper proposes to automatically prepare the targeted content of multimedia applications offered in IBB systems. To support automatic preparation, this work proposes to build a data structure named *preparation plan*.

The preparation plan's creation considers the available space in the media player buffer and the network transmission conditions. Using the preparation plan, the multimedia formatter may request the content of media objects in advance. Thus, it is possible to reduce or avoid synchronization faults during the multimedia application presentation.

The API (Application Programming Interface) between the multimedia formatter and a media player allows them to exchange information. For example, a media player can notify the media presentation state (e.g., paused, stopped,

TABLE 3. Preparation plan of the targeted advertising application.

Instant	Transition	Event	Object
0s	start	*preparation*	mainVideo
16s	start	*preparation*	adVideo

or running) and the available buffer space. The multimedia formatter also communicates with network components to obtain data transmission statistics, such as throughput, jitter, and transmission delay.

The content preparation loads media information units into the media player buffer in advance at the receiver side. The preparation duration of a media content ($dur_{preparation}$ parameter) depends on the buffer size and the media player instantiation time ($t_{instantiation}$), and can be estimated as defined in Equation 1. The *throughput* parameter represents the average network transfer rate between the content server and receiver. Moreover, the *buffer_size* parameter represents the available buffer space.

$$dur_{preparation} = t_{instantiation} + \frac{buffer\_size}{throughput} \quad (1)$$

In summary, the multimedia formatter proposed in this work can obtain the moment of content presentation, the media player buffer size, and the estimated time to prepare it. Besides that, the multimedia formatter can use that information to automatically create the preparation plan for the media objects that compose the application. The instant to start preparing a media object can be defined using Equation 2.

$$t_{preparation_i} = t_{presentation_i} - dur_{preparation_i} \quad (2)$$

For example, consider that the estimated time needed to prepare media *adVideo* in the application previously described in Table 2 is 7 seconds. In this case, it is possible to build a preparation plan, shown in Table 3, that derives from the presentation plan presented in Table 2. As the start time of *adVideo* is 23 seconds in Table 2 and it takes 7 seconds to prepare it, the preparation start time of *adVideo* is (23 - 7), that is, 16 seconds in Table 3. When the estimated duration of the media preparation ($dur_{preparation_i}$) is longer than the presentation moment of the same media ($t_{presentation_i}$), the preparation instant would be a negative value. In this case, the multimedia formatter assigns the zero value to the preparation time. The same occurs with the media that starts at the beginning of the application, which is the case of *mainVideo* in Table 3.

After building the preparation plan, the temporal graph is updated to represent the preparation event. Thus, one or more vertices may be inserted into the HTG. The new vertex specifies a start action in the media preparation event that must be prepared for presentation. Figure 2 presents the updated HTG with the preparation event (vertex v_5). Figure 2 also shows how edge conditions can be specified testing if the `adVideo prepared` attribute is "true" to avoid presenting a targeted content if it is not ready for presentation. Notice the edge condition between vertices v_1 and v_2.

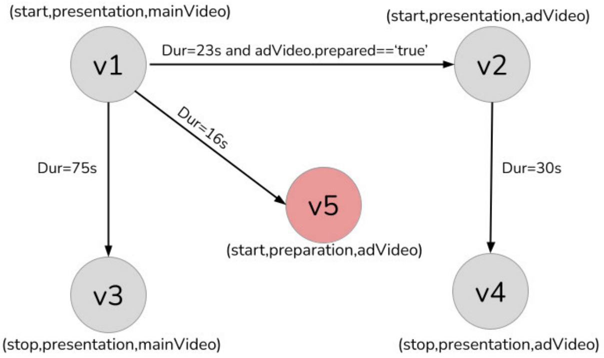
(start,presentation,mainVideo)
(start,presentation,adVideo)
v1
Dur=23s and adVideo.prepared=='true'
v2
Dur=16s
Dur=75s
Dur=30s
v5
(start,preparation,adVideo)
v3
v4
(stop,presentation,mainVideo)
(stop,presentation,adVideo)

FIGURE 2. Updated temporal graph with a preparation event.

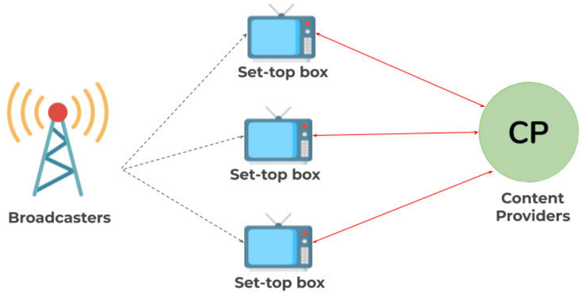

FIGURE 3. Architecture to support targeted advertising.

