R&D Progress on TV 3.0 Application Coding Layer

Marcelo F. Moreno (Fraunhofer IIS / UFJF),
Carlos Pernisa Júnior, Eduardo Barrere, Stanley Teixeira, Cristiane Turnes (UFJF),
Li-Chang Shuen, Carlos de Salles Soares Neto (UFMA),
Débora Christina Muchaluat Saade, Marina Ivanov Josué, (UFF),
Joel A. F. dos Santos (CEFET-RJ),
Sérgio Colcher, Daniel de S. Moares (PUC-Rio),
Derzu Omaia, Tiago Maritan Ugulino de Araújo, Guido Lemos de Souza Filho (UFPB)

Abstract—**This paper outlines the methodology, progress, and initial outcomes of a collaborative R&D effort by 40 researchers from six academic institutions, focused on addressing critical application-coding requirements for SBTVD Forum's TV 3.0 Project, Phase 3. The team is developing prototypes and use cases applications to validate and demonstrate TV 3.0 application coding features. Key developments include architectural changes, a persistent media player, a viewer's journey design, besides extensive research in requirements engineering, user data analysis, and novel codec support. The team also explored application coding extensibility, enhanced accessibility, immersive experiences and multimodal interaction. During SET Expo 2023, partial implementations of the prototyped use cases were showcased, highlighting the project's progress and significance. This paper provides technical details and diagrams, facilitating a thorough discussion of this innovative project.**

Index Terms—**Application coding, TV 3.0, Application-based TV experience, Personalized TV experience, Immersive TV experience, accessibility.**

I. Introduction

Television plays a social role of immense relevance in Brazil. TV is more than a technological object in the room: it is also an important cultural artifact and an element of national integration. Therefore, any technological evolution that represents a change in the way of watching television may imply some cultural change for the society itself. Thus, the development process of SBTVD Forum's TV 3.0 Project [1], at least for its most highlighted application coding use cases and features, are being carried out not only as a technology research, but also as a social study.

Brazil has been watching TV since 1950, with the inauguration of TV Tupi in São Paulo. In seven decades, Brazilian television has experienced technological, social and content development that makes the national experience one of the richest in the world. Remarkably, the current terrestrial DTV system specifies the Ginga middleware, a national technology, as the standard for multimedia interactivity since 2007. Ginga has been proven to support a consistent evolution that made it the first Brazilian technology adopted as an international standard by ITU-T in 2009 [2] and recognized by ITU-R as an integrated broadcast-broadband system since 2017 [3].

TV 3.0 Project is currently under Phase 3[1], carrying out further tests and evaluations over the physical and video coding layers, as well as developing a reference mux/demux. Regarding the application coding layer, most of the innovative requirements established by the Call for Proposals (CfP) [4] are under study by selected Academia research groups, since those requirements were not appropriately addressed in the previous phases [5]. The CfP specified 17 requirement groups for application coding, including basic aspects on backward compatibility with Ginga specifications and its implementation reuse, besides support for TV 3.0 underlying technologies. The advanced requirements include support for application-based TV experience, immersive audiovisual content, multimodal interaction, sensory effects, multi-user profiling, audience measurement, IP convergence, and extensibility, just to name a few of them.

This paper focuses on the methodology, progress and early achievements of the Academia R&D team on addressing the high-priority application-coding requirements for TV 3.0. The team is composed of 40 researchers from 6 academic institutions, namely PUC-Rio, UFPB, UFF, UFJF, UFMA and CEFET-RJ. The work started in April 2023.

As a means of actively collaborating with the research methodology, SBTVD Forum's Technical and Market Modules jointly decided on a prioritization of requirements to determine the sequence of studies for the R&D team. In addition, the Forum's Application Coding Working Group (WG) specified initial guidelines on how to tackle each requirement, based on the evaluation results from Phase 2 and the WG's expertise on standardizing/implementing digital TV middleware. Finally, the WG specified a total of 7 use cases to be prototyped, aiming at validating the R&D solutions for the prioritized requirements and publicly demonstrating the new TV 3.0 application coding features. The R&D team diligently incorporated all SBTVD Forum contributions into its methodology, allowing for consistent progress on certain requirements.

[1] TV 3.0 Project Phase 3 is funded by the Brazilian Ministry of Communications (MCom), managed by the Brazilian Network for Education and Research (RNP).

First, relevant architectural changes in the application coding layer were already proposed and agreed, based on the fundamentals of Ginga specifications for Profile-D receivers. The requirements on application-based TV experience impose changes that include a new user interface for listing each broadcaster's initial application. In addition, a new media player is needed, capable of keeping running over application switches, regardless of whether the current audiovisual content is delivered over-the-air (OTA, broadcast) or over-the-top (OTT, broadband).

There is also significant effort in requirements engineering and social studies regarding this application-based TV experience, as mentioned before. The team is running focus groups and opinion polls with a probabilistic sample, as well as prototyping the entire viewer's journey based on the quanti/qualitative data obtained. This prototype will be further refined following the principles of Design Thinking, under discussion with a team of experts proactively assigned by RNP (Brazilian Network for Education and Research).

Another area of focus involves evaluating the features introduced by the adopted audio and video codecs, with the objective of identifying properties that can be utilized by applications and determining the necessary implementation support.

The extensibility requirement is also under study, with a focus on identifying Ginga-NCL and Ginga Common Core WebServices (Ginga CC WS) APIs as metadata so that new applications can obtain granular information about the functionality support of their interest in the receiver, thus allowing them to adapt according to the available resources.

