AURORA, a framework enabling multimodal interactions

Florent Chuffart
France Telecom
42 rue des coutures
14000 Caen
France
(+33) 2 3175 95 73
florent.chuffart@francetelecom.com

Filip Van Gool
Intesi Group Belgium
Drie Eikenstraat 661
2650 Edegem
Belgium
(+32) 3 826 93 81
fvangool@intesigroup.com

Lionel Courval
France Telecom
42 rue des coutures
14000 Caen
France
(+33) 2 3175 90 31
lionel.courval@francetelecom.com

ABSTRACT

The goal of this paper is to present the Multimodal Engine Architecture used in the AURORA project.

AURORA is a project proposed following the ITEA (Information Technology for European Advancement) programme's 6th Call for Projects (EUREKA Cluster). This research project aims to deal with multimodality, messaging service, multimodal authentication, presence management and VoIP architecture.

The AURORA consortium joins together 7 partners (Intesi, XandMail, EADS Defence and Security Systems, BULL, Philips CE, France Telecom and Telefónica Móviles España) from 4 countries (Netherlands, Belgium, France and Spain). The project will end in June 2006.

France Telecom is in charge of the Multimodal Engine for the project. A modular approach has been adopted in designing a Multimodal Engine (XML Multimodal Platform: (XMP) and Maestro (a HTTP server)). The XMP is organized around a VoiceXML browser, ASR and TTS resources and front-end resources.. VoiceXML Browser uses web services to access external data or services. ASR and TTS resources are used to play the service. Front-end resources are used to interact with user.

The services are hosted on Maestro. Externalizing services confers to the XMP a high level of interoperability with database, servlets and web services. Maestro ensures the orchestration of these web services and provides VoiceXML service pages to XMP.

EADS provides a secured SIP network access in order to connect SIP user devices to the multimodal messaging service and to ensure call flow.

Philips designs a gesture recognition device and a gesture user interface. This interface enables interactions between user and XMP. The interactions are based on interpretation of user device motion.

Intesi, XandMail and Bull are respectively responsible for multimodal authentication, unified data storage & communication tools, and presence & availability management technological components. Maestro manages information exchanges between these components using web services.

A UMTS network is provided by Telefónica Móviles España.

This paper describes the main components of the platform and their interconnection within the global architecture.

In the second part of the paper, a use case illustrates how the AURORA platform would enable multimodal interactions. This use case deals with the display of a slide show using multimodal authentication for slide show access, voice and gesture interactions for navigation in the slide show.

The use case illustrates the interactions between all components of the platform and is, as such, representative of the global architecture.

Categories and Subject Descriptors

D.2.11 [**Software**]: Software Engineering – *Software Architecture Domain Specific architecture, Language, Patterns*.

General Terms

Design, Experimentation, Human Factors, Standardization.

Keywords

Multimodality, dispatched architecture, VoIP, ToIP, SIP, H323, RTC, IVR, TTS, ASR, VoiceXML, Web Service, Presence Management, Biometric Authentication, Messaging service.

1 INTRODUCTION

Current means of interactions with applications are almost exclusively the keyboard and the mouse (or emulations thereof). In contrast, the principle of **multimodality** is to allow a user to command an application using either vocal or manual actions (keyboard, mouse) through the same Internet channel of a device (PDA, PC…).

The AURORA project aims to develop a software platform extending the user interface in order to allow multiple modes of interactions, offering users the choice of using their voice (headset, phone …) or an input device such as a keypad, keyboard or other input device. For output, users will be able to listen on audio devices, and to view information on graphical displays (such as a TV screen or by using a projector): this concept is called "distributed modality".