V. ARCHITECTURE TO SUPPORT TARGETED ADVERTISING IN IBB TV SYSTEMS

In digital terrestrial television systems, broadcasters deliver general content for all devices tuned into a specific channel. Therefore, different viewers with different preferences and features always receive the same content. To enable personalized content delivery, it is necessary to integrate broadband communication into the user receiver. This section presents the communication architecture to support the targeted advertising service in IBB television systems.

The architecture proposed in this paper is composed of the broadcaster, viewer digital receivers (set-top boxes or TVs), and content providers, as depicted in Figure 3. During the commercial break, the set-top box may request personalized content, considering the user profile, from content providers instead of presenting a general broadcasted advertisement. In this way, the creation of the preparation plan and the automatic preparation of the broadband content is carried out in the receiver. The content provider can be managed by the broadcaster that controls the personalized content delivered to each viewer or by a third party. The targeted content request is carried out through a broadband channel that connects the TV receiver to the content provider, represented by red lines in Figure 3. Thus, the content provider delivers an advertising content according to the user profile.

The user profile may be defined in different ways, as through registration on the television receiver or an analysis of the viewer behavior. In the first one, the viewer explicitly specifies his/her preferences, location, and features (i.e., age, gender, and schooling). In the second option, the digital receiver may gather usage data from the viewer and apply mechanisms to define the user profile implicitly [30]. Different studies in the literature [31], [32] propose mechanisms for choosing the most suitable content based on the user profile. As our proposal focuses on a guaranteed delivery of targeted advertising synchronized with the broadcast content, it is important to notice that it is independent of the user profile definition mechanism employed. Our proposal considers that this profile has already been defined previously at the receiver side.

Regarding the broadcaster viewpoint, the time slots for broadcasting advertisements represent a mean to increase the revenue. Therefore, it is mandatory that the targeted content, replacing the general advertisement, be presented at the correct time. Besides that, the duration of targeted advertising must follow the interval pre-established by the broadcaster. In order to satisfy those requirements, this work proposes incorporating a module for content preparation management in digital TV receivers, in order to reduce the switching time between broadcast and broadband.

The preparation management module is responsible for analyzing the metadata provided by the broadcaster to define when the targeted content should be presented. Based on this analysis, a preparation plan is built, as defined in Section IV-C. Moreover, the preparation module notifies the middleware if it is not possible to prepare the content. This situation may occur due to congestion problems in the broadband transmission channel. This approach allows the application to display targeted content only if it is prepared, i.e., its `prepared` attribute is "true". Otherwise, the viewer will watch the broadcasted advertisement.

VI. IMPLEMENTATION

In order to validate our proposal, this work extended the reference implementation of Ginga-NCL,[2] which is the standard middleware of the Brazilian Digital TV System. Besides that, we created a targeted advertising application using the NCL language.

A. NCL APPLICATION

NCL is an XML-based language and all media objects that compose the application are specified through `<media>` elements. Listing 1 presents part of the NCL code that specifies a targeted advertising application used to validate our proposal. The NCL language also enables to define temporal or spatial media anchors. A media anchor represents a subset of information units of a `<media>` element's content, which is specified by `<area>` elements (see line 33 in Listing 1).

The NCL language enables to represent causal relation among media objects, through the `<causalConnector>` element. In a causal relation, a condition must be satisfied in order to trigger an action. Besides that, a causal relation may contain simple (`<simpleCondition>` element) or compound (`<compoundCondition>` element) conditions. In compound conditions, the `operator` attribute specifies the logical expression among the child elements of the `<compoundCondition>`.

As aforementioned, the `prepared` attribute can be used to verify if a media content is prepared or not. Thus, one of the conditions of the `<causalConnector>` element, specified in Listing 1 (lines 5-16), is the prepared attribute verification of the preparation event (`<assessmentStatement>` element). The `<causalCon- nector>` element may be used for creating relationships (`<link>` elements) in NCL documents. An NCL link (lines 41-46 in Listing 1) specifies the relation between media objects `mainVideo` and `adVideo`, using the semantic specified by the causal connector. This link specifies that the `adVideo` should be started after 23 seconds of the `mainVideo`, only if `adVideo` was prepared properly. This relation can be represented in the temporal graph using two vertices (v_1 and v_2) and one edge ($v_1 v_2$), as depicted in Figure 2.

