

Web services and speech-based applications around VoiceXML

José Rouillard

Laboratoire LIFL (Trigone) - CUEEP - Bat B6
 Université des Sciences et Technologies de Lille
 59655 Villeneuve d'Ascq Cedex - France
 Email: jose.rouillard@univ-lille1.fr

Abstract—VoiceXML applications are described by context-free grammars. Then, recognized vocabulary is limited. We show in this paper one possible approach in order to use VoiceXML applications and speech web services, together. The idea is to use capabilities of speech-dictation systems when the input value is not recognized in an active grammar. A prototype of VoiceXML application using speech web services is presented. A user can speak a free sentence, in English for instance, and receive a French translation, on the same modality (phone) or via another one (PC screen, for example).

Index Terms—Web services, Speech interfaces, VoiceXML, Context free speech

I. INTRODUCTION

Telephones are much more numerous than computers on the planet. That assertion can explain why vocal technologies and interfaces are an important part of Human-Computer Interaction (HCI) area. Using natural language within interaction is supposed to facilitate exchanges between humans and machines. That's why simple and efficient vocal interactions are awaited in many domains such as E-health, E-learning, E-trade, M-trade or E-administration. These factors pushed the World Wide Web Consortium (W3C) to work on this direction and to publish a recommendation concerning a vocal interaction language, based on XML, which allows describing and managing vocal interactions on the Internet network.

VoiceXML is becoming the standard language used for developing interactive voice response and speech-enabled applications. The key idea is to use the same networks, architectures and protocols as previous applications deployed on the Web. More precisely, VoiceXML is a programming language, designed for Human-Computer audio dialogs that feature synthesized speech, digitized audio, recognition of spoken, DTMF (Dual Tone Multi Frequency) key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of web-based development and content delivery to interactive voice response

applications [32], [33], [34]. Since a few years, VoiceXML is used to conceive and develop vocal but also multimodal solutions [1], [24].

VoiceXML 1.0 was published in March 2000, and since March 2004, a W3C recommendation of the version 2.0 is available. The changes which appear between the two versions are relatively light. In the short run, new VoiceXML 3.0 version should allow multimodal interactions [6]. Additionally, X+V language [36] (XHTML+Voice Profile 1.0) is available since December 2001 and is designed for Web clients that support visual and spoken interaction at the same time. Users can choose to enter information through a traditional Web browser (Opera for instance), or to speak to the computer, via a microphone connected to the PC.

But VoiceXML applications are described by context-free grammars. Thus, recognized vocabulary is limited. We will see in this paper the potential power to couple VoiceXML applications and context free grammar speech-based web services. The idea is to use capabilities of speech-dictation systems, seamlessly connected to VoiceXML applications using web services.

The paper is organized as follows. Section 2 describes VoiceXML platform features. Section 3 illustrates the proposed solution. Section 4 discusses this solution.

II. VOICEXML PLATFORM FEATURES

A VoiceXML platform is the foundation for developing and operating voice automation applications. During the Human-Computer interaction, it executes the commands and logic specified by applications written in VoiceXML. It also provides the speech processing capabilities (speech recognition, speech synthesis, voice authentication...).

As mentioned in Fig. 1, VoiceXML platform architecture is based on HTTP protocol, and uses both phone and Internet networks. The Web server is often connected to a database which the user can question and update.

A zoom on the VoiceXML gateway shows that this computer is equipped with a telephone card, able to manage incoming and outgoing calls, a VoiceXML interpreter charged to carry out all the orders programmed in this language, and a connection with Internet network.

Based on "Web services and speech-based applications", by José Rouillard, which appeared in the Proceedings of the IEEE International Conference on Pervasive Services (ICPS'06), 26-29 June 2006, Lyon, France © 2006 IEEE.