A further R&D task has focused on the accessibility requirements, more specifically on the captioning part, where IMSC1 standard is adopted for encoding and transmitting subtitles and sign language gloss. It uses a subset of TTML, which consists of an XML file with several possible settings for captioning, such as position, color, font, display time, synchronism, emojis and images. In order to test and validate the forwarding of captions and gloss to mobile devices over the local network, a prototyping environment was developed, composed of a partial Ginga CC WS implementation and NCLua and HTML5 applications. New required APIs are added to Ginga CC WS prototype, in this case for the real-time forwarding of captions and gloss in TTML format over sockets or websockets. The synchronism between the applications is performed by the Ginga CC WS, which delivers the same content, at the same time, for all socket clients, the results are being evaluated. The gloss stream is shown in the application by a 3D avatar playing sign language.

The team has also been working on the implementation of use cases related to sensory effects, immersive content and multimodal interaction. To accomplish this, the team is working on harmonizing the adopted proposals NCL 4.0 [6] and Guaraná [11]. Combined, they allow for the inclusion of sensory effects (wind, scent, light, …) in interactive multimedia applications and the execution of parts of the application on head-mounted displays connected to the TV, in a 360° scene including 3D objects, immersive MPEG-H 3D audio [12] and traditional multimedia objects. In addition, users will be able to interact with applications using different modalities (voice, gestures…) using input recognition devices.

In conclusion, partial implementations of prototyped use case apps were demonstrated at SET Expo 2023 in the SBTVD Forum booth. This paper is structured as follows: Section II discusses the rationale and agreed-upon changes to the application coding architecture for TV 3.0. Section III describes the application-based TV experience and a projected viewer's journey, providing context for focus group discussions. Sections IV reports progress on application coding support for TV 3.0 audio/video codecs and application coding extensibility. Section V examines developments in supporting accessibility content, including second-screen delivery of captions and sign language glosses. Section VI presents achievements in immersive experiences support, encompassing sensory effects, multimodal/multiuser, and virtual reality content. Finally, Section VII offers concluding remarks. This paper includes diagrams and technical details for deepening the discussion on each study of this challenging project of unparalleled opportunity.

II. Application Coding Architecture for TV 3.0

Several of the use cases designed for TV 3.0 clearly indicate the need for intensive use of multimedia applications, which, according to CfP TV 3.0 [4], will be based on extensions to Ginga specifications for Profile-D receivers (a.k.a. DTVPlay). Undoubtedly, it is through Ginga applications that broadcasters and partners will be able to leverage TV3.0 greatest innovations, including personalization of the TV content consumption experience, segmented programming, manipulation of additional content in more immersive formats, accessibility, sensory effects, as well as new forms of interaction. In addition, it is through Ginga applications that it will be possible to build and manipulate viewer profiles that enable such personalized experiences, obviously for the viewers who consent.

It is therefore expected that Ginga applications will be running and switching constantly on TV 3.0 receivers, leading to a need to rethink the application coding support specifications. Application coding becomes no longer an accessory for broadcasters, but a key element for enabling the vast majority of new TV 3.0 use cases. The CfP makes this clear through its requirements group AP6 "Enable application-oriented TV" [4]:

- AP6.1: application-oriented user experience with TV
- AP6.2: handling the presentation of all audiovisual content
- AP6.3: application switching delay (lower is better)

This represents, objectively, a paradigm shift, which in fact needs also to be discussed from the viewer's point of view, according to our study on the viewer journey possibilities, presented in Section III. Nevertheless, this evolution towards an application-oriented TV has to be reflected in the receivers' application coding support architecture. The challenge set to the TV 3.0 Project R&D team was to propose adaptations to the current Ginga architecture, according to ABNT NBR 15606-1, in order to maintain compatibility and to reuse, as much as possible, the existing implementations of TV 2.5 middleware components and subsystems.

The TV 2.5 middleware architecture can be depicted as shown in Figure 1. In summary, this architecture demonstrates the capability for broadcasters to transmit Ginga applications via OTA that are developed using NCL/Lua or HTML5/ Javascript languages. Consequently, the Ginga-NCL [13] or Ginga-HTML5 [14] presentation engines execute these applications based on the OTA signaling rules provided. The current APIs of both presentation engines offer interesting possibilities for integrating broadcast and broadband features in D-profile receivers. These features encompass receiving broadband content through adaptive streaming, with or without DRM protection, as well as facilitating TCP and UDP communication in both client and server modes. Additionally, the architecture supports content preparation to enhance the quality of the viewing experience during the seamless transition to broadband, including the insertion of targeted advertising, among other functionalities.

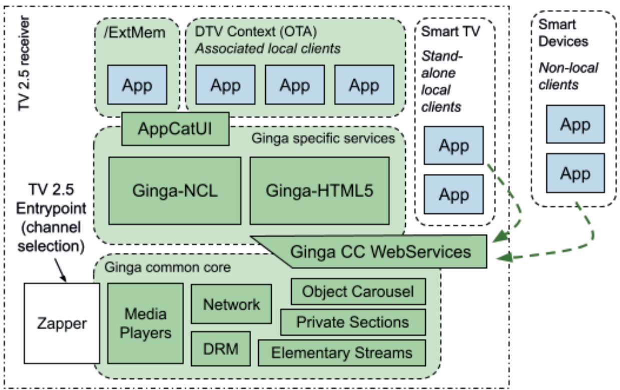

Figure 1. TV 2.5 middleware architecture.