During the targeted advertising exhibition, the broadband content must overlap the broadcast. As it is not possible to "pause" the content transmitted by broadcast, the broadcast content has its audio volume set to zero when starting the broadband content presentation (lines 52-58 in Listing 1). Besides, for one content to overlap the other, in NCL, it is necessary to set the "zIndex" property of the media that will overlap with a value higher than that of the overlapping media (lines 35 and 39 in Listing 1).

Aiming at ensuring that the targeted advertising ends at the right moment, i.e., when the main broadcast program starts again, we created a second synchronization relation, as specified in Listing 1 (lines 17-20). The NCL link (lines 47-51 in Listing 1) uses the semantic specified by the previously mentioned causal connector, which stops the presentation of the targeted advertising content at the end of the commercial period. In the example application, we consider the commercial with a duration of 30 seconds.

In DTV transmissions, some broadcast contents are live (i.e., sports or news) and the moment at which the targeted advertising should be presented is dynamic. In this case, the NCL language supports live edition commands [33] that modify the multimedia document during runtime. In case the multimedia document is modified, the presentation and preparation plans will be updated to reflect such changes.

B. GINGA-NCL MIDDLEWARE

The internal Ginga-NCL API is composed of three main components, *Formatter*, *Parser*, and a representation of the multimedia document named *Document* component. The *Formatter* controls the entire life cycle of the NCL application, which begins with the analysis of the multimedia document, carried out by the *Parser*.

To enable building the hypermedia temporal graph, the *Parser* component was modified. In this step, the temporal graph is built based on the media nodes that compose the

[2] https://github.com/TeleMidia/ginga

```
<ncl>
<head>
...
<connectorBase>
 <causalConnector id="onBeginAndTestStart">
  <compoundCondition operator="and">
    <simpleCondition role="onBegin" />
    <assessmentStatement comparator="eq">
     <attributeAssessment role="test"
      eventType="preparation"
      attributeType="prepared"/>
     <valueAssessment value="true" />
    </assessmentStatement>
  </compoundCondition>
  <simpleAction role="start" />
 </causalConnector>
 <causalConnector id="onEndStop">
  <simpleCondition role="onEnd" />
  <simpleAction role="stop" />
 </causalConnector>
 <causalConnector id="onBeginSet">
  <connectorParam name="val"/>
  <simpleCondition role="onBegin" />
  <simpleAction role="set" value="$var"/>
 </causalConnector>
</connectorBase>
...
</head>
<body>
...
 <media id="mainVideo" src="stream.mp4"
  descriptor="dVideo">
  <area id="a1" begin="23s" end="53s"/>
  <property name="soundLevel" value="1"/>
  <property name="zIndex" value="0"/>
 </media>
 <media id="adVideo" src="http://..."
  descriptor="dAdv">
  <property name="zIndex" value="1"/>
 </media>
 <link xconnector="onBeginAndTestStart">
  <bind role="onBegin"
   component= "mainVideo" interface="a1"/>
  <bind role="test" component="adVideo"/>
  <bind role="start" component="adVideo"/>
 </link>
 <link xconnector="onEndStop">
  <bind role="onEnd"
   component= "mainVideo" interface="a1"/>
  <bind role="stop" component="adVideo"/>
 </link>
 <link xconnector="onBeginSet">
  <bind role="onBegin" component= "adVideo"/>
  <bind role="set" component="mainVideo"
   interface="soundLevel">
   <bindParam name="val" value="0"/>
  </bind>
 </link>
 <link xconnector="onEndSet">
  <bind role="onEnd" component= "adVideo"/>
  <bind role="set" component="mainVideo"
   interface="soundLevel">
   <bindParam name="val" value="1"/>
  </bind>
 </link>
 ...
</body>
</ncl>
```

Listing 1. Targeted advertising application specified in NCL language.

document, and the edges are created based on the relationships defined by NCL links. The *Parser* also creates edges representing the end transition (end time) of a media object presentation. These edges have as condition the duration of the media object. In the NCL language, the media duration can be defined explicitly using parameter `explicitDur`. Moreover, it is possible to obtain its implicit duration if it is a continuous media by analyzing its metadata.

In addition to the modifications to the *Parser* component, this work incorporates three new classes to the Ginga-NCL reference implementation: *TemporalGraph*, *PresentationOrchestrator* and *PreparationOrchestrator*. The *TemporalGraph* class represents the hypermedia temporal graph and contains all the graph manipulation functions, such as the graph search method to obtain the temporal chains. This component is obtained from the application specification.

The *PresentationOrchestrator* class is responsible for building and managing the presentation plan. This class has a structure called *PresentationPlan* that contains the occurrence moments of transitions (start and end times) of the presentation event of each media object that compose the application. The function to create the presentation plan receives a *TemporalGraph* object as a parameter.