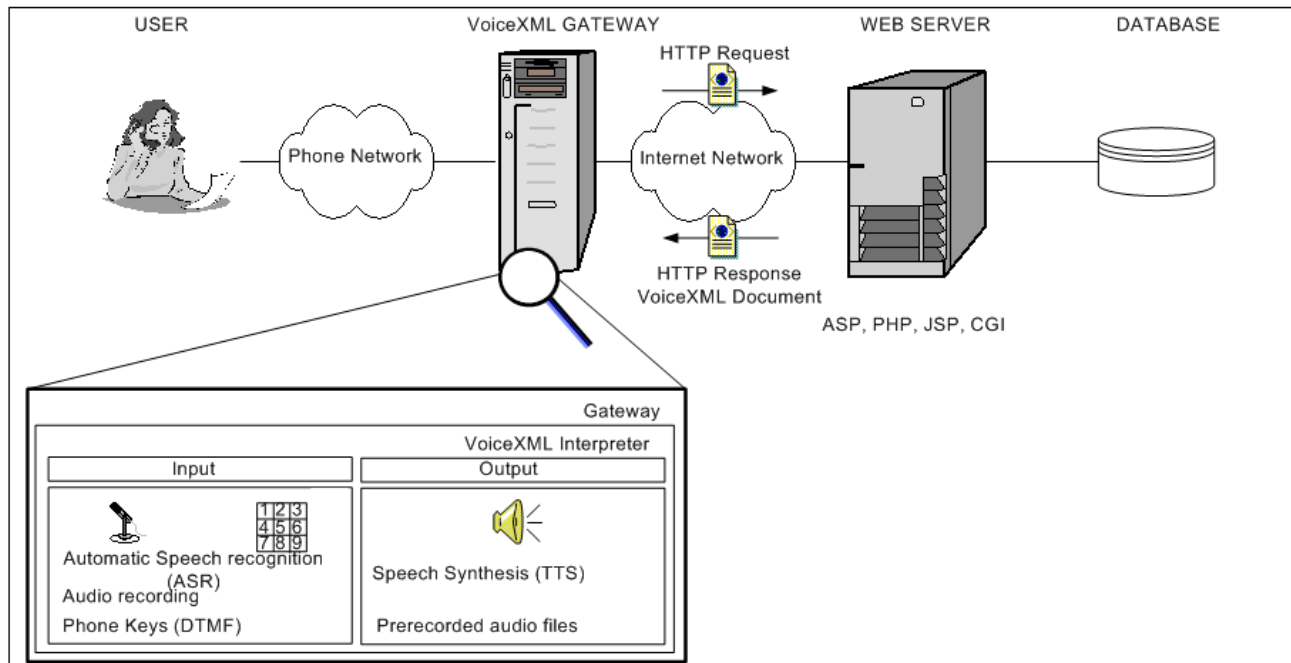


Figure 1. VoiceXML platform architecture

Automatic Speech Recognition (ASR) engine, recording of audio files and Dual Tone Multi Frequency (DTMF) telephone keyboard are the inputs users interact with. Text To Speech (TTS) voice synthesis and restitution of pre-recorded audio files are the outputs the machine can use.

Some VoiceXML features are based on the observation of Human-Human behaviour. The *bargein* attribute, for example, specifies whether or not the caller will be able to interrupt the TTS/audio output with a DTMF keypress or voice utterance. A value of true indicates that the user is allowed to interrupt, while a value of false forces the caller to listen to the entire prompt before being allowed to give input to the application.

A. VoiceXML grammar

In VoiceXML, recognition grammars directly affect the language model used in vocal applications. It is possible to decide if only the system will lead the conversation (system initiative dialogs) or if the user will have the opportunity to anticipate some information (mixed initiative dialogs). This is illustrated by the well known weather forecast vocal application example, in which a user could have the possibility to say "Los Angeles, California", instead of waiting for the two questions: "which city?" and "Which state?".

In a mixed initiative dialog, users have some flexibility in choosing the sequence of interactions. Technically, it means that it is possible to have more than one input field active. It's a choice among concurrently active recognition grammars. But, in a case as in the other, the system must perfectly know what the user is supposed to say. Indeed, a VoiceXML grammar identifies different words or phrases that a user might say: the recognized

vocabulary and syntax are decided *ab initio* by the programmer.

A vocal grammar is, in a certain way, the core of a VoiceXML application, since it determines what the user connected to the server is able to say, or not. Grammars can range from a simple list of possible words to a complex set of sentences. Fig. 2 presents a VoiceXML version 1.0 code example, in which an external grammar is invoked (see Fig. 3).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="1.0">
<form>
  <field name="choice">
    <prompt>Tell me your name,
please.</prompt>
<grammar type="application/x-jsgf"
src="users_names.gram"/>
    <filled>Your name is <value
expr="choice"/>.
      <clear/>
    </filled>
  </field>
</form>
</vxml>
```

Figure 2. A VoiceXML code example

A vocal grammar could be given in the body of the VoiceXML script (inline grammar), or in separate file (external grammar). Different kind of grammar could be used such as Nuance GSL [21] or Java Speech Grammar Format [11]. Figure 3 shows the external vocal grammar used in our example. It uses a BNF (Backus Naur Form¹)

¹ John Backus and Peter Naur introduced for the first time a formal notation to describe the syntax of a given language.

notation [19]. The two firsts rules are private, while the last one is public. The sign "|" symbolizes an alternative (a logical "or"). The sign "*", named Kleene star, indicates that the element before this symbol can appear zero or several times. With the square brackets it is possible to declare an element as optional. The brackets define a regrouping. Finally, the sign "+" indicates that element before this symbol can appear one or several times.