In the case of Ginga-HTML5, since it exclusively employs W3C standardized APIs, all these resources are accessible through a decoupled API outlined in the Ginga CC WebServices specification [15]. This decoupling is achieved by utilizing a remote API that adheres to the RESTful architectural style. Consequently, this approach enables the same Ginga CC WebServices to provide TV 2.5 resources to applications beyond broadcasting, including those operating within the smart TV environment on the same receiver or on any smart device within the home network. However, access permissions must be granted by both the viewer and the broadcaster in such cases.

Even Ginga-NCL applications can utilize Ginga CC WebServices resources, particularly for use cases that encourage their integration with second-screen applications running on smart devices. For all other use cases, Ginga-NCL directly furnishes the required APIs through the NCL and Lua languages, potentially offering improved performance when accessing resources.

Starting from Profile C receivers[2], a component known as AppCatUI (Application Catalog User Interface) becomes available. This component serves the purpose of listing applications accessible within the current DTV context, allowing viewers to trigger them. It also facilitates viewers in adding and removing applications, making them persistent, and initiating their execution. Such applications can be delivered OTA with appropriate installation permissions, installed from external memory devices (/ExtMem), or downloaded from authorized repositories accessible via broadband connections.

However, despite offering these possibilities, the current TV 2.5 specifications fall short of enabling an application-oriented TV approach for several reasons. Firstly, this limitation arises because the initial entry point into the TV content consumption experience revolves around the channel abstraction, typically managed by native software responsible for channel switching, depicted in Figure 1 with the suggestive term “zapper”. Notably, this “zapper” is not an integral part of the middleware specifications and, so, Ginga applications have only limited control over the behavior of the zapper. When they intend to present OTT content, for example, they are required to employ an additional media player, which usually has a lifecycle closely tied to the application itself. If an attempt is made to switch to another application, the current player instance would be terminated. This underscores the necessity of incorporating a persistent media player as an essential element of the new architecture for TV 3.0 application coding.

Moreover, the application-oriented paradigm holds the potential to conceal the concept of traditional channels, presenting each broadcaster as an application capable of providing access to its complete ecosystem of content and services. To achieve this goal, it is imperative to introduce a user interface, a component of the application coding layer, that can list each broadcaster's initial application and provide access to other applications offered by broadcasters for installation or execution. The existing AppCatUI can indeed list installed Ginga applications and those available in the current DTV context, but it lacks the capability to showcase an application at the primary UI level for each broadcaster. Its current channel-oriented approach restricts its prominence.

To address these specifics, the proposal, well-established between the R&D team and the SBTVD Forum for an application-oriented TV, suggests making adaptations to existing middleware architecture components. These changes aim to minimize the impact on current implementations while allowing viewers to embrace the new paradigm. The proposed architecture for coding TV 3.0 applications is depicted in Figure 2.

The revamped AppCatUI evolves into a free-to-air TV super app, serving as a front-end for viewers to identify available broadcasters, explore their content ecosystems, and configure profiles and other potential options previously absent in receivers. In this architecture, a persistent media player enables seamless switching between OTA and OTT content within broadcasters' applications without playback interruptions. The next application can then decide whether to maintain the current content, recommend new content, or

[2] Profile C was specifically designed for receivers distributed during the analog TV signal switch-off process, representing a significant evolution addressing advanced requirements for implementing public policies and digital inclusion [16]

switch automatically to other content, all with the viewer's best experience in mind.

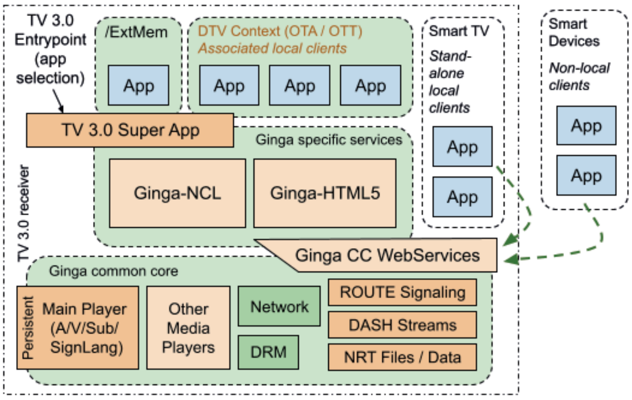

Figure 2. Proposed TV 3.0 application coding architecture.

The figure also incorporates new components to access resources from other TV 3.0 system layers, notably the transport layer based on the ROUTE/DASH specification. Additional extensions to this architecture, discussed in Section VII, support integration with sensory effects, advanced interaction recognition, and virtual reality devices. It is anticipated that APIs supported by Ginga-NCL and Ginga CC WebServices will be expanded to accommodate solutions for the ongoing requirements.

To list an initial application for each broadcaster within the so-called free-to-air TV “super app”, the conventional and time-consuming channel scanning process can be presented to viewers as an application discovery process. To expedite this process in TV 3.0, various options are being considered. These include the automatic instantiation of each application by the free-to-air TV super app, eliminating the need for individual application downloads for each station found. Additionally, when the receiver is connected to the internet, a new discovery web service could provide a list of licensed broadcasters in the installation region. This could enable automatic instantiation of an application, even in cases of weak signals, directing viewers to the broadcaster's linear OTT content if available. Another advantage of such a discovery service is the potential for automatic updates to the applications list, eliminating the need for repeated scanning processes as currently required for adding new TV channels.

