

# VoiceXML

Philippe PRADOS  
philippe@prados.net



## Avant propos

*Ce document explique les fonctionnalités de VoiceXML.*

En 2005, la majorité des internautes utiliseront un autre périphérique qu'un ordinateur pour naviguer sur le réseau.

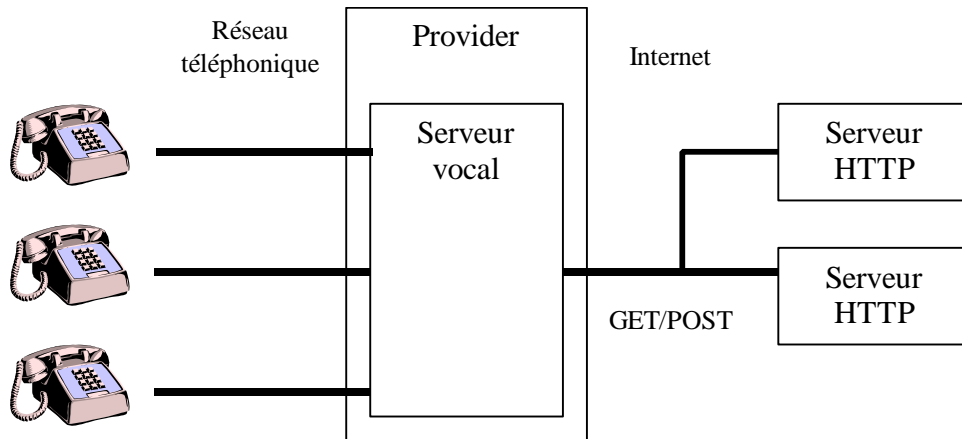
Le WAP est une technologie permettant de consulter son compte, de passer des ordres de bourse, ou de jouer avec son téléphone. Pour le développeur, l'hétérogénéité des téléphones et les bugs des implantations rend très difficile la réalisation d'une application WAP qui fonctionne. Pour le format HTML, il faut garantir une compatibilité entre Internet Explorer et Netscape Communicator, et éventuellement avec le navigateur Opéra.

Il existe pourtant un média universelle, disponible dans tous les foyers, et sans problème de compatibilité : le téléphone. Le téléphone classique avec son micro et son haut-parleur. Cela permet de toucher pratiquement toute la population. 20 à 25% de la population de l'OCDE ne savent pas suffisamment lire et écrire pour utiliser Internet. Le téléphone est d'un accès plus facile.

Une technologie permet de naviguer sur le net avec un téléphone classique. Elle s'appelle VoiceXML. Que viens faire XML, qui est un format texte, avec le téléphone qui utilise des sons ?

VoiceXML permet de décrire l'ergonomie d'un site vocale. Un serveur particulier interprète le format VoiceXML pour diffuser les messages à l'utilisateur et pour analyser ces réponses au format DTMF (les touches du téléphone) ou par reconnaissance vocale. IBM possède 95 brevet sur la reconnaissance de la voix (<http://www.research.ibm.com/hlt/html/patents.html>)

Pour naviguer sur internet, un programme analyse le format HTML, présente le document à l'utilisateur et interprète ses réactions pour remplir des formulaires. Un navigateur VoiceXML procède de même. Il demande initialement un document à un serveur http, diffuse vocalement les informations en synthèse vocale ou en jouant des fichiers pré-enregistrés ; puis, le navigateur analyse les réponses de l'utilisateur ou les touches du téléphones. Cela permet de remplir des formulaires pour demander les pages suivantes.



Comment le téléphone peut-il devenir un navigateur VoiceXML ? Il passe par un serveur intermédiaire s'occupant de faire le lien entre le téléphone et le net. (Voir schéma) L'utilisateur appelle un numéro de téléphone. Le serveur vocale est paramétré pour associer un numéro à une URL. Il interroge alors le serveur http correspondant pour obtenir un fichier au format VoiceXML. Le serveur vocale peut alors proposer une ergonomie adaptée au téléphone et invoquer le serveur http lorsque cela est nécessaire.

Ainsi, sur le serveur http, la même application peut être présente au format HTML, WAP et VoiceXML. Cela permet de s'adapter au média de l'utilisateur. Le client peut choisir le média qu'il désire pour consulter son compte en banque ou passer des ordres de bourse. A terme, l'utilisateur pourra utiliser simultanément les fonctionnalités WAP et VoiceXML. Le protocole WAP s'occupe des interactions textuelles, VoiceXML de lire ou d'enrichir ces informations. Les écrans des téléphones WAP sont insuffisant pour une explication longue. Un message vocale pourra alors commenter les informations textuelle. Inversement, un tableau est très difficile à diffuser en vocale. Le WAP pourra alors prendre le relais. Réunir les différents médias sur le même serveur d'application permettra d'unir ces technologies.

Les spécifications de VoiceXML sont disponible sur le site officiel de référence [www.voicexml.org](http://www.voicexml.org). Elles ont été soumise au W3C.

Vous pouvez télécharger un navigateur VoiceXML en Java sur le site [alphaworks.ibm.com](http://alphaworks.ibm.com). Cette version fonctionne dans deux configurations : sans interface vocale, avec simulation par le clavier des interactions de l'utilisateur, et une version vocale qui nécessite le logiciel ViaVoice d'IBM version US.

