A Novel IMS based UMB-SmartTV system for Integrating Multimodal Technologies

Izidor Mlakar, Danilo Zimšek, Zdravko Kačič, Matej Rojc

***Abstract*—** Several systems with multimodal interfaces are already available, and they allow for a more natural and more advanced exchange of information between man and a machine. Nevertheless, the television domain is still undergoing an innovation/development phase within which standard linear television is further enhanced with several novel technologies. In this way it is already being transformed into a full interactive entertainment environment customizable with several applications and services. Besides, TV set is a most common household device and can, therefore, represent a common platform also for smart-home environment. Current level of personalization and interactive possibilities are still quite limited, especially in terms of context-awareness, recommendation, and multiple user-control-devices (e.g. smart-phones, tablets, game-pads, keyboards, mice, etc.). Therefore, the fusion of evolving IPTV services with natural modalities can be effective solution for users that would like to access these services and IPTV content in a more natural way. In the paper a novel IMS based UMB-SmartTV system is proposed that fuses traditional IPTV services with multimodal services, including text-to-speech synthesis engine, speech recognition engine, and embodied conversational agents, available for several users even remotely. The platform enables flexible migration from often closed and purpose-oriented nature of multimodal systems to the wider scope that IPTV environment can offer. It is designed to overcome problems regarding interoperability, compatibility and integration that often accompany migrations to multiservice (and resource limited) networks. The UMB-SmartTV architecture is developed on IMS core and distributed DATA architecture. In this way it flexibly merges IPTV and non-IPTV services into uniform and highly modular solution that provides entertainment, ambience control, and many other services to the users operating with different devices and speech.

***Keywords*—**IPTV system, multimodality, IMS, speech technologies, Smart TV, intelligent ambience, ECA.

I. Introduction

THE Internet Protocol TV (IPTV) systems have evolved from a revolution in digital broadcasting using the Internet Protocol (IP) (linear television) to a highly advanced user-centric and service-oriented interactive platforms [1]. Nowadays, IPTV system may be described as a collection of modern technologies in Information and Communication Technologies (ICT) and other domains converged to deliver a rich set of services and high-quality multimedia (TV, VOD) content over Internet protocol (IP) [2]. Therefore, IPTV systems already provide advanced, customized, and personalized services, with interactivity assumed to be the major difference from traditional media [3]. These services may be accessed and controlled by using different devices ranging from traditional TV remote controllers to advanced mobile devices (smart-phones, tablets, etc.).

Nevertheless, with additional applications being integrated into the core of IP-TV, the personalization and a natural way of control are becoming key issues. Namely, in most of the current IPTV solutions the personalization is limited to context-aware personalization through recommendation. For instance, in [4] an algorithm is proposed to recommend users preferred VOD program, available in the IPTV environment. And in [1] an advanced IPTV services personalization model is proposed for context-aware content recommendation. Here, e.g. RFID tags are used for user identification, and each user device is connected with RFID reader indicating the identity of the device, and its association with the physical location. Similarly, in [5] a context-aware based content recommendation system is represented. It provides a personalized EPG applying a client-server approach.

Further, in order to combine the technologies into IPTV system, a complex convergence network is required. The evolution of telecommunications combined with the ETSI/TISPAN [6] provides the Next Generation Network (NGN) architecture for integration of communication and interactive IPTV services into a single system. IP Multimedia Subsystem (IMS) [7] is nowadays already recognized standard for the development of IPTV platforms. Namely, IMS can be used to perform tasks related to virtualization, interoperability, subscription, billing, roaming and security etc. Therefore, deployment of IPTV system based on the IMS architecture is a compelling alternative to the proprietary commercial implementations [8]. Further, the IMS architecture allows implementation of services that have a lot of potential to greatly enhance the IPTV experience and extend its capabilities into a smart-home platform.

With the development of NGN architecture and ubiquitous

M. I. Author is with Panevropa d.o.o., Maribor 2000 Slovenia (e-mail: izidor@panevropa.com).

Z. D. Author is with the Faculty of Electrical Engineering and Computer Sicence, University of Maribor, Maribor 2000 Slovenia (e-mail: danilo.zimsek@uni-mb.si).

K. Z. Author is with the Faculty of Electrical Engineering and Computer Sicence, University of Maribor, Maribor 2000 Slovenia (e-mail: kacic@uni-mb.si).

R. M. Author is with the Faculty of Electrical Engineering and Computer Sicence, University of Maribor, Maribor 2000 Slovenia (e-mail: matej.rojc@uni-mb.si).

(pervasive) computing paradigm, the physical objects in our environment became mediums for users to interact with in a digital manner [9]. TV-set, a household device that most users are familiar with and can be found in most homes was transformed from entertainment-platform into a device having a central role, used for controlling home devices and systems. IPTVs have already been used to control intelligent ambience (e.g. smart-home, smart-room). The primary goal of these systems was to improve support and wellness of people in loss of autonomy (elderly and the disabled). For instance in [10] an adaptive and self-configurable platform for the digital home is described. The platform provides on-demand access to a broad portfolio of interactive services. HERA [11] presents an Ambient Assisted Living (AAL) system to provide specialized services for the elderly people through smart TV and set-top-box (STB) devices. In [12] a solution is represented that transforms classical TV set into a front-end of a complete home automation system. A centralized architecture based on a Home Theatre Personal Computer (HTPC) connected to the TV set is suggested. The TV set serves as the common interface to access a broad portfolio of services related to the centralized control of home appliances.

IPTV services are constantly evolving and try to bring as many ICT novelties into IPTV environment as possible. E.g. several initiatives (e.g. MediaHighway [13], OpenTV [14]) are focused to provide more personalized interactivity to the classical TV set and to develop more personalized interactive applications for set-top-boxes (STBs). The personalization and interactivity are still usually limited towards context-awareness in terms of content recommendation and multiple user-control-devices (e.g. smart-phones, tablets, game-pads, keyboards, mice, etc.). Among several technologies used in IP-TV systems and smart homes, speech technology has a lot of potential as interaction modality. Namely, voice control and voice guidance is important functionality for e.g. disabled/elderly people who have difficulties in moving and/or seeing. The tactile interfaces (e.g., remote control) require both physical and visual interaction [34]. Moreover short voice message may also be practical in distress situations. The ever-increasing complexity of systems, home appliances and services, increases the difficulties encountered by a great portion of the general population [35]. And since humans predominantly communicate with speech, IPTV interfaces should allow the user to interact with all services and devices also directly by using voice.