```
#JSGF V1.0;
grammar the_names;
<connector> = and|or|but not;
<names>=john|paul|georges|ringo;
public <sentence> = (<names>
[<connector>*])+;
```

Figure 3: A Java Speech Grammar Format (JSGF) code example

We can see on Figure 4 that user can only answer (to the question pronounced by the machine "Tell me your name, please") a sentence following the defined vocal grammar.

```
Computer: Tell me your name, please.
Human: ringo
C: Your name is ringo.
C: Tell me your name, please.
H: georges and ringo
C: Your name is georges and ringo.
C: Tell me your name, please.
H: john but not paul
C: Your name is john but not paul.
C: Tell me your name, please.
H: paul but not georges paul
C: Your name is paul but not georges paul.
C: Tell me your name, please.
H: ringo and georges but not paul and john
C: Your name is ringo and georges but not paul and john.
```

Figure 4: Example of human-machine dialogue obtained with a VoiceXML application

Here, the important thing to understand is that user can only pronounce a sentence predefined in the VoiceXML application's grammar. In our example, it means that it will be impossible for the computer to understand "Peter" or "Karen", for instance, because those utterances are not available in this grammar. It will be exactly the same thing with a VoiceXML application where the user is supposed to give the name of a shape, a color or an action. To understand a sentence like "move the red triangle", this verb, qualifier and noun must be defined, by the programmer, in grammar rules.

B. Related works

The W3C MultiModal Interaction Working Group (MMIWG) works on standards that will allow users to

interact with applications with a combination of input modalities that include speech, pen, keyboard, and pointing, as well as future input modalities [19].

The MMIWG is developing a XML specification called Extensible MultiModal Annotation (EMMA) for representing the intentions reflected in a user's input, in any modality, along with other information about the circumstances of the utterance [9]. In VoiceXML, the <noinput> tag allows the developer to assign event handlers when the application expects voice or DTMF input, and received nothing from the caller. The <nomatch> tag is useful when the caller inputs a value that is not recognized by any of the active grammars.

In the same way, with EMMA's specifications, the notation for uninterpretable input can refer to any possible stage of interpretation processing engine, including raw transcriptions. For instance, if input speech cannot be correctly recognized or the spoken input is not matched by a grammar, it can be tagged as emma:uninterpreted.

More generally, in voice applications, the performances of voice synthesis can appear disappointing when the system must face ambiguities, when it meets unknown words or proper names. The voice recognition employed is supposed multi-speaker, without enrolment phase, and able to be executed in noisy environment. Vocal grammars are inevitably limited: the speaker will not be able to say all he/she wants, but only what have been planned in design phase. This last point is a significant limit that needs particular efforts.

Some related works were driven by Casey Chesnut, in early 2003, with the FreeSpeech Project [8]: the challenge was to speak freely with a vocal application, running on a Pocket PC, thanks to a speech web service connected to SAPI [27] Voice Recognition engine. In the EvalVoiceXML project (Evaluation of Intelligent Component Technologies for VoiceXML Applications), Mittendorf and other authors proposed to explore possible interfaces between VoiceXML and NLU (Natural Language Understanding) modules.

The first idea was to use a trivial grammar that recognizes an arbitrary string of words, disregarding any grammatical or semantic constraints, and to pass the resulting string of words on to a NLP (Natural Language Processing) system.

The second approach was to use raw speech data. Our works are following this suggestion: "As an extreme approach, the VoiceXML speech recognition mechanisms could be bypassed completely by recording a user utterance and passing the unprocessed audio data to an NLU with its own speech recognition capabilities." [16].

Since the first version of VoiceXML, the concept of transcription from audio to written text was evoked. Indeed, the standard proposed in appendix a <transcribe> element to add into the next version of VoiceXML. However, none of the successive versions of the VoiceXML language support this tag. Moreover, <transcribe> element is not implemented by any manufacturer, and will not be available either in the future version of VoiceXML 3.0.

III. POSSIBLE SOLUTIONS

A possible solution to this issue consists in employing a large vocabulary voice recognition system (such as Nuance - Dragon Naturally Speaking [7] or Telispeech from Telisma [29]), independent of that used by the VoiceXML platform, which allows the transcription of text not awaited by the grammar used in the vocal application. This speech system is used on demand by the VoiceXML Server. An advantage of this method is that all limitations of VoiceXML (e.g. handling of unknown words) are resolved.

A. Scenarios

In this section we give some propositions that could bring interesting solutions.

a) A user connects to a vocal server with its (wired or mobile) phone. The computer suggests recording a free speech. Therefore, the user speaks a free sentence, in English for instance, and receives a French translation, on the same mode (its phone) or via another mode (a screen for example). This is normally unconceivable, in classical VoiceXML, because the platform's speech recognition engine can't do prediction;

b) Some users call a vocal server and record free vocal messages. An electronic white board, located in a "public" place of this organisation (cafeteria, meeting room, etc.) is in charged to display the textual messages to everybody, with the name of the person, date, time, priority, etc. It could be, for example "*Martin, 3:05 pm: Do not forget the lab meeting tomorrow, 10 o'clock, room 206*"; "*Tina, 9:17 pm: A new book concerning VoiceXML is available*" ...

c) A user pronounces a word or a sentence in the phone, and relevant links are obtained thanks to a web service provided by a search engine like Google for example [12]. According to the user's choices, results can be sent by mail, fax, SMS, or synthesised by phone;

d) A person is reading a book while she is waiting for her train. She encounters an unknown word and wants to get a definition. A first web service gives a plain text (i.e. the transcription) when it receives a recorded speech. A second web service provides a complete dictionary definition of a word. The synthesized (TTS) definition is given to the user. She will also receive the complete definition in her mailbox. Figure 5 shows a sample example of E-mail obtained by a user that pronounced the word "medicine".