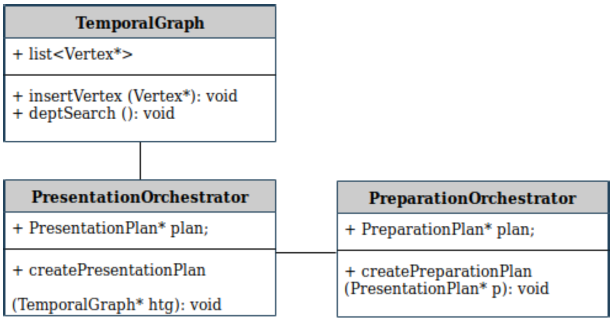

FIGURE 4. New classes added to the reference implementation of Ginga-NCL to support automatic preparation.

Finally, the *Preparation Orchestrator* class implements the methods to build the preparation plan. First of all, the network conditions are analyzed to obtain the time necessary to prepare an object based on the media player buffer size. In this step, the *PreparationOrchestrator* uses equations (1) and (2), presented in Section IV-C, and the presentation plan to create the preparation plan. The resulting preparation plan is a structure containing each media object's preparation times that compose the application. Figure 4 illustrates a class diagram representing the relations between the new proposed components.

VII. EVALUATION

This section presents a performance evaluation to demonstrate the efficiency of the automatic preparation mechanism for supporting targeted advertising using the Ginga-NCL middleware. The evaluation tests were performed with two different TV receiver configurations. The first one (A1) is a Dell computer Intel Core i7-8550U CPU 1.80GHz, 1 TB of hard disk, and 16 GB of RAM. The second one (A2) is an LXC container running over a Raspberry Pi 4 Model B, 4 GB of RAM, and quad-core Cortex-A72.

DTV multimedia applications, like targeted advertising applications, run in set-top boxes or TVs with an integrated receiver. Some TVs available on the market have a processor with ARM (Acorn RISC Machine) architecture, 32 or 64-bit, CPU dual or quad-core, and 2GB of RAM [34], [35], [36]. In those TVs, only a part of the CPU is available to applications since they also need to process the content received by the broadcast channel. Therefore, we used a container with 1 GB of RAM and one CPU core in our tests.

In the first testing phase, we executed the targeted advertising application on a version of Ginga-NCL without support for automatic preparation, which is currently used in the Brazilian Digital Terrestrial TV receivers. This version either does not support using the "prepared" attribute to verify if the content is prepared. In the second phase, the application

TABLE 4. Test scenarios.

Scenario	Rate	Automatic Preparation	TV receiver
Scenario 1	8 Mbps	Applied	A1
Scenario 2	15 Mbps	Applied	A1
Scenario 3	25 Mbps	Applied	A1
Scenario 4	8 Mbps	Not applied	A1
Scenario 5	15 Mbps	Not applied	A1
Scenario 6	25 Mbps	Not applied	A1
Scenario 7	8 Mbps	Applied	A2
Scenario 8	15 Mbps	Applied	A2
Scenario 9	25 Mbps	Applied	A2
Scenario 10	8 Mbps	Not applied	A2
Scenario 11	15 Mbps	Not applied	A2
Scenario 12	25 Mbps	Not applied	A2

was executed in our extended implementation, which uses the preparation plan. Each test phase consisted of 50 executions of the targeted advertising application to obtain the average switching time between the broadcast and broadband content.

The switching time is the period between the moment at which the broadcast video ceases presentation and the beginning of the broadband video or audio presentation, as defined in [2], and is directly related to the delay in presenting targeted content. In this work, during the broadband content presentation, the set-top box continues receiving the broadcast content. However, this content is replaced by the ad video delivered on the broadband channel. Besides, the decoding of broadcast and broadband video content is hardware-accelerated. Finally, we performed a third test to evaluate the performance of multiple preparations occurring in parallel in the same personalized content application.

According to a survey by the Brazilian National Telecommunications Agency [37] about multimedia communication services in Brazil, the average broadband transmission speed in the country is 24.62 Mbps. Besides that, in the country state with the lowest transmission rate, this value is approximately 8 Mbps. Therefore, we used those transmission rates and an intermediate value of 15 Mbps as parameters for our tests. The scenarios considered in the tests are described in Table 4.

After executing the tests, it was possible to observe that, considering TV receiver A1 (Dell computer), the average switching time was 0.02 seconds for *Scenario 1* where the transmission rate is 8 Mbps with automatic preparation. Meanwhile, in *Scenario 4*, with the same transmission rate and TV receiver configuration, but without automatic preparation, this time reached 9.9 seconds. Tests in the *Scenario 2* with a transmission rate of 15 Mbps demonstrate that the automatic preparation reduces in 4.5 seconds the switching time compared to the *Scenario 5* without automatic preparation.