In this paper a novel IMS-based and multimodal IPTV platform that combines traditional IPTV services with multimodal-based non-IPTV services (text-to-speech synthesis, speech recognition, embodied conversational agents, and ambience control), named UMB-SmartTV, is presented. The architecture of the platform is developed based on IMS core [15], and distributed DATA architecture [16]. Its major benefit is that it flexibly merges IPTV services and other non-IPTV services into uniform and highly modular solution for providing entertainment, ambience control, and other useful services to users operating different devices (including TV sets, PDAs, smartphones, tablets, etc.). Furthermore, the UMB-SmartTV user's experience is more personalized and truly multimodal, without increasing STB's load on the client side. The paper is structured as follows. In section 2 the UMB-SmartTV platform is presented. In section 3 the IMS-based UMB-SmartTV architecture is described in more detail. Section 4 then presents the IMS platform, and section 5 describes the multimodal platform for UMB-SmartTV. And section 6 presents converged services, namely, the speech control mechanism, intelligent ambience control service, and text-to-speech synthesis service with ECA running on multimodal platform. The paper concludes with a discussion and an outline of our future research.

II. UMB-SMARTTV PLATFORM

The UMB-SmartTV platform merges several well established IPTV services and also services, non-native to IPTV environment, into a novel, uniform and highly modular solution, capable to provide entertainment, ambience control, and other useful services to users operating different devices (e.g. TV sets, PDAs, smartphones, tablets, etc.) connected to standard (e.g. packet-switched), or next-generation networks (e.g. 3GPP, 3GPP2).

The proposed UMB-SmartTV platform is presented in Figure 1. It is based on IMS core [15], and distributive DATA system [16]. Further, the platform consists of the following main modules: STB, content core module, IMS core module, environment controller module, and multimodal services module, based on distributive DATA system. UMB-SmartTV platform unifies all of them into a flexible and efficient multiuser and multimodal platform.

The Content core module represents an application server that takes care of content provision and content presentation (e.g. Live TV, VOD, RSS, user interfaces). The client-side services and graphical user interfaces (including IMS client) are implemented by open source media platform (Xbox Media center – XBMC) [17]. Each interface is implemented as standalone module, a fully configurable Python plugin.

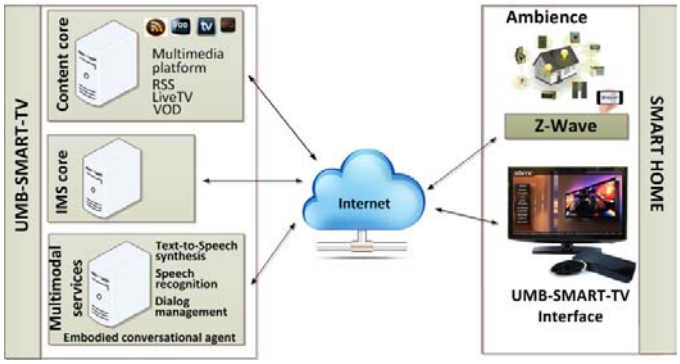

Fig. 1 the UMB-SmartTV platform

The next module is IMS core module. Namely, IMS can serve as multimedia service platform architecture. The multimedia (IP-TV) services (e.g. broadcast services and content on demand) provided by this platform can in this way be controlled and handled by IMS core's subsystems. Furthermore, the service discovery and delivery concepts are independent of underlying IP transport networks. The IMS core in UMB-SmartTV implements standard IMS functionalities, such as: user registration, subscription and management, session management, routing, triggering, interaction with NGN services (messaging, presence, profiling and grouping management), and QoS control [7].

The third module, named multimodal services, is then used for serving multimodal services. In order to separate these heavy processing tasks from the clients' STBs and in order to allow remote access, the module has been developed as a complex distributed system for providing clients with services like capturing and recognizing speech, converting general texts into speech, communicating with clients via talking embodied conversational agent (ECA), home automation services, etc. In this way the provided services emulate far more natural interaction, with far less resources and make users feel much more comfortable whilst operating the IPTV system and/or smart homes.

And the fourth module represents an environment controller that is used for controlling several devices in the environment/household (by using TV remote controller, keyboard, mobile device, or speech). Therefore, the proposed platform enables the development of a sophisticated environment combining user-centric entertainment services, user-centric home automation, and assistance services. All services available within the UMB-SmartTV can be used through interfaces implemented as plugins and operated by using standard input devices (e.g. TV remote controller, keyboard, mouse, etc.), by using smart phone, and/or speech. In the following section the UMB-SmartTV's architecture is described in detail.

III. IMS-Based Multimodal UMB-SmartTV Architecture

The proposed UMB-SmartTV architecture is outlined in Figure 2. The architecture flexibly fuses the IPTV services and the non-IPTV services, running on different underlying infrastructure. In general the IPTV services are accessible and managed through IMS architecture [15], whereas non-IPTV services are managed through multimodal platform based on distributive DATA system [16]. The STB module represents entry point for the users, able to access IMS core and associated application servers, Live-TV broadcast, VOD service, and multimodal platform. The IMS core and application services are used to host and to provide IPTV services, whereas the multimodal platform serves for the delivery of multimodal services to the users' STBs. The STB module integrates IPTV and non-IPTV services within common graphical user-interface based on XBMC [17]. And GUI components and plugins (for environment controller, YouTube, IPTV player, VIDEO player, localization and weather, etc.) are Python based. In addition STB module also implements two Python-based plugins (XBMC Web server and MM Service interface), and Java based client, named DATA client. These plugins serves for fusing non-IPTV services hosted by DATA framework [16] with XBMC environment.