```

medicine
  n 1: the branches of medical science
that deal with nonsurgical
      techniques [syn: {medical
specialty}]
  2: (medicine) something that treats or
prevents or alleviates
      the symptoms of disease [syn:
{medication}, {medicament},
      {medicinal drug}]
  3: the learned profession that is
mastered by graduate training
      in a medical school and that is
devoted to preventing or
      alleviating or curing diseases and
injuries; "he studied
      medicine at Harvard" [syn:
{practice of medicine}]
  4: punishment for one's actions; "you
have to face the music";
      "take your medicine" [syn: {music}]
  v : treat medicinally, treat with
medicine [syn: {medicate}]

```

Figure 5: Sample of E-mail containing the definition of the word "medicine", that the user pronounced freely

All those scenarios need more than a traditional VoiceXML platform to be implemented. Indeed, they are mainly based on a speech-to-text translation feature that is not available for the moment in basic VoiceXML specifications.

B. Architecture

The decomposition of the proposed architecture (see Fig. 6) can be described as the following:

- (1) A VoiceXML application, with no defined grammar, activated by the user, asks to record a free sentence. This vocal recording is converted into an audio file;
- (2) It is transmitted to a traditional server via HTTP;
- (3) This audio file is then treated by an independent speech recognition tool;
- (4) The obtained transcription is transmitted to the vocal server;
- (5) The VoiceXML application can use the textual version of the user input.

C. Technical feasibility

In order to prove the technical feasibility of our project, we have implemented different version of the presented scenarios (synchronous and asynchronous). As we will see on Fig. 8, the chosen architecture is based on the notion of service. We used SOAP [28] to supply speech web services.

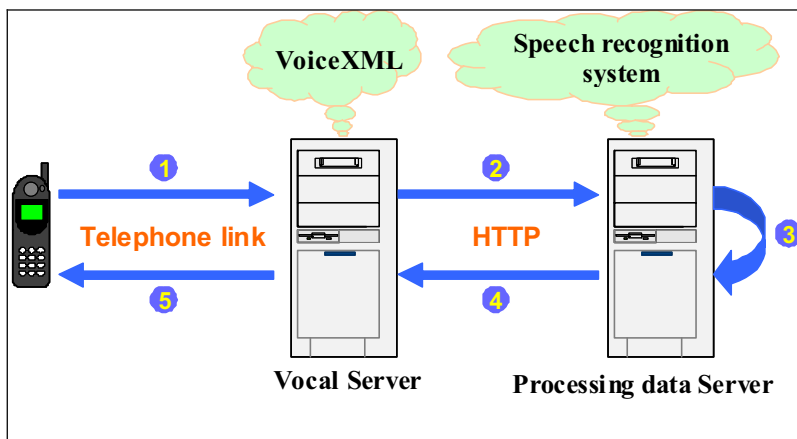


Figure 6. Global architecture

A Service-Oriented Architecture (SOA) is essentially a collection of services [20]. Some ways of connecting services to each other is needed, because a service is seen as an action carried out by a component "supplier" to the attention of a "consuming" component, possibly based on another system [13].

The detailed architecture given on Fig. 8 shows that a vocal server managed in VoiceXML can be beneficially coupled with web services, in order to transmit and obtain useful information. Our contribution lies mainly in a particular web service, able to transcribe free speech into plain text. Technically, this could be done by different means: we present briefly the two principal possibilities chosen: by URL and by DIME attachment. In our project, the first solution was considered as an asynchronous solution while the other one was seen as a synchronous one.

Asynchronous solution

It is possible to provide to our web service an audio file, previously recorded. The file is processed asynchronously (some seconds) and we get a written version.

We decided to set an URL as input parameter for this web service (see the top of Fig. 8). Fig. 7 gives an example of Man-Machine dialogue obtained with a vocal server that used VoiceXML coupled to a web service providing a Wave to text transcription.