Increasing the transmission rate to 25 Mbps, the switching time without preparation was equal to 2.46 seconds - 98 times longer than the time in cases with automatic preparation, at the same rate. Figure 5 presents the comparison of average switching time between the broadcast content and targeted advertising for each transmission rate with a 95% confidence interval. As the values without and with preparation have very different scales we use a logarithmic scale on the y-axis to represent the average switching time in milliseconds.

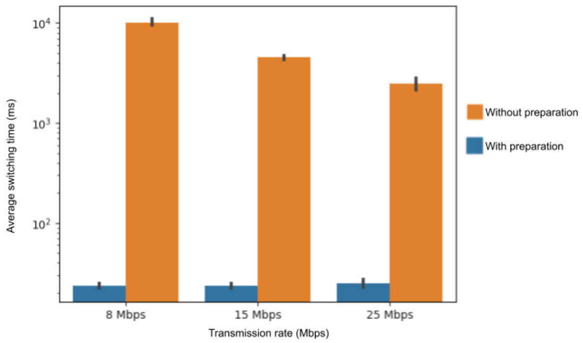

FIGURE 5. Average switching time with and without automatic preparation, considering different transmission rates and TV receiver A1. The y-axis values are represented in a logarithmic scale.

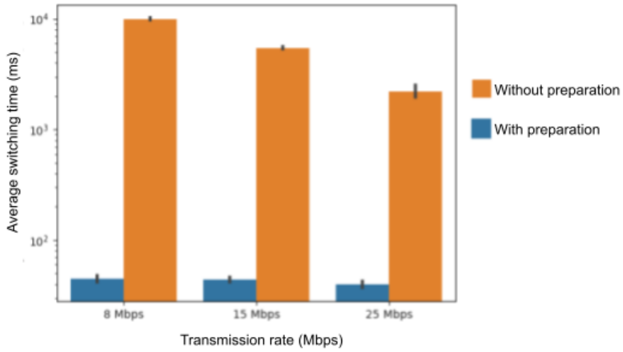

FIGURE 6. Average switching time with and without automatic preparation, considering different transmission rates and TV receiver A2. The y-axis values are represented in a logarithmic scale.

The tests considering TV receiver A2 (Ginga container running in a Raspberry Pi) demonstrate that in *Scenario 7*, automatic preparation reduced the average switching time by an average of 10 seconds compared to tests without automatic preparation (*Scenario 10*). When the transmission rate was 15 Mbps, the average switching time was equal to 5.47 seconds for the cases without automatic preparation (*Scenario 11*) and 0.04 seconds for the tests with the solution proposed in this work. Finally, in *Scenario 12*, where the transmission rate was 25 Mbps, the average switching time was 2.21 seconds, and 0.039 seconds in *Scenario 9* that employs the preparation plan proposed in this work. Figure 6 demonstrates the average switching time for each transmission rate. We choose to represent the y-axis in Figure 6 using a logarithmic scale for the same reason described previously.

Considering the obtained results, we conclude that the preparation mechanism proposed in this work can reduce the switching time between broadcast and broadband content and maintain synchronization relationships during the multimedia application execution. In addition, we verified that the performance of the automatic preparation mechanism proved to be efficient both in devices with greater computational

TABLE 5. Average preparation duration considering 10-50 simultaneous preparation and 20Mbps transmission rate for TV receiver A1.

Preparations	Duration	Standard Deviation
10	2,6 s	0,27
20	4,8 s	0,62
30	9,4 s	2,2
50	16,5 s	2,14

TABLE 6. Average preparation duration considering 10-30 simultaneous preparation and 20Mbps transmission rate for TV receiver A2.

Preparations	Duration	Standard Deviation
10	7,1 s	1,27
20	10,4 s	4,6
30	23,6 s	1,2

capacity and in more limited devices, such as those used in the DTV environments.

The NCL language supports specifying multimedia applications where the user can choose content to be presented using interaction events. Moreover, the language supports content adaptation through `<switch>` elements. This element enables creating, for example, an application that recommends a set of content based on what the user is watching. For these types of applications, where the content to be presented is defined at runtime, the formatter will prepare each content related to each option that the user can choose. Thus, when the user selects one of the recommended content, it would take less time to start presenting. Thus, a third testing phase was carried out to verify the impact of multiple content preparations simultaneously.