A. XBMC web server

The XBMC web server [17] is proprietary built-in web server which supports several activities that can be performed via HTTP: e.g. triggering XBMC system events, calling XBMC python functions, and controlling several python plugins. The communication with XBMC environment is implemented via JSON-RPC, a TCP-IP and/or raw TCP socket-based interface that offers a more secure and robust mechanism for data exchange.

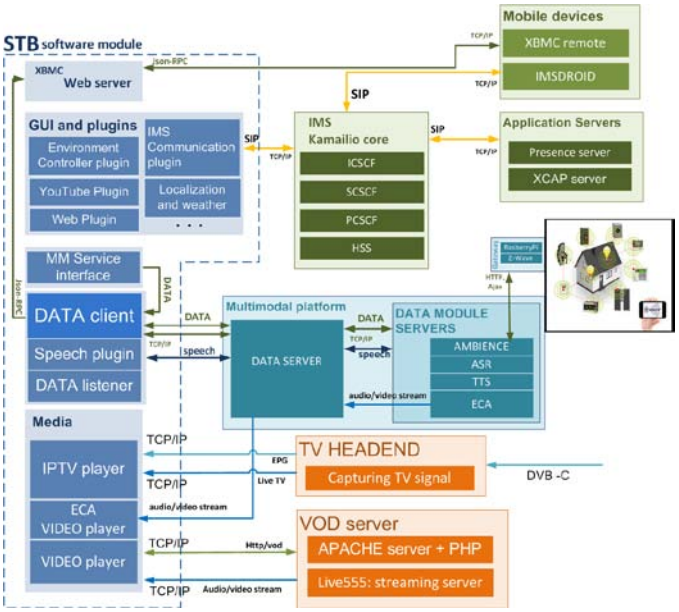

Fig. 2 UMB-SmartTV architecture

B. MM service interface and DATA client

As already mentioned, two plugins within STB are needed in order to fuse multimodal platform with the UMB-SmartTV architecture:

• MM service interface (multimodal service interface) integrated into the XBMC environment, provides the ability to communicate in the direction: STB → multimodal platform. The MM service interface serves as a low-level communication interface that enables the XBMC environment to benefit from natural modalities served by multimodal platform. Functionally, the interface behaves as a DATA module server and will, for each DATA client connected to the

multimodal platform, create a dedicated TCP/IP session. The lifecycle of the session ends, when either the DATA client disconnects from the platform, or the XBMC interface is closed. The MM service interface is implemented as a finite-state engine.

• DATA Client is a small, light-weighted Java module that provides audio to and from the STB and manages all communication between user's STB (XBMC web server and MM service interface) and distributive DATA system. The audio has to be provided to the multimodal platform for ASR, whereas the audio from the multimodal platform (generated by TTS engine) has to be played out on the STB. Namely, during interaction between users and UMB-SmartTV, the XBMC environment has to drive its GUI according to the users' requests when talking to the system (e.g. using recognized words and phrases). In this way multimodal platform and XBMC environment are linked via DATA client. As can be seen in Figure 2, DATA client communicates with XBMC web server by using JSON-RPC communication protocol, and with multimodal platform by using DATA protocol (XML based proprietary protocol).

C. TV-HEADEND and VOD

LiveTV and VOD services are supported by TV-HEADEND-based server [18], and proprietary VOD server. The first one transforms DVB-C digital signal into DVB-IP Live TV stream, and provides EPG. The VOD server is Apache-based web server for content management, fused with LIVE555 Media Server [19] that is used for content delivery. Both modules communicate with IPTV and VIDEO player plugins on the STB. For content discovery in VOD system http protocol is used, and for content delivery RTSP protocol is used for audio and video streaming.

D. MOBILE platform

Mobile platform extends the traditional control capabilities, with several off-the-shelf devices (e.g. smart-phones, tablets, game-pads, etc.). It is implemented by using XBMC remote plugin [17], and adapted IMSDROID library [20]. The platform implements several interface layouts, containing several options arranged in columns and two rows, a scrolling interface and special interfaces imitating graphical objects (e.g. buttons, icons and scroll-bars) as displayed by the XBMC GUI layouts. User interface elements are scalable and adaptable for presentation on different devices.

E. ECA EVA

STB run also ECA-EVA plugin for running talking and affective embodied conversational agent. Within UMB-SmartTV system, the ECA EVA can serve as a virtual guide or as a virtual presenter that is displayed to the user via a dedicated STB's video player.

IV. IMS Platform For UMB-SmartTV

In order to support the entertainment aspect of IPTV, e.g. delivery of multimedia services over IP, and next generation network (NGN), an IMS platform was selected and implemented. The architecture of the IMS platform is based on reference points as described in [21]. It is implemented based on Kamailio IMS [22]. The platform implements IMS Call Session Control Functions (CSCFs), and a lightweight Home Subscriber Server (HSS), which nowadays together form the core elements of all IMS/NGN architectures as specified within 3GPP, 3GPP2, ETSI TISPAN, and the PacketCable initiative. Therefore, several user devices (user equipment - UE) take advantage of IMS platform in terms of session management, service and content discovery, service and content selection, service and content delivery, service personalization, NGN service control, and access to IPTV Application Servers (AS), as proposed in [18]. Current functionalities are implemented through the following entities:

• Proxy Call Session Control Function (PCSCF): entry point for IMS users. It guarantees the delivery of signaling messages between the network and subscriber and the resource allocation for media flows.
• Serving Call Session Control Functions (SCSCF): control entity within the IMS platform. It is used for subscriber registration and authentication, location storage, call management and execution of subscriber policies.
• Interrogating Call Session Control Functions (ICSCF): determinate the SCSCF for given subscriber by querying the HSS.
• Home Subscriber Server (HSS): a centralized database that stores information on particular subscribers (including profiles, policies, subscriptions, and preferences). The entity is involved in tasks related to user authentication, authorization, personalization, billing and session management.