```

05:19:30 C: At the tone, please record
your message.
05:19:30 C: (audio clip)
05:19:34 A: recording started
05:19:42 A: recording stopped
05:19:42 C: Please wait while I'm
analysing your message.
05:20:00 C: I think you said: Have you
got a swimming pool in this hotel
    
```

Figure 7. Result of the signal processing: in a VoiceXML application, the machine gives back the sentence that the user said, without any grammar.

It is clear that, to be relevant, the dialogue manager will have to be connected to a robust and powerful

Natural Language Understanding system, because, even if the transcription is correct, on a syntactic level, it will not necessarily be easy, on a semantic level, to really understand what the users' expectations were. This work goes beyond the boundary of this article.

Synchronous solution

In synchronous mode, we coupled a traditional VoiceXML application to our web service of transcription. The vocal application allows the recording of a free message from any telephone. This message, converted into audio file, is sent to the web service, in SOAP attachment. The obtained transcription is synthesized and presented in oral form to the user. We also tested possible sequences with other web services of translation (French/English for example) or of dictionary, which run correctly.

Instead of passing an URL to the web Service, it is also possible to use a WS-Attachments protocol, proposed by Microsoft and called DIME (Direct Internet Message Encapsulation). This mechanism allows sending directly an attachment (picture, sound, etc.) to a web services. We used C# language and WSE, within Visual Studio, for our developments. Web Services Enhancements 2.0 for Microsoft .NET (WSE) is a .NET class library for building web services using the latest Web services protocols including WS-Security, WS-SecureConversation, WS-Trust, WS-Policy, WS-SecurityPolicy, WS-Addressing, and WS-Attachments².

The way web Services interact with each other at the message level, including the execution order and the data flow, is called orchestration [23], [3]. Since a few years, an important number of web Service orchestration languages or specifications are emerging, just like BPEL (Business Process Execution Language) [2] WSCI (Web Service Choreography Interface), and BPML (Business Process Management Language) for example [4]. We worked in this project with BPEL and the Oracle BPEL Process manager [22].

² A new version of Windows Communication Foundation is available in its release version as part of the .NET Framework 3.0.

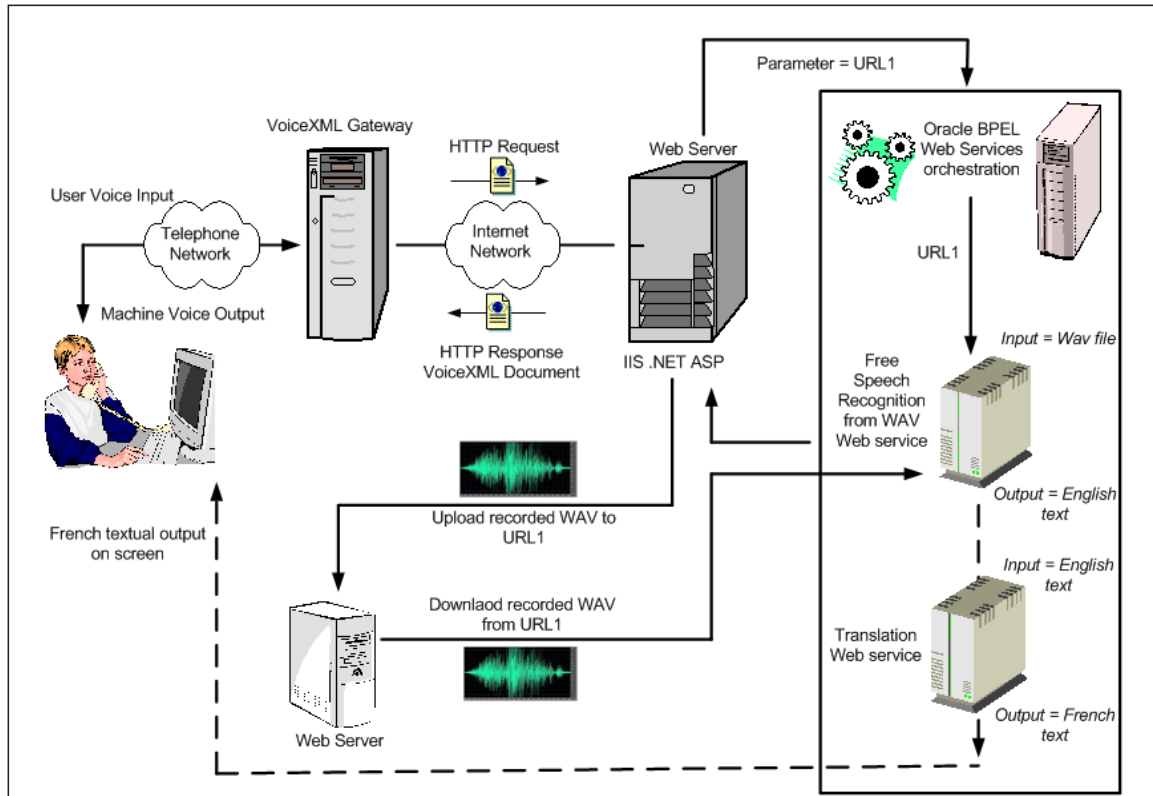


Figure 8. Detailed architecture

An example of Collaxa/Oracle BPEL design view is given on Fig. 9. It shows the web services orchestration used for the scenario number 3 in which the user wants to receive an E-mail containing the definition of the word pronounced on the phone.

Our “Wav_From_URL” web service is invoked and returns the calculated transcription. Therefore, the partner link “Word_definition” is called, with in input parameter the result of the voice recognition. And finally, a “Mailing” web service is called in order to send the definitive answer to the user.

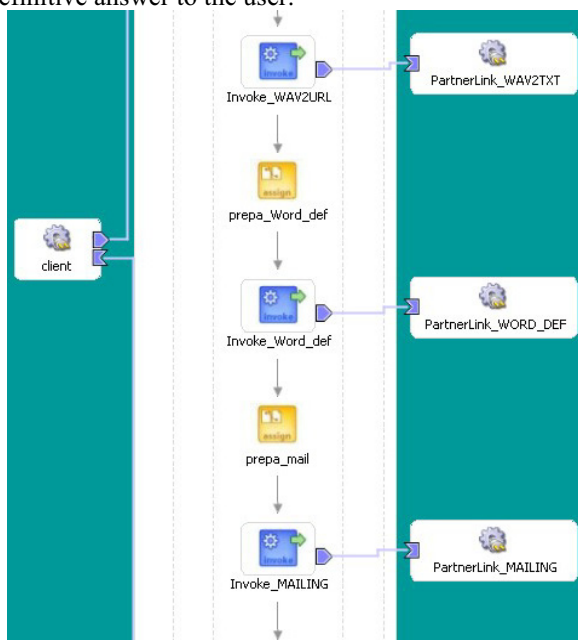


Figure 9. BPEL Oracle Designer view

Fig. 10 is a capture of the Oracle BPEL console used for this project. It provides different views and tools (audit, flow, processes, instances, activities...).

With this view, for example, we can see how the services are managed: the output of one service is the input of the following, and the pronounced sentence “*Is it possible to book a room for the next weekend please*” will be translated in French as “*Est-il possible de réserver une salle pour le week-end suivant svp*”. Here, we are not working on the robustness of the speech translation given by the partner, but we are focusing on the web service’s orchestration. Indeed, in this context, a better English-French translation should give “*chambre*” instead of “*salle*”, and “*prochain*” instead of “*suivant*”, for example. A version entirely implemented with the .NET framework was tested. It increases speed and deployment of the web services by using WSDL [35] for its description and UDDI [30] for its referencing in the directories of web services.

