AURORA, multimodal messaging framework for ubiquitous web context.

Platform architecture and use case.

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 unified multimodal messaging using multimodal authentication for mail access, voice and gesture interactions for navigation in the mailbox.

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, PSTN, 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".

According to Marc Weizer [11], ubiquitous computing represents a powerful shift in computation, where people live, work, and play in a seamlessly interweaving computing environment. Ubiquitous computing postulates a world where people are surrounded by computing devices and a computing infrastructure that supports us in everything we do[12]. AURORA framework is a way to realize this ubiquitous web.

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]
PSTN: Public Switch Telephony Network
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).

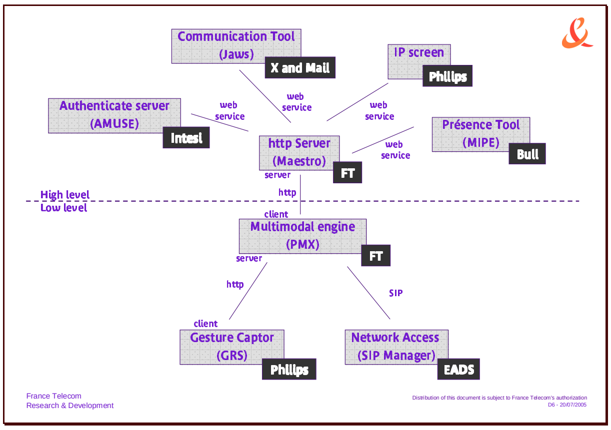

Fig. 1: aurora global architecture

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

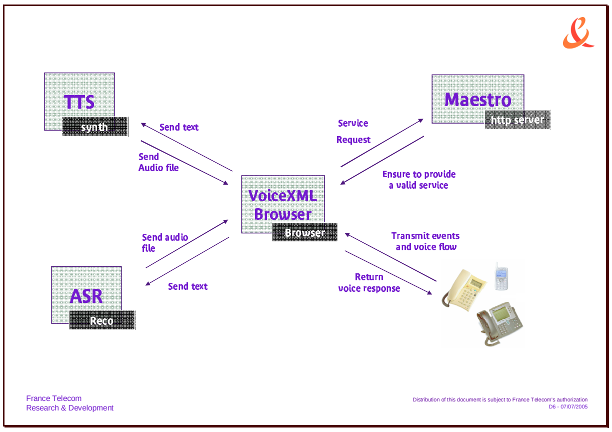

Fig. 2: XMP architecture

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

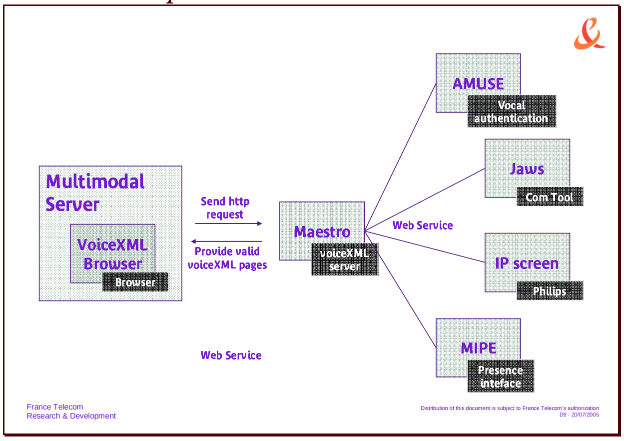

Fig. 3: Component interactions with Maestro.

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*

AMUSE

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

As describe in the figure 4, the user interacts with the system using voice, DTMF and gesture. The system returns audio signal.

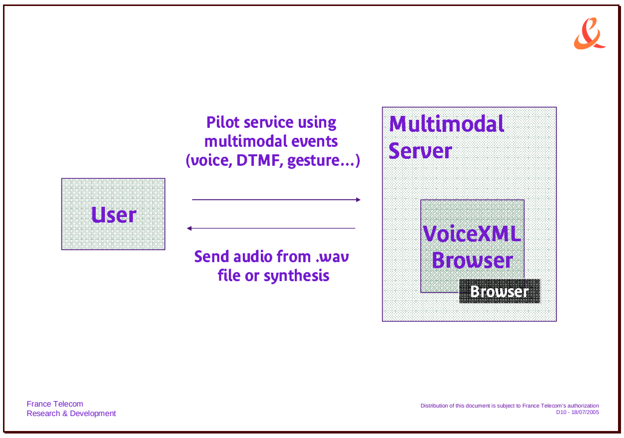

Fig. 4: User interaction with AURORA platform.

For the AURORA demonstrator the following scenario will be implemented:

- step 1: the user initiates a session with the AURORA platform using his phone and biometric voice authentication.
- step 2: the user listens to his mails and navigates trough the application using multimodal interactions (such as gestures, speech commands, etc.)..
- step 3: the user chooses to display an email attachment on an IP device (see above).
- step 4: The user wants to contact the sender of the email.

Most of the figures that represent multimodal services use state machine representation. This abstraction seems [6] to be appropriated to represent the control on this kind of service.

5.1 The user is authenticated using biometry

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

If the system cannot initiate session on the two first attempts (e.g.: the user is in a noisy environment disabling the vocal authentication), a different the authentication method is used for the authentication without compromising the required security level. If the authentication is successful, the MAESTRO initiates a session.