These innovative possibilities are designed to enhance the viewer's experience and can be assessed through experimental prototyping of the entire viewer journey, even involving TV receiver aspects beyond the scope of free-to-air digital TV standards. The subsequent section outlines the initial steps in designing a possible viewer/interactor journey.

III. Application-based TV Experience

Application-oriented TV is the central concept for understanding the paradigm shift proposed for the third generation of Brazilian digital TV. The change is technological, but also cultural, as it alters the way viewers traditionally relate to accessing the content offered by broadcasters. In the proposed model, channels will be replaced by applications offered by broadcasters. From this initial application, each broadcaster will be able to create its own ecosystem of internal apps and offer content both OTA and OTT, which the viewer will access depending on the existence of connectivity on their Smart TV.

This evolution makes sense in a scenario in which the predominant Smart TVs on the market already offer FAST (Free Ad-Supported Television) channels, which compete without regulation for the audience with open channels distributed by broadcast. In addition to FAST channels, there is a whole range of streaming applications that occupy the screen and, in the case of televisions connected to the Internet, monopolize viewers' choice. In fact, there are even keys dedicated to streaming services on the minimalist remote controls. In fact, on these televisions, it is increasingly difficult for viewers to find the free-to-air TV channels whose content they want to consume. The proposal for an application-oriented television paradigm attempts to resolve this issue by offering the viewer an experience in which they can easily identify the devices' native apps and the applications from free-to-air broadcasters.

To facilitate the design of a possible viewer/interactor[3] journey (see subsection III.A), we firstly focused on assessing the interfaces of current smart TVs and studied video streaming on digital platforms. The examination included studying the most commonly used smart TV models in the Brazilian market, based on operating systems such as Roku TV, Android TV or Google TV, WebOS, and Tizen. These platforms exhibit differences in content presentation and viewer interaction. Roku TV and Android TV prioritize application presentation, while WebOS and Tizen focus on keeping audiovisual content on the screen for extended periods, overlaying settings, menus, and other applications only when activated by the viewer. Across all systems, free-to–air TV content occupies a distinct application space, varying in colors, icons, and terminology.

In the analysis of streaming services and their interfaces, we aimed to identify familiar paths for audiovisual consumers, seeking to adapt these experiences to new interactive actions and requirements for a possible TV 3.0 super app interface. We observed that different platforms often share similarities in presenting content on their home screens, primarily dedicated to on-demand content. Tabs categorize content by genre and format, such as drama, comedy, sports, and news. Live content is typically featured within specific applications. For instance, Globoplay includes live TV content, including simultaneous broadcasts from Globo and other affiliated channels. RTVEplay prominently displays live content on its platform, and Pluto TV offers live content on its home page while organizing on-demand content in a separate tab.

Initial findings have been incorporated into a proposed viewer’s journey model, including icon arrangement on TVs, the importance of a universally recognizable identifier for free-to-air TV, and the need to carefully consider the relationship between viewers/interactors and the TV 3.0

[3] Murray [18] defines an interactor as someone who effectively interacts with content on a media device.

super app interface to ensure a seamless transition for those accustomed to traditional TV.

We are also exploring the concept of a "networked time" called "Timelink" to free viewers from rigid linear TV scheduling. To establish Timelink and provide viewers with time control, an intuitive program guide using deep links in Electronic Program Guide (EPG) metadata sent by broadcasters is crucial. While browsing the guide, viewers can access detailed information about each content item and initiate playback with a simple click. The guide can also signal which content is immediately available, considering viewer preferences and internet connectivity. This guide streamlines access to both OTA and OTT content.

In a non-linear TV landscape, a more efficient and intelligent use of remote control, particularly the colored buttons, is under investigation.

Lastly, the concept of a second screen is being reimagined as a mirror of the television screen on a separate device, a departure from the current practice of integrating the remote control within the TV interface.

A. A VIEWER'S JOURNEY PROPOSAL FOR EVALUATION OF THE APP-ORIENTED TV EXPERIENCE

The proposal for a viewer's journey of an application-based television, reproduced below, considers cultural, social and economic aspects of how Brazilians consume audiovisual content on free-to-air TV from the time the device is turned on for the first time to be configured until the moment the viewer chooses and watches what is being broadcasted.

An important TV 3.0 feature lies in personalizing content according to the viewer's preferences. When turning on the TV for the first time, the viewer is invited to choose the configuration language, which includes accessibility options with audio description. This is followed by the possibility of defining a profile that can be shared with broadcasters, with the definition of important characteristics such as whether it is a child and what age recommendation for content is suggested. Figure 3 illustrates this viewer profile creation screen.

Figure 3. Viewer profile screen.

Instead of the channel scanning process, in TV 3.0 there is the process of discovering initial apps from broadcasters. This process is based on the geographic location of the receiver, so it will list the stations that are available in that region. Figure 4 illustrates the discovery of three broadcasters.

In Figure 5, the purpose of this screen is that the viewer can easily identify where free-to-air TV stations are found on Smart TVs. This screen represents a possible harmonization with a clear indication of what are OTT streaming apps, broadcast apps and FAST channels.

Figure 4. Broadcaster scanning.