IV. DISCUSSION

The idea of intermixing grammar-based and free-form recognizers is intuitive and everybody working on the subject certainly meets this idea. It could be considered at the same time a classical suggestion (yet another ...) or an important step toward improvement of voice application’s usability. Our discussion with Max Froumentin, from the W3C, in the early 2005 [10] led us to the observation that, at least, two ways were possible in order to work in this direction.

Figure 10. Oracle BPEL results. A French sentence is proposed, based upon an English recorded speech

The first one was trying to improve the VoiceXML language itself while the second one was to work around VoiceXML and the notion of Web services, with the intention of giving new feature to speech-based applications, including those supporting VoiceXML, but not limited to this only language.

Thus, we made a first submission to the community [26], presenting briefly our main suggestions. James A. Larson³ was the first to give a feedback with his paper entitled « VoiceXML on Steroids » [15], where he explained “ *Researchers at the Université des Sciences et Technologies de Lille have implemented a <transcribe> tag within VoiceXML that converts speech to text using a dictation recognition engine rather than the conversational speech engines traditionally used in VoiceXML applications. The <transcribe> tag recognizes free form text without a developer-specified grammar. This tag could be very useful in several situations, for example to allow the system to express some utterances not modeled by the application, but pronounced by users and then provide feedback to users by re-using their input.*

Some others reviews were less optimistic and criticized the idea this way: if the application is going to do anything more than just store the free-form segment for human use, the problem of recognition accuracy, ambiguity, etc. need to be deal with in the overall system. In short, their argument explaining the reason why this was not already in the VoiceXML protocol is that there are too many complications to deal with and keep the protocol simple enough to be usable.

More generally, researchers and practitioners are adding new tags to VoiceXML (2.0 and 2.1). Many of these tags will find their way into VoiceXML 3.0, announced multimodal. From a technical and scientific point of view, using these new tags can result in the generation of higher quality speech applications faster and more efficiently, but this have to be balanced with the economic point of view, and the problems of portability (for example platforms that do not support new tags or functionalities).

We searched solutions close to SOA philosophy and proposed to use Web services to bring complementary features to speech-based applications [25]. According to us, this approach allows the integration, not only in applications that use VoiceXML, but also in multimodal speech-based, a possibility, for the user, to use natural language, across Web services mechanisms.

One possible limitation is related to the time needed to join remote services together. In our first evaluations, the range of elapsed time was from 3.66 to 46 seconds.

³ Dr. James A. Larson is chairman of the W3C Voice Browser Working Group that is standardizing VoiceXML and related markup languages for developing speech applications. Jim works for Intel Corporation, in which he is Manager of Advanced Human Input/Output. He also teaches courses in developing speech applications at Portland State University and the Oregon Graduate School in Oregon Health and Sciences University. His Web site is <http://www.larson-tech.com>