2 GLOSSARY

ASR: Automatic Speech Recognition
DTMF: Dual Tone Multi-Frequency
EMMA: Extensible MultiModal Annotation markup language
GRS: Gesture Recognition System
HTTP: HyperText Transfer Protocol
IETF: Internet Engineering Task Force
IVR: Interactive Voice Response
MMI: MultiModal Interaction [1]
MRCP: Media Resource Control Protocol [5]
RTC: Réseau Téléphonique Commuté (legacy network phone)
SIP: Session Initiate Protocol
ToIP: Telephony over IP
TTS: Text To Speech
URI: Uniform Resource Identifier [8][9]
VoiceXML: Voice eXtensible Markup Language [3]
VoIP: Voice over IP
XML: eXtensible Markup Language [2]
XMP: XML Multimodal Platform

3 GLOBAL ARCHITECTURE

The figure 1 depicts the global architecture of the project. This architecture is based on the MMI framework [1]. The input modes are gesture (GRS), voice and DTMF over SIP (through XMP front-end component).

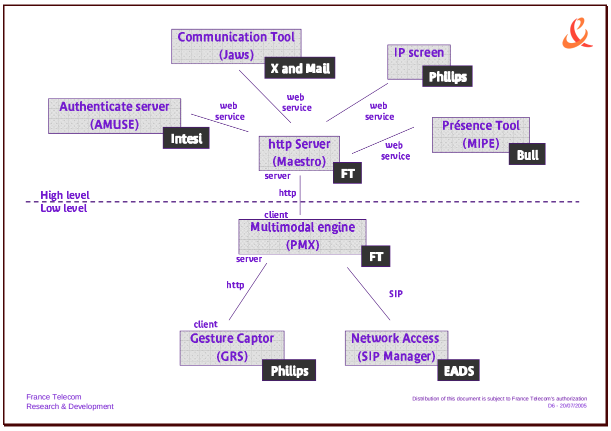


Fig. 1: aurora global architecture

The output devices are IP screen (provided by Philip's), web interface output (provided by the XandMail JAWS platform) and voice output (front-end of XMP component). Maestro ensures the application back-end. The session is dispatched between Maestro and XMP components. The system and environment components are provided by web services such as presence & availability manager MIPE. The XMP component ensures integration manager rules, using SRGS grammar [10] extended to interpret gesture events.We will focus on XMP and Maestro components. We will describe in detail their functionalities and their interactions with the other components.

4 DETAIL COMPONENTS

4.1 XMP (XML Multimodal Platform)

4.1.1 Description

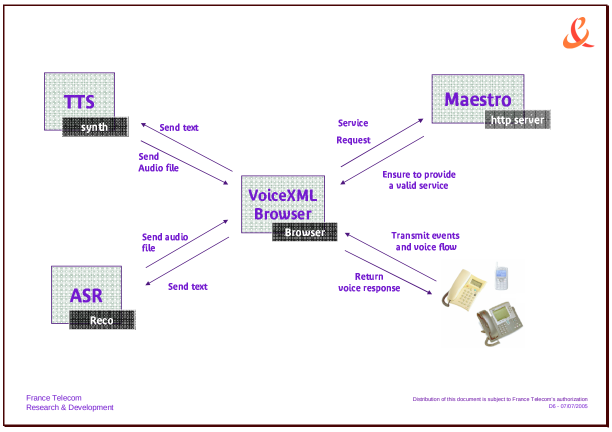


Fig. 2: XMP architecture

The XMP is a multimodal service browser. A multimodal service is an service with which the user can interact using multiple modes. For example a service using voice and gesture to command graphical application is a multimodal service. XMP is a multimodal service browser meaning that the XMP platform is able to load a multimodal service script and to provide a user interface to interact with this service. The purpose of the XMP is:

- to ensure the coherence of the different signals received from the different modes,
- to pilot a vocal service via http request,
- to execute a service process provided by "high level modules" (figure 1)of the project (Presence, Communication, Authentication coordinated by Maestro) ,
- to transmit information to the user using vocal mode and web interactions.

The XMP browser uses many web resources (web services, data base access, etc.) as described in the figure 1. The user interacts with the services through vocal interface and/or gesture interface. XMP provides recognition and vocalization functionalities based on remote TTS and ASR resources. XMP browser services provided by Maestro (HTTP or HTTPS server). The Maestro module will be further described in detail.

4.1.2 Back-end dispatching control

This part defines the internal XMP architecture. The figure 2 describes XMP internal architecture.

4.1.2.1 Network access

The funtionalities of the telephony front-end module are:

- to hang up when someone calls the XMP
- to transmit voice flow and DTMF coming from the user device to the browser
- to transmit voice fluxes coming from service to user device
- to ensure telephonic features (transfer …).