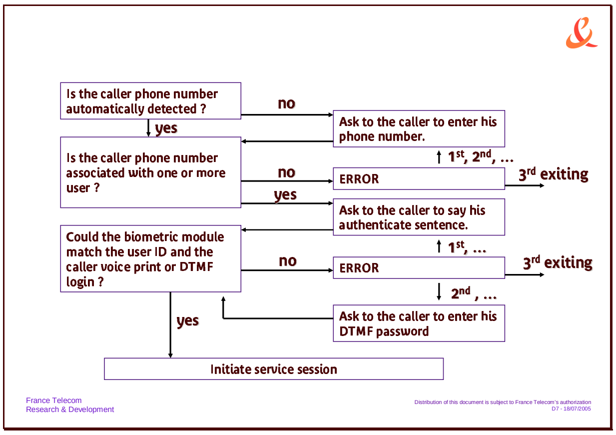

Fig. 5: Biometric authentication procedure.

5.2 The user consults his mails using his mobile phone

The user is now authenticated. Automatically, the platform

The service presents him the state of his mail account. A list of his mail is presented sorted by different possible criteria (e.g. new , date, from, etc)te. Typically, the system prompts the user: "*You have two unread messages and one archived message. First message: ...* ".

The service displays the user messages starting with the first element of the presented list. The user listens to their messages beginning with the most recent messages. TTS component is used to convert the text message to audio output.

For each element, user operations allowed are:

- Go to the next (expect for the last element)
- Go to the previous (expect for the first element)
- Erase the current
- Push attachment on IP device
- Contact the sender

The user is able to terminated or interrupt the service at any moment.

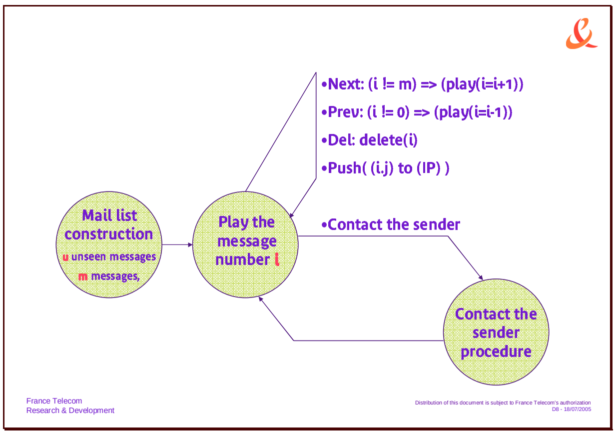

Fig. 6: Mail consultation procedure.

5.3 The user displays an attachment on an IP screen

When a message contains an attachment, the user can display this attachment on an IP screen. For example, the user asks the system to "*display first attachment to the kitchen screen*". The system knows the URI "*first attachment*" and the IP of the "*kitchen screen*". The system resolves the "*push(it, here)*" function as described in figure 7.

- The user ask to the system : "display first attachment to the kitchen screen "
- "first attachment" → "http://.../myAttachment.png"
- "kitchen screen" → "192.168.1.10"
- ScreenContainer = http://.../myAttachment.png
- Refresh the display of the IP screen.

Fig. 7: Display attachment procedure.

5.4 The user wants to contact the sender.

When the user has finished listening to a message, the user can ask the system to contact the sender of this message. The Maestro consults the MIPE module to obtain presence and availability of the sender (the MIPE module can easily be extended to incorporate user preferences and privacy protection).. The MIPE module provides information to the MAESTRO module. This on its turn replies to the user one of the following answers:

- The sender is "on line"
- The sender is busy

If the sender is "on line", the platform establishes a connection between the user and the sender of the message, using the VoiceXML tag "transfer". If the sender is status is “busy”, the platform offers the functionality to leave a voice mail to the sender of the message. This voice mail is embedded as an attachment in the message in the mail box of the sender.

This choice is describe in the figure 8.

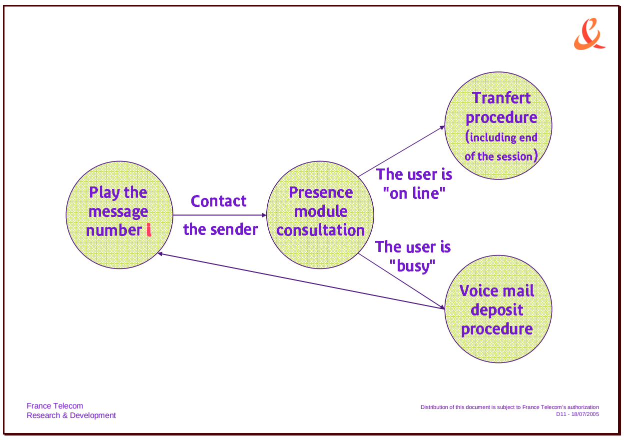

Fig. 8: Contact user procedure.

6 ACKNOWLEDGMENTS

Our thanks to Philipp Hoschka, Dave Raggett, Marie-Claire Forgue, AURORA Consortium, LATEMS (common laboratory between ENSICAEN - Ecole Nationale Supérieure d'Ingénieurs de Caen - and FRANCE TELECOM R&D) 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/

[11] Marc Weiser. Sept 1991. *The Computer of the 21st Century, Scientific America.*

[12] Dave Raggett, *june 2005, W3C, Multimodal Web Applications for Embedded Systems.*