Here we have to face the famous usage/usability conflict, well known in Human-Computer Interfaces sciences. This particularly technical point is essential within the acceptance or not of services by end-users. The time needed to compute raw speech on distant machines and give back the N-best hypothesis solution can be considered for many people as a serious drawback. In another hand, some users seem to be ready to wait a little bit more, if the given solution allows using a large vocabulary instead of a limited grammar.

V. CONCLUSION

We explained that a strong constraint curbs the employment of VoiceXML to the use of a predetermined grammar. By coupling a traditional VoiceXML platform with an independent system of voice recognition, we showed that it is possible to increase its capacities of understanding, since a user can pronounce a sentence initially not configured, at the VoiceXML grammar level. We implemented a few number of the presented scenarios, in order to prove that the solution is technically feasible.

The solution presented in this article, based on a web service able to process a signal can be used both in synchronous and asynchronous way. As we showed, it can be integrated in a web services orchestration, with BPEL, for example. According to us, that opens many directions for multimodal human-computer interface development.

A version entirely implemented with the .NET framework was tested. It increases speed and deployment of the web services. Of course, this service will be more robust if the speech recognition software associated is more powerful.

Our future works are oriented in this direction. Commercial systems (speech recognition software) usually used [7], [31] based on Hidden Markov Models incorporate an acoustic model, a large vocabulary, and a linguistic model. One of the software version [7] used for the test phase is equipped with a vocabulary gathering more than 250000 current and specialized words. Ideally, the use of a voice recognition system, really speaker independent, like the CMU Janus system [14], should improve the robustness of the whole application.

On the top of this, the possibility to change on the fly the desired resources (ASR, most of the time) using MRCP⁴ (Media Resource Control Protocol) [17] will be a way to be more efficient. Best results, according to a context, will be chosen among different speech recognition engines, working at the same time.

ACKNOWLEDGMENT

The present research work has been supported by the French « Région Nord Pas-de-Calais » and the FEDER (Fonds Européen de Développement Régional) during the

MIAOU and EUCUE projects. The authors gratefully acknowledge the support of these institutions. The author also wishes to thank Dr Philippe Truillet, from the IRIT laboratory (Toulouse, France) which is at the origin of a part of this work for his fruitful collaboration, and Rémi Thomas for his help in improving this paper.

REFERENCES

- [1] Anderson, E. A., Breitenbach, S., Burd, T., Chidambaram, N., Houle, P., D. Newsome, D. Tang, X., Zhu, X., Early Adopter VoiceXML, Wrox, 2001.
- [2] Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S. and Trickovic, I, Business Process Execution Language for Web Services version 1.1., Technical report, BEA, IBM, Microsoft, SAP, Siebel Systems, May 2003.
- [3] IBM Web Service BPEL: <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [4] Courbis, C., Finkelstein, A., Toward aspect weaving applications, 27th International Conference on Software Engineering (ICSE'05), St. Louis, Missouri, USA, 2005.
- [5] Courbis, C., Finkelstein, A., Weaving Aspects into Web Service Orchestrations, 3rd IEEE International Conference on Web Services (ICWS 2005), Orlando, Florida, 2005.
- [6] Dettmer, R., It's good to talk, speech technology for on-line services access, IEE Review, Volume: 49, Issue:6, June 2003, pp 30-33.
- [7] Dragon NaturallySpeakingTM Preferred, <http://www.scansoft.com/naturallyspeaking>