4.1.2.2 Gesture interface

The GRS module is a user interface able to capture user motions, interpret them and transmit some result to XMP. The motion capture module uses a gyroscope.

The GRS module allows the user to incline his mobile device to the right, to the left, to the front and to the back. Each action is associated with an event that the XMP is able to interpret using a VoiceXML tag extension.

In order to transit only desired motions, GRS module uses "push to transmit" technologies. It means that, to transmit information using motions, the user has to use a specific command.

4.1.2.3 VoiceXML browser

The VoiceXML [3] browser module is the service interpreter: it loads the VoiceXML service and is able to play the service to the user. This module ensures the interactions between the user and the service. Gesture, DTMF and vocal events are interpreted as instructions. The ASR module enables these interpretations and therefore the navigation in the service. The audio files are generated by the TTS module. These generated audio files are sent to the user via the multimodal front-end.

The VoiceXML tags have been extended in order to interpret gesture events. The extension concerns *grammar* tag *mode* attribute. Voice XML 2.0 SRGS provides support for the use of *DTMF* or *voice* for *grammar* tag *mode* attribute. We extended this VoiceXML 2.0 tag attribute with *gesture* . A *gesture* mode grammar item is one of *left*, *right*, *back* and *front* as describe below.

```
<grammar mode="gesture">
 <rule id="incline">
  <one-of>
   <item> left </item>
   <item> right </item>
   <item> back </item>
   <item> front </item>
  </one-of>
 </rule>
</grammar>
```

4.1.2.4 TTS and ASR (Text To Speech, Automatic Speech Recognition)

The two modules: TTS and ASR are responsible for respectively text to speech and speech to text conversion.

The VoiceXML browser supports a wide range of ASR and TTS solutions. XMP can provide a service with support for different languages (English, French, etc.) depending on the ASR used,, with support for a variety of ASR types (natural language based ASR, large vocabulary ASR, isolated word ASR, connected words ASR or more specific ASR).

The protocol used to communicate between VoiceXML browser and ASR/TTS is MRCP [5].

4.1.2.5 Maestro

This module is described in more details in the next section. Maestro is a VoiceXML page provider for the VoiceXML browser. Within the AURORA project, PHP is used to generate VoiceXML pages, and to interact with the web services

4.2 Maestro

4.2.1 Description

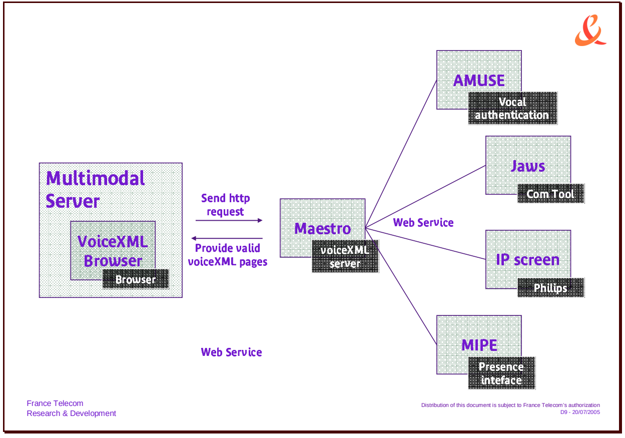


Fig. 3: Component interactions with Maestro.

The Maestro module:

- is the main controller that interacts with the different backend components such as the multimodal authentication component, the presence and availability component and the communication tools component. All theses interactions are done through web services.
- ensures the connectivity between web services, database resources, multimodal plateform and user interface,
- provides VoiceXML pages to the XMP VoiceXML browser,

PHP is used to generate VoiceXML pages at runtime. The VoiceXML interpreter is modified in order to enable multimodal services.

Maestro is a VoiceXML page server linked over IP with AMUSE (multimodal authentication server), JAWS (communication tools component), MIPE (presence and availability module) and the Philips IP screen.