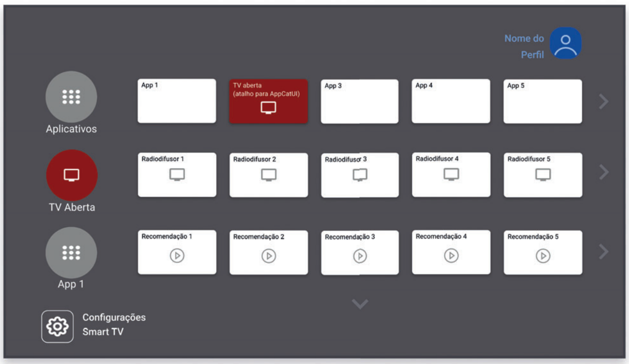

Figure 5. Home screen harmonizing streaming and broadcast TV apps.

Figure 6 shows the super app environment, a screen where all the initial apps for free-to-air TV broadcasters in the region are listed for the viewer. This screen corresponds to the usual broadcast TV screen, which is zapped using the remote control in the traditional way.

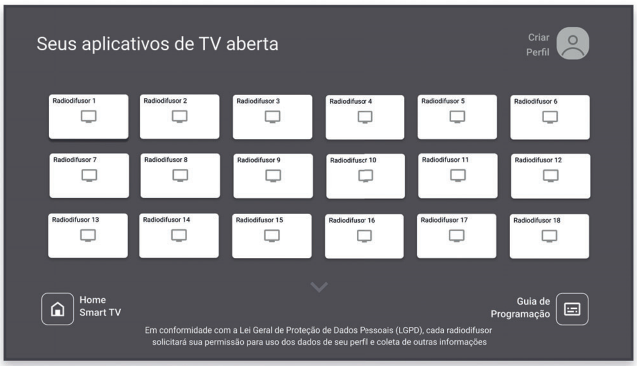

Figure 6: Super app environment.

Figure 7 illustrates the initial app of a broadcaster. Here, the viewer has access to traditional, linear audiovisual content that comes over the air. To watch this content, the TV receiver does not need to have an active Internet connection and the broadcaster has some control over it, including visual identity settings to be applied to such a common initial app.

Finally, in Figure 8, we see the recommendation ecosystem of a broadcaster that suggests both OTA and OTT content. Depending on the receiver's connectivity, they will have access to a larger catalog of content.

Figure 7: Broadcaster initial app.

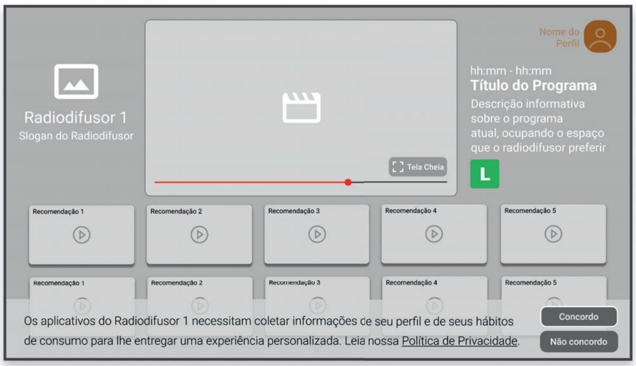

Figure 8: Broadcaster content recommendation ecosystem.

IV. Application coding support for TV 3.0 audio/video codecs and extensibility

Following the recommendations resulting from the tests and evaluations of TV 3.0 Project Phase 2, new video and audio codecs will be incorporated into TV 3.0. At the application layer, the R&D team has been working on how to provide support so that applications can use the advances and features of these new media. Efforts have been put into finding and integrating decoders and players for the video codecs adopted for standardization - Versatile Video Coding (VVC) ISO/IEC 23090-3, MPEG-I part 3 and Low Complexity Enhancement Video Coding (LCEVC) MPEG-5 Part 2 - which brings with it several advantages, such as better video quality with lower bit rates than its predecessors. Furthermore, the team is also committed to integrating the MPEG-H audio (ISO/IEC 23008-3) player, developed by the proponents themselves, which in addition to reproducing the new immersive audio standard, also features an interface that allows interaction with audio objects and customization of the various functionalities offered, such as changing channels to choose a track, selecting a language, etc.

Concerning the extensibility requirement, the team has focused on surveying the APIs for listing receiver properties and resources, both in Ginga-NCL and Ginga CC WS. The idea is that these APIs can be updated and harmonized according to new features introduced with the TV 3.0 project, such as version 4.0 of the NCL language. Therefore, TV 3.0 applications will be able to consult what features and functionalities are available on a receiver, and thus be able to adapt to them. This can allow even different receivers to run applications with adapted functionalities.

V. Application coding support for accessibility content

To test and evaluate the transmission and reception of captions, sign language glosses and audio description a prototyping environment was proposed.

This environment is centered on the Ginga CC WS server, which manages the distribution and synchronization of the accessibility media. A REST API, which is still under development, has been extended from the existing one in TV2.5. It provides routes that allow a client, external to the TV and connected to the same local network, to access this media, as long as it is authenticated on the TV.

In this way, client applications can request and access the sign language gloss, subtitles and audio description media. From this, various scenarios can be explored. In the case of hearing-impaired viewers, the glosses received can be displayed in Sign Language visual format on the user's device, without overlaying the video being shown on the TV screen. In the case of subtitle display, different viewers can access subtitles in the language they prefer, allowing different people to receive different content. And the audio description client running on their cell phone can receive the audio so the user can listen to it through headphones, without disturbing other viewers.