Nous allons regarder comment rédiger un fichier VoiceXML. Voici un premier fichier.

```
<?xml version="1.0"?>
<vxml version="1.0">
  <form>
    <block>Hello World!</block>
  </form>
</vxml>
```

Cet exemple demande au navigateur vocale de diffuser le message « Hello World! ». Un moteur de synthèse à partir du texte analyse le texte contenue dans le marqueur <block> et diffuse en synthèse vocale le message.

Si le message est pré-enregistrés par une voix chaleureuse, le fichier XML doit utiliser le marqueur <audio> et faire référence au fichier vocal correspondant.

```
<?xml version="1.0"?>
<vxml version="1.0">
```

```

<form>
  <audio src="hello.wav"/>
</form>
</vxml>

```

Deux ergonomies sont proposé avec VoiceXML : Les menus et les formulaires.

Les menus permettent de proposer à l'utilisateur différents chemins. Celui-ci traverse les liens à l'aide des touches du téléphone ou de mot clef reconnu par le moteur de reconnaissance vocale.

```

<menu>
  <prompt>Bonjour. Choisissez entre : <enumerate/></prompt>
  <choice next="/sport/index.vxml">
    Sport
  </choice>
  <choice next="/meteo/index.vxml">
    Météo
  </choice>
  <choice next="/finance/index.vxml">
    Finance
  </choice>
  <noinput>
    Indiquez l'un des mots <enumerate/>
  </noinput>
</menu>

```

Le marqueur `<enumerate>` diffuse les différents mots possibles du menu. Une session pourrait ressembler à ceci :

```

Serveur : Bonjour. Choisissez entre : Sport, Météo, Finance.
Utilisateur : bourse
Serveur : Indiquez l'un des mots Sport, Météo, Finance.
Utilisateur : météo
Serveur : diffuse l'ergonomie décrite dans le fichier /meteo/index.vxml.

```

La deuxième ergonomie proposée par VoiceXML permet à l'utilisateur d'entrer un formulaire. Une boucle principale s'exécute tant que tous les champs du formulaire ne sont pas renseignés correctement.

```

<form id="meteo">
  <block>Bienvenu sur le service de météo.</block>
  <field name="departement">
    <prompt>Quel département ?</prompt>
    <grammar src="departement.gram" type="application/x-jsgf"/>
    <catch event="aide">
      Indiquez distinctement le nom d'un département.
    </catch>
  </field>
  <field name="ville">
    <prompt>Quelle villes ?</prompt>
    <grammar src="ville.gram" type="application/x-jsgf"/>
    <catch event="aide">
      Indiquez une ville du département.
    </catch>
  </field>
  <block>
    <submit next="meteo.jsp" namelist="departement ville"/>
  </block>

```

Ce formulaire demande à l'utilisateur d'indiquer le département et la ville où il désire obtenir la météo. Le marqueur `<grammar>` permet d'indiquer un fichier contenant l'ensemble des mots du vocabulaire concerné par le champ.

Le marqueur `<catch>` permet de capturer des évènements. En l'occurrence, l'exemple capture le mot `aide` pour diffuser un message particulier.

Tant que l'ensemble des champs n'est pas renseigné, le navigateur VoiceXML ne soumet pas la requête. Une fois le formulaire entièrement validé, le marqueur `<submit>` s'occupe de demander la page `meteo.jsp` en lui indiquant les deux champs `departement` et `ville`.

Techniquement, les formulaires permettent de valoriser des variables JavaScript dont le nom est indiqué dans le marqueur `<field>`. Le marqueur `<submit>` indique la liste des variables à soumettre.

Les variables JavaScript sont également manipulable par d'autres marqueurs VoiceXML.

```

<var name="un" expr="1"/>
<field name="deux" expr="un+1" type="number"/>

```

Cela permet de proposer une valeur calculée à un champ ou d'adapter le formulaire à l'aide d'`<if>`

```

<catch event="aide">
  <if cond="type_carte == 'amex'">
    Indiquez 15 chiffres.
  <else/>
    Indiquez 16 chiffres.

```

```
</if>  
</catch>
```

Dans un formulaire, l'utilisateur peut être amené à enregistrer un message.

```
<form>  
  <record name="greeting" beep="true" maxtime="10s"  
    finalsilence="4000ms" dtmfterm="true" type="audio/wav">  
    <prompt>Après le bip, enregistrez votre message.</prompt>  
    <noinput>Je n'ai rien entendu. Recommencer.</noinput>  
  </record>  
  ...  
</form>
```

Le fichier correspondant est envoyé au serveur http lors de la soumission du formulaire.

Des scripts permettent d'enrichir les fonctionnalités de VoiceXML.

```
<?xml version="1.0"?>  
<vxml version="1.0"?>  
<script><![CDATA[  
function factorial(n) { return (n <= 1)? 1 : n * factorial(n-1); }  
]]> </script>  
...  
</vxml>
```

VoiceXML permet d'offrir facilement une application vocale à partir d'un site Internet. Pour que cela fonctionne, il faut faire appel à un provider qui s'occupera d'héberger le standard téléphone et le serveur vocal (DirectTalk d'IBM par exemple). Le serveur http possédant l'application vocale peut se trouver n'importe où sur le réseau. A quand votre page vocale perso ?

Liens :

W3C : [www.w3c.org](http://www.w3c.org)

VoiceXML : [www.voicexml.org](http://www.voicexml.org)

WML : [www.allnetdevices.com/faq/](http://www.allnetdevices.com/faq/)