For TV receiver configuration A1, we initially considered the preparation of 10 movies simultaneously, 20 movies, 30 movies, and finally 50 simultaneous preparations and a transmission rate of 20 Mbps, as presented in Table 5. However, for TV receiver configuration A2, with the LXC container running on Raspberry, it was possible to perform a maximum of 30 simultaneous preparations. This is because, when increasing the number of videos, the receiver device presented low performance. The average preparation times for each scenario with TV receiver A2, with transmission rate of 20 Mbps, can be seen in Table 6. All tests were run 50 times. Thus, if a receiver has to prepare several content simultaneously, this additional delay should be considered to build the preparation plan.

VIII. DISCUSSION

In DTV environments, a broadcaster may send its content to different broadcast relay stations, which can be in different geographic locations. In these scenarios, the absolute time when a broadcasted program starts may be different at each location. In scenarios where the synchronization between the broadcast and broadband streams are defined through absolute times (timeline), if the broadcaster sends an application indicating the display of a targeted advertisement at time t_i of the broadcast content presentation, there may be synchronization faults at the relay stations.

In our proposal, the multimedia formatter builds a temporal graph of the application on the TV receiver side. This graph represents the synchronization relationships between the broadcast stream and the broadband stream based on events, not on the video's absolute time. For example, the broadcaster may specify that broadband content should be presented t minutes after the start of the presentation of certain broadcast content. Thus, if the broadcast content is presented at different times in each location, the synchronization between the broadcast and the broadband content will be respected.

The automatic preparation mechanism proposed in this work considers the network's current conditions responsible for transmitting broadband content and the receiver device's storage capacity. In our tests, we considered only one user at a time making requests to the targeted content server. Thus, the time to access the server was not considered in our calculation of the preparation time.

In addition, the transmission conditions of the broadband channel, such as transmission delay and available bandwidth, may vary while the multimedia application is executing. Therefore, the preparation plan created when the application was received will need to be updated periodically in response to possible fluctuations in the network.

It is important to highlight that, in cases where the preparation time obtained by the automatic preparation mechanism does not reflect the actual network delay, the broadband content will not be prepared (`prepared` attribute value will be "false"). In this case, the content will not be presented. This way, there will be no failure of synchronization between the broadcast content and the broadband. Considering the access time to the targeted advertising server in calculating the preparation time is left as future work.

The automatic preparation presented in this work aims at ensuring that the targeted content transmitted via broadband is presented at the correct time, as defined by the broadcaster. In general, broadcasters offer a specific space of time for advertisers to display their ads. Thus, by ensuring that broadband content will start at the right time, if the advertiser follows the broadcaster time, that content will also end at the right time. However, suppose the broadcaster wishes to guarantee the end of the targeted content's presentation at the correct time. In that case, it is possible to specify a synchronization relationship that ends the targeted content after t seconds (where t represents the duration of the targeted advertisement). This way, advertising will always end when the main broadcast program starts again.

IX. CONCLUSION

In integrated broadcast-broadband systems, advertisers can target their ads considering the user profile. In general, the transmission of targeted advertising uses a broadband connection. Besides that, targeted advertising should be synchronized with the broadcast content. Multimedia synchronization maintenance is mandatory for preserving the quality of content presentation and user's quality of experience. Thus, it is necessary to use mechanisms to ensure that media objects will be presented at the correct time, as defined by the application's author.

This paper proposed the automatic preparation of media object content, based on building a preparation plan, which allows the preparation in advance of media content for multimedia applications, reducing synchronization failures. The preparation mechanism used in this work considers the storage space available in the receiving device and the communication channel's transmission rate. In order to validate our proposal, we extended the reference implementation of the Ginga-NCL middleware to support automatic preparation.

It is important to highlight that, although we used the NCL language to validate our proposal, the automatic preparation mechanism is independent of the authoring language. Automatic preparation can be used on multimedia documents specified in other multimedia languages, as long as the multimedia formatter can translate those languages into a hypermedia temporal graph, which is the basis for the construction of the preparation plan.

Moreover, we ran performance tests evaluating the switching time between broadcast and broadband transmission. Test results demonstrated that the automatic preparation mechanism significantly reduced broadband content presentation delays.

As future work, we intend to extend other authoring languages, such as HTML5, and presentation machines, to support the automatic preparation mechanism. The specification of user profile is an important issue in the targeted advertising service. Therefore, another future work is the creation of a method to automatically define user profiles.

REFERENCES

[1] *Digital Terrestrial Television Data Coding and Transmission Specification for Digital Broadcasting—Part 1: Data Coding Specification (In Portuguese)*, ABNT, São Paulo, Brazilfeb 2021.