This allows content to be customized simultaneously and in a non-imposing way, since each user can have their own customization on their personal device, without interference from others.

Figure 9 shows this environment and demonstrates the three scenarios presented. It's possible to see the Ginga CC WS server on the TV delivering the 3 accessibility media contents to the mobile devices via a Wi-Fi network. Each device receives its media and plays it according to its type. An accessibility user can view or listen to the content received on the devices.

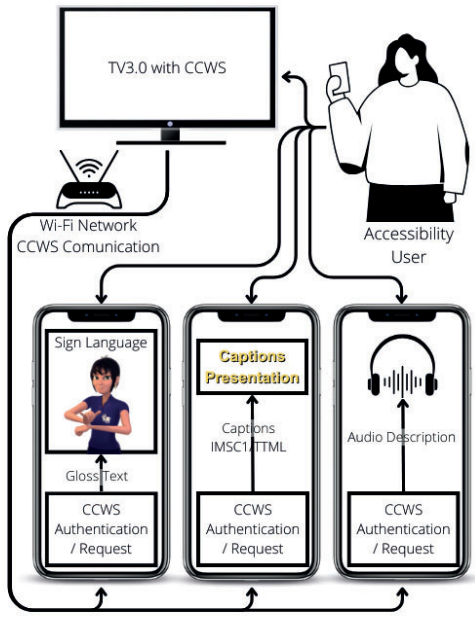

Figure 9. Accessibility prototyping environment.

The Ginga CC WS implementation was done in node.js and it relies on a static cyclic content to simulate a real environment.

The media to be provided were prepared in such a way that the subtitles, glosses and audio description had equivalent content. To do this, the subtitles for a given video were generated manually, followed by their translation into glosses using the VLibras translator [10]. The audio file was produced and recorded by one of the team members. To simulate a real broadcast environment, in which subtitles and glosses are broadcast continuously, they were segmented into 2-second chunks, and each was stored in a different TTML file. In this way, the content is transmitted every 2 seconds to clients connected via sockets or websockets. At the end of the file transmission, the Ginga CC WS server restarts the cycle, transmitting the first files again.

The test application was created using HTML5 and Javascript. It offers both a desktop version suitable for TV screens and a mobile version optimized for mobile devices. Regardless of the platform it's accessed from, the application consists of three main modules: one for displaying sign language, another for showing captions, and a third for playing audio descriptions.

The visual representation of captions in IMSC1 format is done using the open-source library imscJS [8]. This library interprets the content present in IMSC1/TTML subtitle files, allowing subtitles to be displayed in the application with the appropriate graphic formatting.

The sign language module receives the glosses from the Ginga CC WS server via a websocket. For the representation in sign language format, the application was integrated with the VLibras Widget [7], a tool that has a 3D avatar that reproduces the glosses in sign language format.

The audio description module requests the Ginga CC WS server to this media and receives the HTTP URL for the audio in DASH format [9]. The audio is then played back and the user can listen to it on their mobile device's speaker or through connected headphones.

Figure 10 shows this application running during SET Expo 2023. It's possible to see the caption and sign language modules in execution at the same time on TV and on a tablet device.

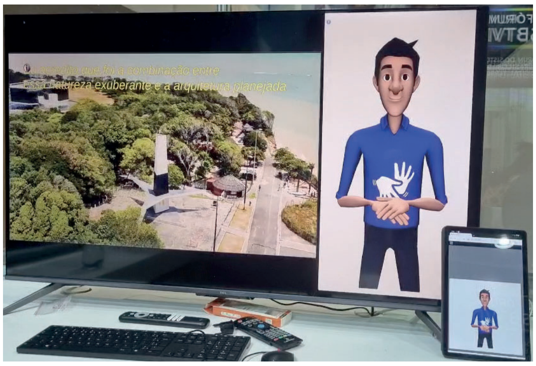

Figure 10. Accessibility application for mobile and TV.

VI. Application coding support for immersive experiences

Regarding the support for immersive experiences, the R&D team presented two use case applications during SET Expo 2023. The first focused on the execution of sensory effects and multimodal/multiuser support. The second allowed executing parts of the application on a head-mounted display (HMD) connected to the TV.

Sensory effects are used in entertainment (e.g cinema and games) to increase the user experience providing more immersion when consuming content. Aiming to provide immersive experiences in Digital TV environments, NCL 4.0 allows integrating sensory effects into interactive TV applications. In the first immersive experience use case, the R&D team specified an NCL 4.0 application that allows synchronizing light and aroma sensory effects with the audiovisual content transmitted by the broadcasters.

For the Ginga middleware to support the execution of the multisensory applications specified in NCL 4.0, it is necessary to add components capable of communicating with sensory effect renderers in the DTV receiver environment. In this way, the multisensory application will be able to activate/deactivate sensory effects and control effect presentation characteristics, such as position, the light effect color, the smell of the aroma effect, etc.

The sensory effects rendering is performed by the component named *Sensory Effect Renderer* present in Ginga common core as illustrated in Figure 11. Each *Sensory Effect Renderer* effect is associated with only one sensory effect and vice versa. The renderer defines interfaces enabling the Ginga-NCL formatter to communicate with the rendering devices and trigger actions such as starting the effect presentation or preparing a sensory effect to guarantee temporal synchronization of the application.