The Presence application server is implemented based on *Presence Module* that implements SIP/SIMPLE protocol [22]. SIMPLE is a set of extensions to SIP (RFC 3261) developed by the SIMPLE working group by the IETF, specifically to support instant messaging and presence [28]. The *Presence module* implements the core functionality of the SIP event notification. It handles PUBLISH and SUBSCRIBE messages, and generates NOTIFY messages in general and in event independent way. For presence rules communication between user agents and XCAP server, XCAP protocol is used (RFC 4825).
As seen in Figure 2, the UMB-SmartTV architecture uses SIP protocol for session management, session control, and communication over NGN (next-generation-networks). In order to communicate with the IPTV services over IMS infrastructure, a Python interface (with accompanying user-interfaces) named IMS communication plugin, was developed. The plugin has been implemented based on PJSIP library [23].

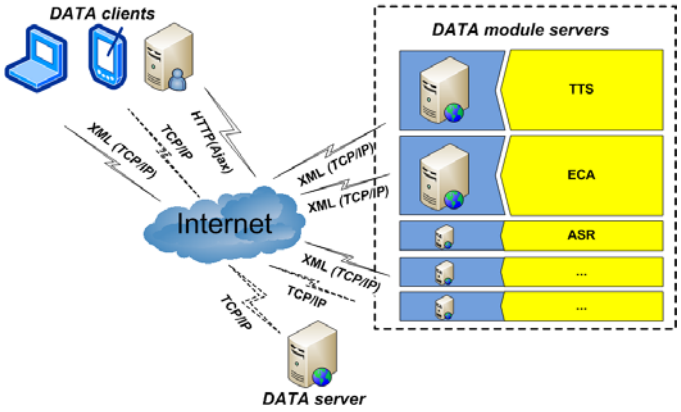

Fig. 3 architecture of the DATA system

V. Multimodal Platform For UMB-SmartTV

In order to integrate several multimodal services (speech recognition, text-to-speech-synthesis, dialog manager, and ECA) into UMB-SmartTV platform, a multimodal platform was implemented. It is based on the distributed DATA system. DATA system (Figure 3) is a complex distributed client/server architecture composed of one main server (named DATA server), and several module servers (named DATA module servers). It can serve several light clients (named DATA clients, running on the user equipment (UE)). DATA server is used to manage all communication and control all system's modules. And it is able to handle one or several module servers for each client (for each user). Furthermore, all modules can handle several clients simultaneously.

All system's modules can run on several computers, and are locally connected via TCP/IP connections. And only the main server is accessible from the internet (by the users). All modules are implemented using a proprietary Java framework, named DATA framework [16]. It is a set of Java packages that are needed for creating and managing TCP/IP connections, the creation and management of protocols, interfacing Java with a native code (C/C++), creation, compilation, validation and management of Java based finite-state machine (FSM) engines, interfacing modules with databases, parsing XML documents, and audio/video capturing/transmission over the TCP/IP. The framework includes also other Java frameworks, Unimod framework [29][30], and JMF framework [31]. The framework has been dedicated to provide context and device independent natural modalities as services. Therefore, it has been developed as an independent subsystem, based on Java, and able to run on Linux and Windows platforms.

The integration of new modality (additional engine) into the system must be as flexible as possible. And each new engine requires additional module server. The inclusion of the new engine driven by a dedicated Java based DATA module server is simply performed via the XML based configuration file. In this way, it is unnecessary to change the code, or to develop any additional APIs in order to provide a new modality in the multimodal platform. Further, modules are multi-threaded engines and each module contains a pool of threads that is initiated with a predefined number of threads. These threads are used to serve the clients' (users's) requests. All clients' requests are redirected via the main server to the corresponding module server(s). When the client's request is accepted, the main server and involved module servers, pick-up a thread from the pool and start serving the client. The dedicated session remains active until user ends the session. When no threads are available in the pool, the main server rejects new client's requests, until some session is closed, and corresponding thread is free again.

Further, the full functionality of DATA modules (handling protocol and communication) is very flexibly and efficiently described and implemented in the form of finite-state machines (FSM) [32] that are constructed by using a UniMod framework [29][30]. This Java framework defines objects for the construction and execution of finite-state engines that uses finite-state engine descriptions in specific XML-based data-format. The first step is to draw a graph representation of the desired DATA module's functionality, by considering the protocols and specifications of the multimodal platform. Graphs are simply finite-state machines composed of states, transitions and events on transitions, which trigger graph traversals during certain task executions. Finally, graph representations are transformed into custom XML descriptions and loaded by the system's module.

The main idea of the multimodal platform is to provide the ability for IPTV systems to integrate several modalities that provide additional communication channels for users. The TTS, ASR, ECAs are those technologies that have been proven for bringing more advanced natural experience, when interfaced with computer systems. As can be seen in Figure 2, the UMB-SmartTV platform uses a server-side multimodal platform entity that consist of a DATA server and the DATA module servers for: ambience control, speech recognition engine (ASR), text-to-speech synthesis engine (TTS), dialog manager engine, and personalization (PERSONALIZATION). The communication between STBs and multimodal platform, and the wrapping process are, as proposed in [24], implemented through MM Service interface (MMSI), and DATA client plugin, which is extended by speech plugin and DATA listener. MMSI translates any IPTV GUI event into a DATA system event. The MMSI interface, therefore, only extrapolates and redirects user events to DATA Client (by using a TCP/IP listener named DATA Listener). The communication between MMSI and DATA listener is implemented by using DATA protocol (TCP/IP based protocol). And the communication between DATA Client and XBMC is implemented based on JSON-RPC protocol. In this way it is possible to directly manage, control and update several XBMC plugins and IPTV services within XMBC GUIs. The speech plugin integrated into DATA Client is used for capturing audio and for playing out synthesized speech. In this way, the UMB-SmartTV architecture allows flexible bi-directional interaction between multimodal platform and IMS based IPTV system, where IPTV and non-IPTV services are

merged into a powerful, user-centric UMB-SmartTV platform, offering more advanced multimodal digital services, and complex user-friendly mechanism for controlling those services by using not only standard remote devices, but also other modalities (e.g. speech, ECA). The multimodal platform currently runs TTS engine, ASR engine, Dialogue engine, and ECA EVA engine.