4.2.2 Connectivity with other components

<u>AMUSE</u>

Amuse is a multimodal authentication server. It enables authentication using different modalities (e.g. biometric voice authentication, pin authentication, PKI authentication). It has been designed as a PAM easily extensible with other authentication modalities and could be used to adapt authentication modalities based on device capabilities and security constraints. Within the AURORA platform, the MAESTRO uses the AMUSE server for authentication that is required for session initiation.

JAWS

JAWS is a data storage platform. It interacts with Maestro using web services though IMAP4. Its aim is to store user personal data. This platform is able to convert any SMSs, MMSs, emails with attachments or any instant messaging transferred files into an internal XML formalism. These data are available on the network and the access is protected by an authentication procedure.

IP screen

IP screen is a user interface able to display any multimedia content (e.g. picture, video, audio…). This module provides support for multimedia rendering display. The module within the project is mainly used for the presentation and rendering of mail attachments and more generally graphical data.

MIPE

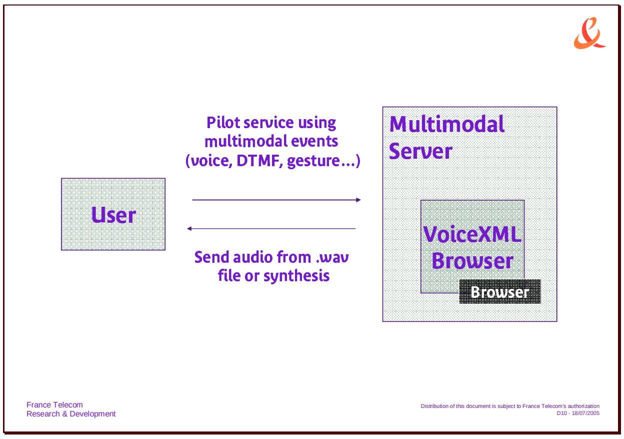
MIPE is a module for presence and availability management. It provides an interface to many presence servers. The core functionality of MIPE is to provide detailed presence information related to the user on different devices. The MAESTRO communicates with the MIPE component through web services.

4.2.3 Connectivity between AMUSE and MIPE

Note that AMUSE is able to determinate the mode where the user is able to be authenticated. MIPE provide this functionality.

5 PLATFORM USE CASES

In the AURORA demonstrator, we place an user in a slide show presentation context. As describe in the figure 4, the user interacts with the system using web interface, voice, DTMF and gesture. The system returns graphical data on the IP screen.


Pilot service using multimodal events (voice, DTMF, gesture…)

User

Multimodal Server

VoiceXML Browser

Browser

Send audio from .wav file or synthesis

France Telecom
Research & Development

Distribution of this document is subject to France Telecom's authorization
D10 - 18/07/2005

Fig. 4: User interactions with AURORA platform.

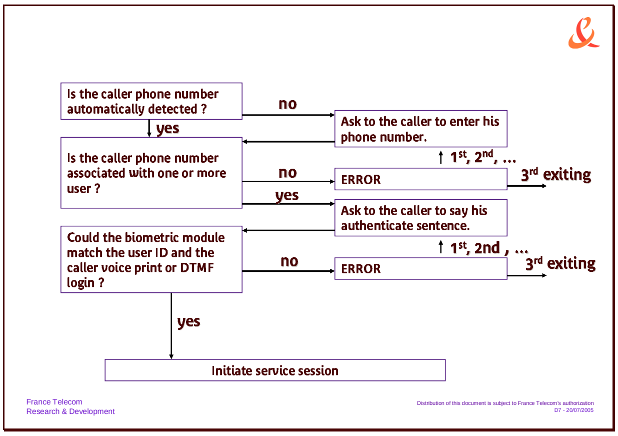
For AURORA demonstrator, the following scenario will be implemented:

- step 1; the user initiates a session with the AURORA platform using his PDA and biometric voice authentication,
- step 2; the user chooses to use the "slide show" vocal service using voice or DTMF,
- step 3; the user asks to display the right slide show on the IP screen,
- step 4; the user interacts with the slide show using multimodal events.

5.1 The user is authenticated using biometry

First of all, the user initiates a session with the AURORA platform using his phone and the biometric voice authentication. The login (his phone number) is automatically detected by the platform. When identified, the user is authenticated using a biometric pass phrase (i.e. his/her voice print).