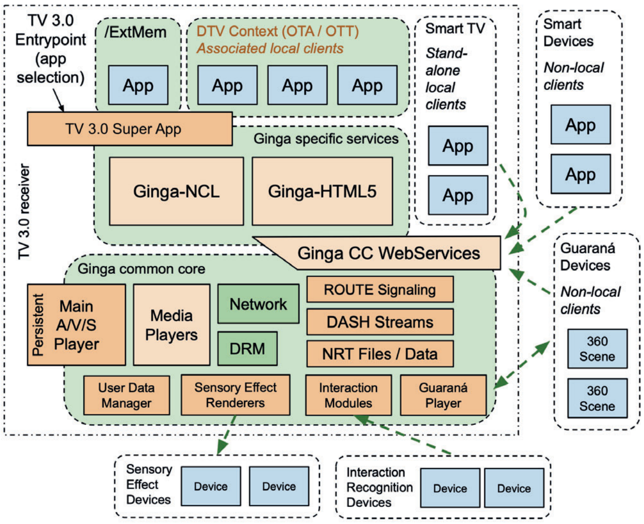

Figure 11. Ginga architecture to support immersive applications

The physical devices responsible for rendering effects can be from different manufacturers and implement different communication protocols. This communication must be implemented by the Device API, which is specific to each rendering device. The Device API must implement a set of functions such as connection to the physical device, activation/deactivation and commands to modify the effect intensity.

Another feature of TV 3.0 is the support of user interaction using different modes of interaction such as gestures, voice and even facial expression recognition. For example, a broadcaster can transmit an NCL 4.0 application capable of adapting the content presented according to the viewer's facial expressions that it identifies.

User interactions with the multimedia application are managed by *Interaction Modules* present in Ginga common core. These interaction modules communicate with physical devices through predefined methods and notify the middleware when an interaction has been recognized. Additionally, the module can inform the middleware the user that performs an interaction if the recognition device is capable of identifying the user.

The first demo application showcased during SET EXPO 2023 is a travel show that presents four videos related to tourist attractions in the city of Rio de Janeiro. Initially, the application presents two videos related to beaches of Rio de Janeiro (Ipanema and Copacabana). In both videos, a sea aroma is triggered by the application. Furthermore, a yellow light effect is presented when the sunset appears in Ipanema, and a blue light effect happens when it is a sunny morning in Copacabana. The third video presents the Botanical Garden that is synchronized with a green light effect. Additionally, the viewer can interact with the third video using gesture interaction to pause or resume the video, as illustrated in Figure 12.

Figure 12. Support for gesture interaction.

Finally, the last video presents Christ the Redeemer and a voice-based viewer interaction is asked in order to choose the last part of the show. Based on the viewer's choice, a personalized fifth video is presented. When a viewer interacts, his/her profile identification, which is already registered in the TV receiver, is recognized and shown, as presented in Figure 13. This use case also demonstrates that TV 3.0 can identify the viewer that interacts with it.

Figure 13. Support for user interaction identification.

In the use case related to the Guaraná proposal, the user starts its experience on a broadcaster's application. In that application the user has the option to watch a program where additional content is executed in an HMD. That program presents a classical music performance inside the Tiradentes Palace in Rio de Janeiro. The same content presented on the TV is available in 360º in the viewer's HMD. Together with the 360º video, the application presents photos of the palace and a video presenting the palace's architecture. Whenever the viewer turns its head in the direction of the orchestra conductor, an image describing his biography is presented. Figure 14 presents an overview of the 360º scene presented in the HMD.

Figure 14. 360º scene overview.

To implement that use case the R&D team used an implementation of the middleware Ginga Common-Core Web Services component (CC-WS) with the new API proposed for registering remote devices that will execute part of the application (in this case, a 360º scene). Once a device registers itself as a remote device, the CC-WS creates a WebSocket for the bidirectional communication between CC-WS and HMD. Once the main application (executed at the TV) starts its execution, *i.e.*, the 2D version of the orchestra presentation, at the TV, the CC-WS component transmits to the HMD the description of the 360º scene. As the presentation unfolds, the CC-WS component sends commands instructing the HMD to start/stop presenting the content in the scene. Whenever the Guaraná logo at the orchestra conductor is in the user field of view, the HMD sends to CC-WS an interaction report indicating the start of the View event of the logo. The same is performed when the logo exits the field of view, triggering the end of the View event. Once the CC-WS receives a start/stop of the View event, it replies to the HMD with a command to start/stop the conductor biography.

VII. Final remarks

In conclusion, this paper presents a comprehensive exploration of the innovative TV 3.0 project, focusing on the R&D progress on various aspects of the application coding layer.

TV 3.0 project has ushered in a new era of television technology by prioritizing an application-oriented TV experience. Through meticulous research and development, the project has been fundamental to rethink how viewers interact with television content. The transformation of the traditional TV interface into a versatile, application-centric platform has the potential to enhance user engagement and to offer viewers a personalized control over their content consumption.

However, it's important to acknowledge that the TV 3.0 project remains a work in progress. The application coding requirements addressed in this paper are part of an ongoing journey, and solutions will continue to evolve. As the project progresses, it is expected that the remaining requirements will also be tackled with innovative solutions, further enhancing the TV 3.0 possibilities.

Extensibility plays a crucial role in the TV 3.0 project, with a commitment to evolving APIs in Ginga-NCL and Ginga CC WS. This adaptability ensures that TV 3.0 applications can seamlessly integrate with a wide range of receiver configurations, accommodating diverse user preferences and hardware capabilities. The ongoing development in this area promises even greater flexibility and compatibility in the future.