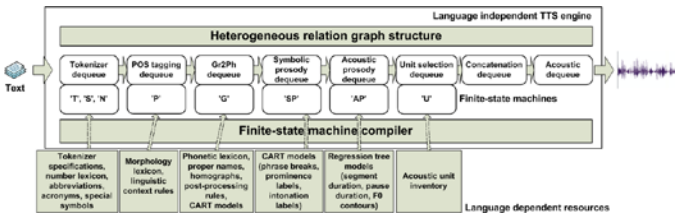

Fig. 4 TTS engine PLATTOS

A. TTS engine

Within multimodal platform, the TTS engine is implemented by fusing native TTS-engine and dedicated Java based module server. The module server serves as interface between distributed DATA system, and TTS engine, receiving text from dialogue manager, and transmitting generated speech and EVA script, for driving ECA engine. The TTS engine is corpus-based TTS system, named PLATTOS (Figure 4), using concatenative speech synthesis approach. As can be seen, the engine is language independent, since finite-state machines and CART models are used for representing all language-dependent resources [25].

B. ASR engine

Within multimodal platform, the ASR engine is implemented by fusing native ASR-engine and dedicated Java based module server. The module server serves as interface between distributed DATA system, and ASR engine, receiving acoustic features from the DATA server, and transmitting ASR output from the engine (recognized words and phrases) back to the dialogue manager. The ASR-engine is proprietary SPREAD system, supporting isolated, connected words recognition, and large vocabulary speech recognition based on perfect hash automata and tuples. At each stage of the dialogue, new grammar is loaded and used, in order to improve speech recognition accuracy and robustness.

C. Dialogue engine

Within UMB-SmartTV system all services (e.g. intelligent ambience control, XBMC interface, IPTV plugins etc.) can be driven by using speech, since multimodal platform flexibly adds new multimodal capabilities. Namely, the platform captures users' speech, recognizes words and phrases, generates audio from general texts (e.g. system messages, and other text etc.), and talking to the users via ECA-EVA. In order to manage all tasks, users' requests and system responses, the dialog manager has to be added in the form of additional DATA module server. The dialogue manager can support several users, even simultaneously, and using different dialogue scenarios. It is implemented also as Java based FSM engine, using UniMOD framework. Dialogue scenarios can be written already off-line, and loaded by dialogue manager online. The first step is to draw a graph representation of the specified dialogue scenario, by considering all supported modalities (engines), and their capabilities in the multimodal platform. Graphs are simply finite-state machines composed of states, transitions, actions on transitions and events, which should trigger dialogue graph traversals during interaction. They are also human readable, therefore, these graph representations are easily transformed into custom XML descriptions and loaded by the dialogue manager. The dialogue manager accepts ASR outputs and other events triggered within multimodal platform. It also provides TTS engine with text or just demand prerecorded audio data. The desired dialogue flow is simply specified by dialogue description. In this way, the STBs can be personalized for several users, since each user can use specific dialogue scenario, from less complex to more complex ones. Further, dialogue manager supports features like no response, or no match events. The first one is triggered when there is no response from the user, and the other when ASR engine can't determine ASR output with acceptable confidence. DATA module server running dialogue manager doesn't run any native engine, since it is completely implemented within Java environment.

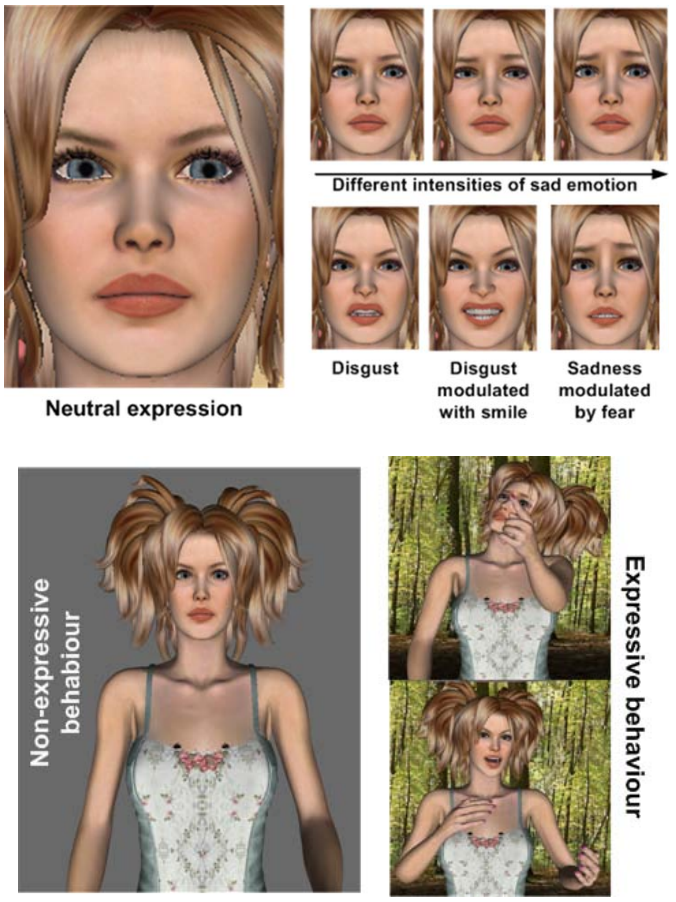

Fig. 5 ECA EVA engine's capabilities

D. ECA EVA engine

ECA EVA engine [33] animates TTS-driven conversational agent. In this way, ECA EVA presents the personification of the core PLATTOS TTS system, and is implemented by using the EVA framework. As mentioned before, the TTS engine outputs synthesized speech and EVA script, containing sequences of phonemes, visemes, and gesture descriptions. Each EVA script also includes temporal (duration), and spatial information (e.g. articulation). By using this input, the ECA EVA engine is able to visualize a PLATTOS TTS system's output in the form of animated verbal and non-verbal behavioral response. Currently, ECA EVA is a female agent. It can animate also expressive speech sequences considering different levels of co-articulation, head and hand gestures, facial expressions and emotions, and gaze. Further, the lip-sync mechanism synthesizes verbal features, and employs the articulation parameter at spoken sequence and utterance levels. To meet the general articulation properties of the sequence as a whole, all utterances are additionally modified at the spoken sequence level of articulation. And articulation at the utterance level modifies the spatial properties of the selected utterance.