[2] HbbTV. (2015). *HbbTV 2.0. 1 Specification*. HbbTV Association. [Online]. Available: https://www.hbbtv.org/pages/about_hbbtv/HbbTV_specification_2_0.pdf

[3] ATSC. (2020). *ATSC 3.0*. AbnTeX. [Online]. Available: https://www.atsc.org/

[4] *Integrated Broadcast-Broadband Systems—Broadcasting Service(Television)*, document ITU ITU-R BT.2075-3, 2020.

[5] L. F. G. Soares, R. F. Rodrigues, and M. F. Moreno, "Ginga-NCL: The declarative environment of the Brazilian digital TV system," *J. Brazilian Comput. Soc.*, vol. 12, no. 4, pp. 37–46, Mar. 2007.

[6] *Nested Context Language (NCL) and Ginga-NCL for IPTV Services*, document ITU, ITU-R H761, Apr. 2009.

[7] M. Moreno, D. Saade, G. Lemos, R. Kuleza, and A. Guedes, "A brief overview about D-profile of Ginga DTV receivers," in *Proc. Workshop IMX Latin Amer.*, 2020, pp. 289–296.

[8] *Digital Terrestrial Television—Data Coding and Transmission Specification for Digital Broadcasting—Part 2: Ginga-NCL for Fixed and Mobile Receivers XML Application Language for Application Coding (in Portuguese)*, ABNT, São Paulo, Brazil, Feb. 2021.

[9] W3C. (2014). *HTML5 A Vocabulary and Associated APIs for HTML and XHTML*. World-Wide Web Consortium Recommendation. [Online]. Available: https://www.w3.org/TR/html5/

[10] G. Blakowski and R. Steinmetz, "A media synchronization survey: Reference model, specification, and case studies," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 1, pp. 5–35, Jan. 1996.

[11] M. Josué, M. Moreno, and D. Muchaluat-Saade, "Mulsemedia preparation: A new event type for preparing media object presentation and sensory effect rendering," in *Proc. 10th ACM Multimedia Syst. Conf.*, Jun. 2019, pp. 110–120.

[12] B. Meixner and C. Einsiedler, "Download and cache management for HTML5 hypervideo players," in *Proc. 27th ACM Conf. Hypertext Social Media*, Jul. 2016, pp. 125–136.

[13] B. D. Davison, "Predicting web actions from HTML content," in *Proc. 13th ACM Conf. Hypertext Hypermedia (HYPERTEXT)*, 2002, pp. 159–168.

[14] R. M. D. R. Costa, M. F. Moreno, and L. F. G. Soares, "Intermedia synchronization management in DTV systems," in *Proc. 8th ACM Symp. Document Eng. (DocEng)*, 2008, pp. 289–297.

[15] F. Boronat, D. Marfil, M. Montagud, and J. Pastor, "HbbTV-compliant platform for hybrid media delivery and synchronization on single- and multi-device scenarios," *IEEE Trans. Broadcast.*, vol. 64, no. 3, pp. 721–746, Sep. 2018.

[16] *Dynamic Adaptive Streaming Over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats*, Standard iSO/IEC 23009-1, ISO/IEC, 2012.

[17] H.-J. Yim, S. Kim, B.-M. Lim, S.-I. Park, and N. Hur, "Application-based targeted advertisement system for ATSC 3.0 UHD service," *IEEE Trans. Broadcast.*, vol. 67, no. 1, pp. 56–67, Mar. 2021.

[18] W. Zhang, Y. Wu, N. Hur, T. Ikeda, and P. Xia, "FOBTV: Worldwide efforts in developing next-generation broadcasting system," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 154–159, Jun. 2014.

[19] A. Abdelli and N. Badache, "Context-aware adaptation of multimedia documents for consistent presentations," *Multimedia Syst.*, vol. 17, no. 6, pp. 465–486, Nov. 2011.

[20] World Wide Web Consortium. (2008). *Synchronized Multimedia Integration Language Smil 3.0 Specification*. World-Wide Web Consortium Recommendation. [Online]. Available: http://www.w3c.org/TR/SMIL3

[21] *Hybrid Broadcast Broadband TV*, Standard ETSI. ETSI TS 102 796 V1.4.1, European Telecommunications Standards Institute, 2016.

[22] R. Pantos and W. May. (2021). *Http Live Streaming*. [Online]. Available: http://tools.ietf.org/html/draft-pantos-http-live-streaming-07

[23] M. J. Perez-Luque and T. D. C. Little, "A temporal reference framework for multimedia synchronization," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 1, pp. 36–51, Jan. 1996.

[24] M. Josué, D. Muchaluat-Saade, and M. Moreno, "Preparation of media object presentation and sensory effect rendering in mulsemedia applications," in *Proc. 24th Brazilian Symp. Multimedia Web*, Oct. 2018, pp. 45–52.