Accessibility is at the forefront of TV 3.0, with a strong emphasis on customization. The project's dedication to providing tailored captions, sign language glosses, and audio descriptions ensures that television content is inclusive and accessible to a wide audience. As the project matures, these accessibility features will continue to evolve to meet the evolving needs of viewers.

The paper also delves into the realm of immersive experiences, demonstrating TV 3.0's capability to synchronize sensory effects, support multimodal/multiuser interactions, and integrate with head-mounted displays. These developments represent a significant shift in television engagement, offering viewers interactive and captivating content experiences.

The TV 3.0 project, in general, is promoting the way for a future where television transcends its traditional confines and provides viewers with unparalleled and personalized experiences. This paper serves as a testament to the exciting possibilities and innovations that lie ahead in the realm of TV 3.0, with the understanding that the journey is ongoing, and the best is yet to come.

• REFERENCES

[1] SBTVD Forum. “TV 3.0 Project”. Website. https://forumsbtvd.org.br/tv3_0

[2] ITU-T. Recommendation ITU-T H.761 (2009) “Nested Context Language (NCL) and Ginga-NCL”

[3] ITU-R. Recommendation ITU-R BT.2075-1 (2017) “Integrated broadcast-broadband system”

[4] SBTVD Forum. “Call for Proposals (CfP): TV 3.0 Project”. https://forumsbtvd.org.br/wp-content/uploads/2020/07/SBTVDTV-3-0-CfP.pdf

[5] SBTVD Forum. TV 3.0 Project - Phase 2 - Results https://forumsbtvd.org.br/tv3_0/#panel-phase2

[6] BARRETO, F. ; DE ABREU, R. S. ; JOSUE, M. I. P. ; MONTEVECCHI, E. B. B. ; VALENTIM, P. A. ; MUCHALUAT-SAADE, D. C. . Providing multimodal and multi-user interactions for digital tv applications. MULTIMEDIA TOOLS AND APPLICATIONS, v. 82, 2023. https://link.springer.com/article/10.1007/s11042-021-11847-3

[7] Universidade Federal da Paraíba (UFPB). 2023. VLibras - Governo Digital. https://vlibras.gov.br/. Online; Accessed on August 30, 2023.

[8] Pierre-Anthony Lemieux, Nigel Megitt, and Robert Bryer. 2022. ImscJS Repository. https://github.com/sandflow/imscJS. Online; Accessed on August 30, 2023.

[9] British Standards Institution. 2022. ISO/IEC 23009-1 AMD 1. Information Technology. Dynamic Adaptive Streaming Over HTTP (DASH).: Part 1. Media presentation description and segment formats. Technical Report pt. 1. https://www.iso.org/standard/83314.html

[10] Luana S. Reis, Tiago M. U. Araújo, Yuska P. C. Aguiar, Manuella A. CB Lima, and Angelina S. S. Sales. 2018. Assessment of the treatment of grammatical aspects of machine translators to Libras. XXIV Simpósio Brasileiro de Sistemas Multimídia e Web. Anais Salvador, Brasil: SBC–Sociedade Brasileira de Computação, 2018. DOI: https://doi.org/10.5753/webmedia.2018.4570.

[11] Gabriel Souza, Daniel Silva, Matheus Delgado, Renato Rodrigues, Paulo R. C. Mendes, Glauco Fiorott Amorim, Alan L. V. Guedes, and Joel dos Santos. 2020. Interactive 360-Degree Videos in Ginga-NCL Using Head-Mounted-Displays as Second Screen Devices. In Proceedings of the Brazilian Symposium on Multimedia and the Web (São Luís, Brazil) (WebMedia ’20). ACM, New York, NY, USA, 289–296.

[12] O. Major, Z. Shaban, B. Czelhan, A. Murtaza. “Immersive Audio Application Coding Proposal to the SBTVD TV 3.0 Call for Proposals”. SET International Journal of Broadcast Engineering, vol. 7, pp. 48-56 2021. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi:10.18580/setijbe.2021.4.

[13] ABNT Standard NBR 15606-2 (2023) “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”

[14] ABNT Standard NBR 15606-10 “Digital terrestrial television - Data coding and transmission specification for digital broadcasting - Part 10: Ginga-HTML5 - Ginga HTML5 profile specification”

[15] ABNT Standard NBR 15606-11 “Digital terrestrial television - Data coding and transmission specification for digital broadcasting - Part 11: Ginga CC WebServices - Ginga Common Core WebServices specification”

[16] CASTRO, Cosette; BARBOSA FILHO, André. Proyecto Brasil 4D Interactividad en televisión pública. Rev. Cienc. Soc. [online]. 2016, vol.29, n.38, pp.145-159. ISSN 0797-5538.

[17] G. K. Walker, T. Stockhammer, G. Mandyam, Y. -K. Wang and C. Lo, "ROUTE/DASH IP Streaming-Based System for Delivery of Broadcast, Broadband, and Hybrid Services," in IEEE Transactions on Broadcasting, vol. 62, no. 1, pp. 328-337, March 2016, doi: 10.1109/TBC.2016.2515539.

[18] Murray, Janet H. (1997). Hamlet on the Holodeck: The Future of Narrative in Cyberspace. New York: Simon & Schuster. ISBN 068482723-9.