Therefore, during interaction can not only adapt articulation, but also influence the speed at which a certain response has to be spoken. In addition to articulation relating verbal features, the general articulation can also define several personality features of ECA, e.g. fast-speaker, speaker with good articulation etc. When the user did not understand some parts of the spoken sequences, such sequences can be repeated at a slower rate and with a higher level of articulation. The non-verbal behavior, such as: emotion, facial expressions, head and hand movements, are generated based on linguistic and acoustic information (stored in the form of HRG structures) that the PLATTOS TTS system also provide (e.g. morphology information, phrase-break labels, prominence labels, trigger words, and phrases, stress levels, pitch etc.). By considering predicted emphasis markers and word/phrase-break markers, ECA EVA engine can generate even different speech-driven pointing gestures that can visually emphasize a certain word/phrase. By linking words/phrases with different emotions, and facial expressions, ECA EVA engine can generate TTS-driven facial expression, such as: speaking with a gentle smile, saying something sadly, etc. All these features represent very important part of the visual synthesis that is extracted from general text in the TTS engine. Figure 5 demonstrates the output of the ECA EVA engine including expressiveness and emotions that are well-supported by the EVA framework [33]. Further, different speech segments can be accompanied by different facial gestures, e.g. emphasis can be defined by a higher level of articulation, slightly lower pronunciation rate, and by raising eyebrows. The gestures performed on the right-hand side (expressive behavior) in Figure 5 are independent and don't directly influence each other. Further, the animation blending technique enables the deployment of facial expressions, emotions, and speech, simultaneously. The bone-based ECA also removes most of the "jerky", or unnatural poses that usually result when animating expressive ECAs (e.g. eyes don't follow whilst the head is turning etc.). ECA EVA engine's multipart concept uses a shared skeleton. Namely, even though the eyes and head are of different body types, the eyes will automatically be sub-parented to the joint chain of the head. This results in eyes following the head's movements. Further, emotions, gestures, gaze and verbal communication, can vary in composition (which combinations of control points are used to form them), in amplitude (to what extent a gesture forms; e.g. co-articulation of utterances), in speed, and in repetitiveness. ECA EVA engine can, therefore, generate different speech-driven types of gestures, gaze, and both simple and complex emotions, in an expressive, fully adjustable way. All the ECA's body movements are defined and described hierarchically and as a composition of movements of the control units. The ECA-EVA enables also the animation of rich sets of gestures, expressions, or event speech utterances that can vary in time, space and composition. In the following chapter running multimodal capabilities within UMB-SmartTV services will be discussed in more detail.

VI. UMB-SmartTV Services Running On Multimodal Platform

As seen in Figure 6, running multimodal service involves the following modules: XBMC GUI, MM Service interface, XBMC web server, DATA Client and several DATA module servers, running ASR, TTS, ECA EVA, and dialogue manager engines.

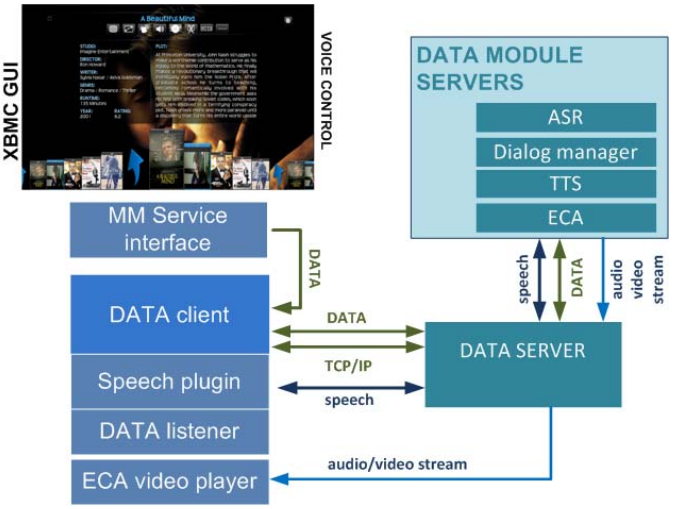

Fig. 6 running multimodal service within UMB-SmartTV system

Each IPTV service is identified either by XML content specification, or Meta content/service description. In this way, content and services clearly identify what has to be read and the context of the text being read. For instance, VOD content item clearly defines the title of the content, genre, description,

etc. Similarly, EPG clearly identifies channel description, description of current and upcoming programming, etc. When user issues a read command (e.g. read title, read description, read complete info, etc.), an event within the MM service interface plugin is triggered. Such events are then sent to the DATA Client as DATA protocol request packet via TCP/IP. The request contains context of the content, and text to be read. When intercepted by the DATA Client module, it is processed and forwarded to the multimodal platform. The DATA module server running TTS engine PLATTOS accepts it, performs text-to-speech synthesis, and sends audio to the ECA EVA engine. After generating animated video sequence, it is streamed back to the DATA client, where the video is played out to the user. Currently, PLATTOS TTS system is developed for Slovenian language. Nevertheless, it is highly modular, time and space-efficient, and flexible. Further, by following the multilingual aspect, the language-dependent resources are separated from the language-independent core TTS engine. Therefore, it can be prepared for other languages, when language resources are available. The ECA-EVA engine provides visualized speech output for the XBMC GUI's and other IPTV services (Figure 7).

Fig. 7 XBMC plugin and ECA EVA

Further, one DATA module server is responsible for running proprietary ASR engine, named SPREAD. Currently, it is developed for Slovenian language, but can be prepared also for other language, when resources are available. For UMB-SmartTV is supported recognition of isolated words and short phrases. In this way users can use also longer instructions when interacting with the system. In order to guarantee high recognition accuracy, at each dialogue level, smaller grammars are used. In this way, although the ASR vocabulary contains several hundred of words, there is no noticeable degradation in the performance of the system regarding speed or accuracy. Additionally, confidence measure is supported. Therefore, UMB-SmartTV system and multimodal platform can respond also with no match event, and then it is expected from the user that repeats his requests. ASR vocabulary can easily be extended with additional words or word phrases.