[25] D. Flanagan, *JavaScript: The Definitive Guide*. Sebastopol, CA, USA: O'Reilly Media, 2006.

[26] R. Ierusalimschy, *Programming in LUA*, 2nd ed. South South Korea: LTC, Mar. 2006.

[27] M. C. Buchanan and P. T. Zellweger, "Automatic temporal layout mechanisms revisited," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 1, no. 1, pp. 60–88, Feb. 2005.

[28] G. V. Rossum, J. Jansen, K. S. Mullender, and D. C. Bulterman, "CMIFed: A presentation environment for portable hypermedia documents," in *Proc. 1st ACM Int. Conf. Multimedia*, 1993, pp. 183–188.

[29] M. Josué, M. Moreno, and R. Costa, "An adaptable transmission management framework for push-mode hypermedia content," in *Proc. 23rd Brazillian Symp. Multimedia Web*, Oct. 2017, pp. 349–356.

[30] B. Wang, J. Wang, and H. Lu, "Exploiting content relevance and social relevance for personalized ad recommendation on internet TV," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 9, no. 4, pp. 1–23, Aug. 2013.

[31] E. Kim, S. Pyo, E. Park, and M. Kim, "An automatic recommendation scheme of TV program contents for (IP)TV personalization," *IEEE Trans. Broadcast.*, vol. 57, no. 3, pp. 674–684, Sep. 2011.

[32] T. Bozios, G. Lekakos, V. Skoularidou, and K. Chorianopoulos, "Advanced techniques for personalized advertising in a digital tv environment: The imedia system," in *Proc. eBus. eWork Conf.*, 2001, pp. 1025–1031.

[33] R. M. de Resende Costa, M. F. Moreno, R. F. Rodrigues, and L. F. G. Soares, "Live editing of hypermedia documents," in *Proc. ACM Symp. Document Eng. (DocEng)*, 2006, pp. 165–172.

[34] *Mediatek s900*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.mediatek.com/products/digitalTv/s900-mt9950/

[35] *What are the Advantages of Quad Core Processor in Samsung 7/8/9 F Series Smart TV*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.samsung.com/hk_en/support/tv-audio-video/what-are-the-adva%ntages-of-quad-core-processor-in-samsung-7-8-9-f-series-smart-tv/

[36] *lG Intelligent Processor A9 Vs A7: A9 Vs A7 Smart Processors, Specifications and Comparison*. Accessed: Aug. 20, 2021. [Online]. Available: https://en.tab-tv.com/?p=14681

[37] *Monitoring Report of the Telecommunications Department (In Portuguese)*, ANATEL, Brasília, Brazil, 2018.

MARINA IVANOV P. JOSUÉ received the Ph.D. degree in computer science from Fluminense Federal University (UFF), in 2021. She is currently a Postdoctoral Researcher with the Postgraduate Program in Computer Science, UFF, actuanting on e-health and digital TV projects. Her research interests include multimedia systems, computer networks, IPTV, and digital TV.

MARCELO F. MORENO received the Graduate degree in computer science from the Federal University of Viçosa, in 2000, and the master's and Doctorate degrees in informatics from the Pontifical Catholic University of Rio de Janeiro, in 2002 and 2008, respectively. He is currently an Associate Professor at the Federal University of Juiz de Fora, where he is also a Permanent Member of the Graduate Program in Computer Science. He is also a Rapporteur (the Leader) of ITU-T Question 13/16 (multimedia application platforms and end systems for IPTV) of the International Telecommunications Union (ITU), the Co-Chair of the ITU Intersectorial Group on Broadcast-Broadband Integration (IRG-IBB), and the Vice-Chair of the ITU-T Study Group 16 (ITU-T SG16 Multimedia). He is an Editor of several ITU recommendations, including ITU-T H.761 "NCL and Ginga-NCL" and technical papers. He is also the Coordinator of the Middleware Technical Group, Brazilian Digital TV System Forum (SBTVD). His research interests include operating systems, computer networks, and distributed systems, with an emphasis on infrastructure for multimedia communication.

DÉBORA C. MUCHALUAT-SAADE received the Ph.D. degree in computer science from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), in 2003. She is currently a Full Professor with the Department of Computer Science, Fluminense Federal University (UFF), Brazil. She is also one of the Founder and the Head of the MídiaCom Research Laboratory. She has contributed to the design and development of the Ginga-NCL middleware, used in the Brazilian digital TV standard and IPTV services. Her research interests include multimedia systems, computer networks, the IoT, smart grids, and e-health.