- [8] Chesnut, C., FreeSpeech project, <http://www.mperfect.net/freespeech>
- [9] EMMA: Extensible MultiModal Annotation markup language, W3C Working Draft (16 September 2005), <http://www.w3.org/TR/emma/>
- [10] Froumetin, Max, The W3C and its Multimodal Interaction Activity, March, 2005. University of Lille, <http://www.w3.org/2005/Talks/0321-maxf-w3c>
- [11] JSGF: Java Speech Grammar Format, <http://www.w3.org/TR/jsgf>
- [12] Google API : <http://www.google.com/apis/>
- [13] Krafzig, D., Banke, K., Slama, D., Enterprise SOA: Service Oriented-Architecture Best Practices, Prentice Hall PTR, 2004, 416 pages.
- [14] Levin L. , Lavie A. , Woszczyna M. , Gates D. , Ga-valda M. , Koll D., Waibel A., The JANUS-III translation system: Speech-to-speech translation in multipledomains, Machine translation, 2000 , vol. 15, no 1-2 , pp. 3 – 25.
- [15] Larson, James A, "VoiceXML on Steroids ", speech technology magazine, November/December 2005, online version available : <http://www.larson-tech.com/Writings/Steroids.htm>
- [16] Mittendorfer, M., Winiwarer, W., Niklfeld, G., Making the VoiceWeb smarter Integrating Intelligent Component Technologies and VoiceXML, Proc. WISE 2001, Kyoto, Japan
- [17] Media Resource Control Protocol (MRCPv1 (RFC 4463)): <http://www.ietf.org/rfc/rfc4463.txt>
- [18] MMIWG, MultiModal Interaction Working Group, <http://www.w3.org/2002/mmi/>
- [19] Naur, P., Revised Report on the Algorithmic Language ALGOL 60, Communications of the ACM, Vol. 3 No.5, pp. 299-314, 1960.
- [20] Newcomer, E., Lomow, G., Service-Oriented Architecture with web services, Addison Wesley, 2005, 444 pages.
- [21] GSL: Nuance Grammar Specification Language, <http://studio.tellme.com/grammars/gsl>

⁴ MRCP is a communication protocol which allows speech servers to provide various speech services (such as speech recognition and speech synthesis) to its clients. See www.ietf.org for details.

- [22] Oracle BPEL Process manager home page: <http://www.oracle.com/technology/products/ias/bpel/index.html>
- [23] Peltz, C., Web Services Orchestration - a review of emerging technologies, tools, and standards, Technical report, HP, January 2003. Technical white paper, http://devresource.hp.com/drc/technical_white_papers/WSOrch/WSOrchestration.pdf
- [24] Rouillard, J., VoiceXML, Le langage d'accès à Internet par téléphone, Paris, Vuibert, 2004.
- [25] Rouillard, J., Web services and speech-based applications, ICPS'06, IEEE International Conference on Pervasive Services 2006 Lyon, 2006.
- [26] Rouillard, J., Truillet, P., Enhanced VoiceXML, HCI International 2005, Las Vegas, 2005.
- [27] <http://www.microsoft.com/speech/download/sdk51/>
- [28] SOAP: Simple Object Access Protocol, <http://www.w3.org/TR/soap>
- [29] Telispeech 1.2 from Telisma, <http://www.telisma.com/>
- [30] UDDI, Universal Description, Discovery and Integration protocol, <http://www.uddi.org>
- [31] ViaVoice (IBM), <http://www.scansoft.com/viavoice>
- [32] VoiceXML 1.0., W3C Recommendation, <http://www.w3.org/TR/voicexml10>
- [33] VoiceXML 2.0., W3C Recommendation, <http://www.w3.org/TR/voicexml20>
- [34] VoiceXML 2.1, W3C Candidate Recommendation <http://www.w3.org/TR/2005/CR-voicexml21-20050613/>
- [35] WSDL: Web Services Description Language, <http://www.w3.org/TR/wsdl>
- [36] X+V, XHTML + Voice Profile, <http://www.voicexml.org/specs/multimodal/x+v/12>

José Rouillard was born in Cavaillon, France, on October 23, 1970. He received his PhD degree in Computer Science from University of Grenoble (France) in 2000.

Then, he joined the Science and Technology University of Lille (USTL) as a lecturer, the same year.

Dr. Rouillard has written the only French speaking book talking about VoiceXML and his research interests include HCI plasticity (see PlasticML for example), multi-modality and multi-channel interfaces. He has written more than 45 articles and refereed conference papers and journals. He is now engaged in research on mobility and pervasive/ubiquitous computing.