After running XBMC GUI, the interface establishes TCP/IP connection with multimodal platform. It is kept until the user ends the dedicated STB session. Meanwhile the DATA server continuously captures audio, performs feature extraction, and feeds features to the ASR DATA module server. ASR engine then responds to dialogue manager, by sending its output. Dialog manager traverses to the next dialogue state and responds by performing actions specified on the specific FSM transition. In this way, the UMB-SmartTV system is completely supported by multimodal platform and its engines without any additional load on the users' side. And users are able to interact with the system by using speech or any other traditional controllers. Further, XMBC GUI personalized by ECA EVA, enables much better and more sociable interactions with the user, than in the case of traditional IPTV systems.

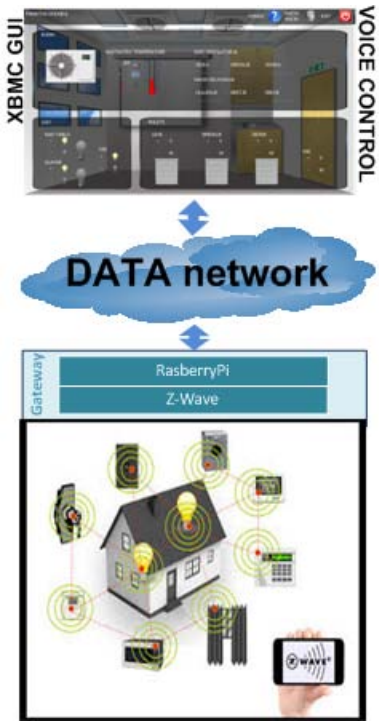

Fig. 8 UMB-SmartTV intelligent ambience control running on multimodal platform

Further, the intelligent ambience control service enables performing the home automation and the control of several household devices by interacting with UMB-SmartTV. Without multimodal platform this was possible only by using traditional controllers. Regarding general complexity of intelligent ambience and smart homes, consisting of lots of devices and possible profiles, this can be too complex for many users. Figure 8 outlines the implementation of this non-IPTV service, named "Intelligent ambience control", supported by proposed multimodal platform. It is important that the XBMC plugin for this service can remain the same. And can be used in the same way as before. Therefore, household devices, such as lights, window blinds, air-conditioning, and other smart appliances may be operated by

TV set, using a classical TV remote, dedicated user interface running on the mobile device, and now also speech. In this case the following modules in the UMB-SmartTV architecture are involved: XBMC GUI plugin (Environment controller plugin), XBMC web server, Python plugin (MM service interface), DATA Client, and several DATA module servers (and additional one for AMBIENCE control). The Z-Wave household devices establish Z-Wave network [26]. The Z-Wave alliance has standardized device classes and operation protocols for devices to be compliant with Z-Wave network. And Z-Wave family already provides a vast variety of devices that can control every aspect of a smart home: from home entertainment, to motors for window blinds, home security systems, and environmental sensors. The middleware for this service is implemented through a gateway, Raspberry Pi [27]. The middleware plays the role of a uniform abstraction layer between the application layer (AMBIENCE DATA module server) and the heterogeneous and platform-specific hardware layer. The gateway is running Raspberry Pi operating system [27] and Z-Way protocol stack. This gateway connects TCP/IP and Z-Wave protocol by using Z-Wave transceiver board. In this way all devices in the Z-Wave network can be flexibly controlled directly from the UMB-SmartTV system. Namely, each user-action within the Environment controller plugin triggers an event in the MM Service Interface plugin. For instance, pressing on a grey light bulb or just saying "switch on the light" will trigger a light-on event. The event is then sent to the DATA Client in form of DATA request (over TCP/IP). When intercepted by the DATA Client, it is forwarded to the multimodal platform. The AMBIENCE DATA module server accepts it, processes it and transmits it to the gateway in form of Ajax HTTP requests. The post-processing of the AMBIENCE DATA module server can implement different context-aware personalization mechanisms (e.g. neuro-fuzzy services, context ontology) hosted by PERSONALIZATION DATA module server, when enabled in the multimodal platform. At the end, the Ajax request is processed and executed on the gateway by RaZberry solution.

VII. Conclusion

This paper presented a novel IMS based IPTV platform for flexible integration of multimodal technologies. The proposed platform is capable of providing standard IPTV services with more advanced interaction capabilities, using also speech and audio-visual output. Further, UMB-SmartTV platform combines traditional IPTV services with speech-based non-IPTV services (automatic speech synthesis, speech recognition ambience control) in very flexible way. The architecture is developed on IMS core and distributed DATA architecture, merging services into a novel, uniform and highly modular solution for providing entertainment, ambience control, and other useful services to users operating different devices (including TV sets, PDAs, smartphones, tablets, etc.) or speech. The evaluation of the UMB-smartTV shows that user interaction with the system is much more natural an offers a better user experience. Namely, users are able to communicate with the system, by using not only mobile devices, but also speech. And responses from the system also incorporate multimodal output (ECA EVA), and not just text or synthesized system messages. Due to the fact that users are able to bypass more and more complex menus, generally used in IPTV systems, the interaction with the system is more efficient, faster and more user-friendly.

References

[1] Songbo Son, Moustafa H., and Afifi H. 2012. Advanced IPTV Services Personalization Through Context-Aware Content Recommendation. IEEE Transactions on Multimedia, vol.14, no.6, pp.1528-1537.

[2] Zeadally S., Moustafa H., and Siddiqui F. 2011. Internet Protocol Television (IPTV): Architecture, Trends, and Challenges. IEEE Systems Journal, vol.5, no.4, pp.518-527.

[3] Kim J., and Ki Hoon Lee. 2013. Towards a theoretical framework of motivations and interactivity for using IPTV. Journal of Business Research, vol. 66, no. 2, pp. 260-264.

[4] Park K., Choi J., and Lee D. 2010. IPTV-VOD program recommendation system using single-scaled hybrid filtering. In Proc. of 10th WSEAS international conference on Signal processing, computational geometry and artificial vision (ISCGAV'10), pp. 128-133.