MAESTRO is in charge of the multimodal authentication web service invocation. If the authentication is successful, the MAESTRO initiates a session.


Is the caller phone number automatically detected ?

no

Ask to the caller to enter his phone number.

yes

Is the caller phone number associated with one or more user ?

1st, 2nd, …

no

ERROR

3rd exiting

yes

Ask to the caller to say his authenticate sentence.

Could the biometric module match the user ID and the caller voice print or DTMF login ?

1st, 2nd , …

no

ERROR

3rd exiting

yes

Initiate service session

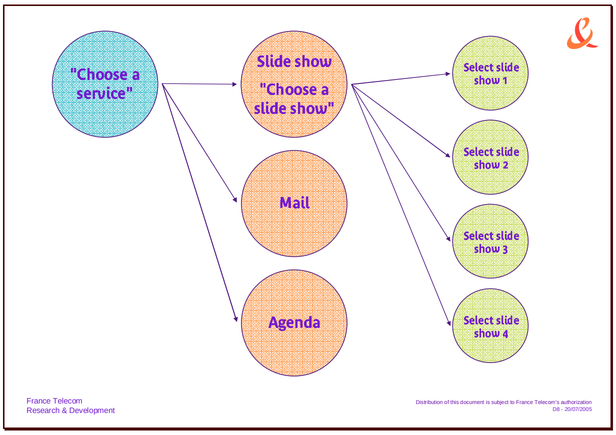
France Telecom
Research & Development

Distribution of this document is subject to France Telecom's authorization
D7 - 20/07/2005

Fig. 5: Biometric authentication procedure.

5.2 The user chooses to use the "slide show service"

The user is now authenticated. Automatically, the platform proposes the available services (service 1: mail, service 2: agenda, service 3: slide show…).


"Choose a service"

Slide show "Choose a slide show"

Mail

Agenda

Select slide show 1

Select slide show 2

Select slide show 3

Select slide show 4

France Telecom
Research & Development

Distribution of this document is subject to France Telecom's authorization
D8 - 20/07/2005

Fig. 6: Slide show service access procedure.

The user chooses the slide show using voice – he says: "*slide show*" – or if using DTMF – he presses the DTMF '3'.

Here, the service enumerates the available slide shows. The user chooses the right slide show using voice or DTMF as the same the previous step.

When the right slide show is selected, the platform pushes this slide show to the IP screen.

5.3 The platform displays the right slide show on IP screen

The system has a logical representation of a slide show (figure 7). It is represented like a list of URI with a first element and an access to the next slide.

When the user has found the right slide show, the first element is displayed on the IP screen. The system pushes to the IP screen the first slide URI. The IP screen loads the pointed slide, renders the data and displays the slide.

Now, the user can interact with the IP screen using multimodal events.

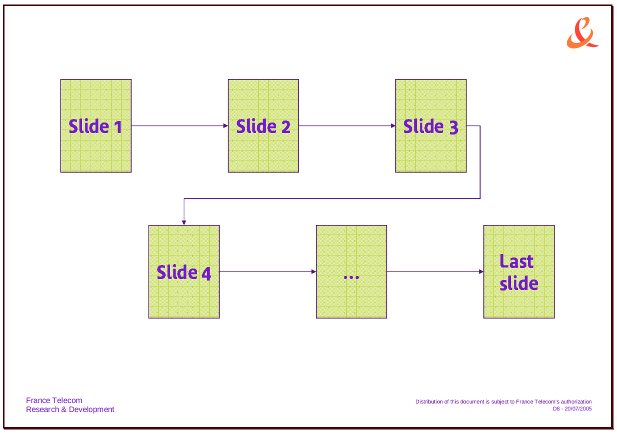


Fig. 7: Logical slide show representation.

5.4 The user interacts with the slide show

The user is able to interact with the slide show:

- to navigate to the next, previous or to a specific slide,
- to zoom in/out on a slide,
- and to move the zoomed slide focus.

"Zoom in" and zoom out" functionalities are provide by IP screen. Maestro accesses to these functionalities by web services.