[5] da Silva F. S., Alves L. G. P., and Bressan G. 2010. PersonalTVware: A Proposal of Architecture to Support the Context-aware Personalized Recommendation of TV Programs. In Proc. of 7th Eur. Conf. Interactive TV and Video, pp. 39-42.

[6] ETSI TS 182 028. 2008. Telecommunications and Internet Converged Services and Protocols for Advanced Networking (TISPAN). NGN Integrated IPTV Subsystem Architecture.

[7] Arnaud J., Négru D., Sidibé M., Pauty J., and Koumaras H. 2011. Adaptive IPTV services based on a novel IP Multimedia Subsystem. Multimedia Tools and Applications, vol. 55, no. 2, pp. 333-352.

[8] Shakir M. 2010. Evolving to IMS as the convergence platform. In Proc. of the 10th WSEAS international conference on applied informatics and communications, and 3rd WSEAS international conference on Biomedical electronics and biomedical informatics (AIC'10/BEBI'10), pp. 329-332.

[9] Stavropoulos T. G., Gottis K., Vrakas D., Vlahavas I. 2013. aWESoME: A web service middleware for ambient intelligence. Expert Systems with Applications, vol. 40, no. 11, pp. 4380-4392.

[10] Valladares S. M., Fernández-Iglesias M. J., Rivas C., Gómez M., and Anido L. E. 2013. An Adaptive System for the Smart Home. Recent Advances in Electrical and Computer Engineering, pp. 128-123.

[11] Spanoudakis N., Grabner B., Kotsiopoulou C., Lymperopoulou O., Moser-Siegmeth V., Pantelopoulos S., Sakka P., and Moraitis P. 2010. A novel architecture and process for Ambient Assisted Living - the HERA approach. In Prof. of 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB), pp. 1-4.

[12] Rivas-Costa C., Gómez-Carballa M., Anido-Rifón L., Fernandez-Iglesias M. J., and Valladares-Rodríguez S. 2013. Controlling your Home from your TV. Recent Advances in Electrical and Computer Engineering, pp. 124-130.

[13] MediaHighway. 2013. http://www.nds.com/Software_Solutions/MediaHighway_STB_Software/, last visited in July 2013.

[14] OpenTV. 2013. http://community.opentv.com/, last visited in July 2013.

[15] Cuevas A., Moreno J.I., Vidales P., and Einsiedler H. 2006. The IMS service platform: a solution for next-generation network operators to be more than bit pipes. IEEE Communications Magazine, vol.44, no.8, pp. 75-81.

[16] Rojc M., and Mlakar I. 2009. Finite-state machine based distributed framework DATA for intelligent ambience systems. In Proc. of CIMMACS '09, pp. 80-85.

[17] XBMC, Open Source Home Theatre Software. 2013. http://xbmc.org/, last visited in July 2013.
[18] Tvheadend. 2013. https://tvheadend.org/, last visited in July 2013.
[19] LIVE555 Media Server. 2013. http://www.live555.com/mediaServer/, last visited in July 2013.
[20] IMSDROID. 2013. http://code.google.com/p/imsdroid/, last visited in July 2013.
[21] Maisonneuve J., Deschanel M., Heiles J., Wei Li, Hong Liu, Sharpe R., and Yiyan Wu. 2009. An Overview of IPTV Standards Development. IEEE Transactions on Broadcasting, vol.55, no.2, pp.315-328.
[22] Kamailio (OpenSER). 2013. http://www.kamailio.org/dokuwiki/doku.php/presence:presence-module,last visited in July 2013.
[23] PJSIP library. 2013. http://www.pjsip.org/, last visited in July 2013.
[24] Mlakar I., and Rojc M. 2013. A new distributed platform for client-side fusion of web applications and natural modalities: MWP platform. Appl. artif. intell., vol. 27, no.7, pp. 551-574.
[25] Rojc M., and Mlakar I. 2011. Multilingual and Multimodal Corpus Based Text-to-Speech System - PLATTOS-. Speech Technologies, Book 2, Chapter 7, ISBN: 978-953-307-322-4, 2011.
[26] Z-Wave. 2013. http://www.z-wave.com/modules/AboutZ-Wave/, last visited in July 2013.
[27] Razberry. 2013. http://razberry.zwave.me/, last visited in July 2013.
[28] SIP for Instant Messaging and Presence Leveraging Extensions (simple) http://datatracker.ietf.org/wg/simple/charter/, last visited in July 2013.
[29] Shalyto, A. A. 2001. Logic Control and "Reactive" Systems: Algorithmization and Programming. Automation and Remote Control, vol. 62, no. 1, pp. 1-29. Translated from Avtomatika i Telemekhanika, no. 1, pp. 3-39.
[30] Weyns D., Boucke N., Holvoet T., Demarsin B. 2007. DynCNET: A protocol for flexible transport assignment in AGV transportation systems. Katholieke Universiteit Leuven, Report CW 478.
[31] Terrazas A., Ostuni J., Barlow M. 2002. Java Media APIs: Cross-Platform Imaging, Media and Visualization, Sams publishing.
[32] Mohri, M. 1996. On Some Applications of Finite-State Automata Theory to Natural Language Processing. Natural Language Engineering, vol. 2, no. 1, pp. 61-80, DOI: 10.1017/S135132499600126X.
[33] Mlakar I., Rojc, M. 2011. EVA: expressive multipart virtual agent performing gestures and emotions. International journal of mathematics and computers in simulation, vol. 5, no. 1, pp. 36-44.
[34] Hamill, M., Young V., J. Boger, and Mihailidis A. 2009. Development of an automated speech recognition interface for personal emergency response systems. Journal of NeuroEngineering and Re-habilitation, vol. 6, no. 26.
[35] Giannakopoulos, T., Tatlas N. A., Ganchev, T., and Potamitis, I. 2005. A practical, real-time speech-driven home automation front-end. IEEE Transactions on Consumer Electronics, , vol. 51, no. 2, pp. 514-523.