The user can navigate in the slide show using multimodal events. The user can interact with DTMF, gesture or voice.

Voice. The user can interact with the slide show using voice, he has to say: "next", "previous", "go to the slide *i*", "zoom in", "zoom out", "right", 'left", "up" and "down" to carry out the action. The Voice interactions use the "push to transmit" technology.

DTMF. A DTMF is associated to an action. This association is adjustable. We choose a default configuration that uses three control pads on the DTMF keyboard:

- First pads; the */# DTMF to control next and previous slide access,
- Second pads; 5 and 0 DTMF to control zoom in/out actions.
- Last control panel; 2, 6, 8, 4 DTMF to move the focus when "zoom in" is active.

Gesture. In our case, the gesture allows four actions. It means we have to choose the associated actions:

- First choice; the four action are associated with navigation and "zoom in/out" functionalities,
- Second choice; the four action are associated with navigation and "zoom in/out" functionalities

Gesture interactions use the "push to transmit" technology.

Table: event/action association table

Actions	Events		
	Voice + push to transmit	**DTMF**	**Gesture + push to transmit**
Go to the next slide	"next"	#	Incline to the right*
Go to the previons slide	"previous	*	Incline to the left*
Zoom in	"zoom in"	5	Incline to the front*
Zoom out	'zoom out"	0	Incline to the back*
Move to the right	"right"	6	Incline to the right**
Move to the left	"left"	4	Incline to the left**
Move to the top	"up"	2	Incline to the front**
Move to the bottom	"down"	8	Incline to the back**

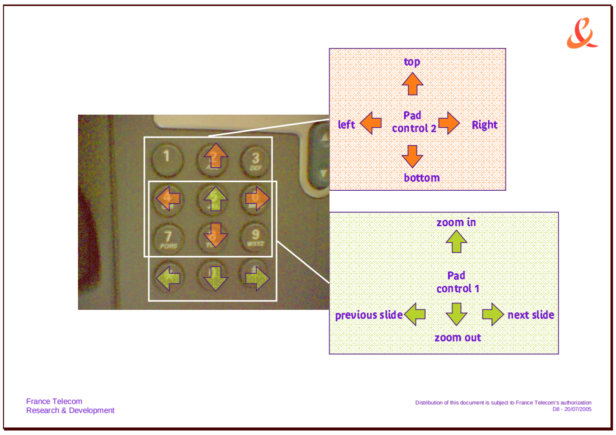


Fig. 8: DTMF default configuration.

6 ACKNOWLEDGMENTS

Our thanks to ICMI05 organization, Alexandro Vinciarelli (IDIAP Research Institute), AURORA Consortium, LATEMS, and Jacques Madelaine (Laboratoire GREYC – CNRS UMR 6072).

7 REFERENCES

[1] W3C. 2003. *W3C Multimodal Interaction Framework* http://www.w3.org/TR/mmi-framework/

[2] W3C. 2004. *eXtensible Markup Language (XML) 1.0 (third Edition)*. http://www.w3.org/TR/REC-xml/

[3] W3C. 2004. *Voice Extensible Markup Language (VoiceXML) Version 2.0*. http://www.w3.org/TR/voicexml20/

[4] W3C. 2004. EMMA: *Extensible MultiModal Annotation markup language* http://www.w3.org/TR/emma/

[5] IETF. 2005. A Media Resource Control Protocol Developed by Cisco, Nuance, and Speechworks. http://www.ietf.org/internet-drafts/draft-shanmugham-mrcp-07.txt

[6] W3C. 2005. *State Chart XML (SCXML): State Machine Notation for Control Abstraction 1.0.* http://www.w3.org/TR/scxml/

[7]Rouillard José. 2004 *VoiceXML, Le langage d'accès à Internet par telephone.* Ed. Vuibert

[8] T. Berners-Lee and al. 1994. *RFC1630 Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web.*

[9] T. Berners-Lee and al. 1998. *RFC2396 Uniform Resource Identifiers (URI): Generic Syntax.*

[10]W3C. 2004. *Speech Recognition Grammar Specification Version 1.0.* http://www.w3.org/TR/speech-grammar/