

Introducción

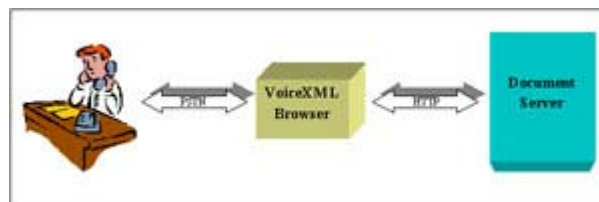
VoiceXML es una especificación propuesta por la W3C que tiene como objetivo crear archivos XML, llamados documentos, que puedan reproducir sonido digitalizado, sonido sintetizado usando la tecnología TTS(1), reconocer información ingresada por el usuario (tonos DTMF(2)) y reconocer palabra y/o frases pronunciadas por una persona, todo esto usando un dispositivo telefónico (teléfono clásico, celular o cualquier otra variante)

VoiceXML esta basado completamente en XML, es decir necesita que el documento VoiceXML sea "bien formado" para que pueda ser reconocido como correcto. Esto no ocurre con HTML, pero sí con XHTML.

Actualmente esta especificación se encuentra en la versión 2.0 la cual ha recibido el estado de "Recomendado" por parte de la W3C, faltándole muy poco para ser declarada oficialmente como estándar, aunque en estos momentos ya es un estándar "de facto".

Características Generales

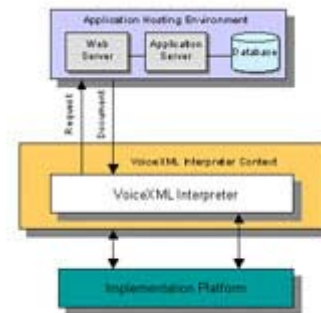
Antes de entrar en discutir VoiceXML tenemos que entender como funciona VoiceXML, para ello revisemos la siguiente figura.



- Mientras que HTML permite crear interfaces "gráficas" para que el usuario ingrese y reciba información, VoiceXML genera interfaces orales. Es decir, el usuario no "ve" la información, la escucha.
- Como VoiceXML no genera interfaces gráficas, el usuario no necesita una computadora (ni una PDA) solo le basta con un teléfono.
- El usuario se conecta al Browser a través la Red Publica de Telefonía (PSTN(3))
- VoiceXML, al igual que HTML, necesita de un browser para procesar la información, cada browser debe ser capaz de reconocer y procesar las etiquetas de cada lenguaje.
- VoiceXML, al igual que HTML, necesita de un browser para procesar la información, cada browser debe ser capaz de reconocer y procesar las etiquetas de cada lenguaje.

Arquitectura

VoiceXML se basa en la siguiente arquitectura.



- **Application Hosting Environment**
Llamado también "Document Server". Es un ambiente que genera dinámicamente documentos VoiceXML. Básicamente esta compuesto por 3 componentes.
 - **Web Server**
Servidor Web que recibe HTTP Request y envía HTTP Response con un documento VoiceXML.
 - **Application Server**
Servidor de aplicaciones que mantiene una lógica de negocio que sobre la base de los parámetros enviados por el Web Server genera documentos VoiceXML.
 - **Database**
Base de Datos de la cual se obtiene información para generar los documentos VoiceXML
- **VoiceXML Interpreter**
Aplicación que recibe un documento VoiceXML y lo interpreta, es decir procesa las etiquetas que dicho documento contiene.
- **VoiceXML Interpreter Context**
Modulo del VoiceXML Interpreter que monitorea las posibles actividades que los usuarios realizan mientras se esta interpretando un documento VoiceXML, por ejemplo el usuario podría presionar desconectarse (colgar el teléfono), lo generaría que cancelación de la interpretación del documento.
- **Implementation Platform**
Este componente viene a ser el Browser en si, pues cada empresa puede desarrollar su propio VoiceXML Browser el cual aparte de interpretar un documento VoiceXML puede implementar mecanismos de cache, procesamiento de llamadas telefónicas, etc.

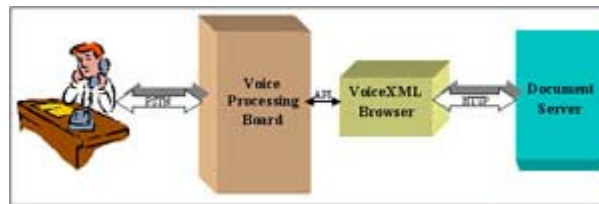
Estos son los componentes generales de la arquitectura de VoiceXML, sin embargo hay empresas que desarrollan VoiceXML Browsers y le adicionan funcionalidades no detalladas en la especificación, lo cual no ocurre solo con VoiceXML sino con casi todas las especificaciones.

¿Y quien procesaría las llamadas telefónicas?

Esa es una buena pregunta. A diferencia de HTML en la que todo se basa en http, VoiceXML, mejor dicho el VoiceXML Browser, recibe la información desde el PSTN(3) por lo tanto debe existir algún mecanismo por el cual se reconozcan y procesen las llamadas telefónicas.

Este proceso requiere de la vinculacion de estos procesos y valores obtenidos al ivr del pbx que funcionaria como VoiceXML Browser.

En realidad, la figura numero 1 se convertiría en:



Y desde una perspectiva más global la Arquitectura seria la siguiente:



How to

Desde mi perspectiva personal el uso de esta tecnología ha sido funcional para aplicaciones propietarias de desarrollo de reconocimiento de voz y comandos en voz, es importante reconocer que la aplicación de la misma para web ó apps portables no ha sido extensamente explotada. Es impórtate acotar también que aplicaciones propietarias de reconocimiento de voz como **Lumen** ó **Verbio** están desarrolladas en su totalidad bajo este lenguaje.

Ejemplos de uso:

www.lakevision.tv/voice

www.lakevision.tv/voz

<http://www.lakevision.tv/voice/GetDigits.php>

Para visualizar o usar correctamente estos archivos es necesario usar un explorador de Internet (Web Browser) que soporte voz como Opera Navigator.

Para el caso especifico de Opera después de instalarlo es necesario habilitar el modulo de voz dentro de Herramientas > Opciones > Avanzado > Voz.

Ing. Juan Pablo Díaz Ezcurdia

www.lakevision.tv

www.lakemp.com/jdiaz

Glosario

1. TTS: Text to Speech.
Tecnología mediante la cual es posible convertir teóricamente cualquier texto, en formato escrito, en palabras, en formato oral. Dicho en otras palabras, tecnología que permite leer texto.
Por ejemplo se puede ingresar a una aplicación el texto "Tecnologías de Voz" y dicha aplicación pronunciara estas palabras según la configuración de la misma (idioma, género de la voz, volumen, etc.)
2. DTMF: Dual Tone Multi-Frequency
Sistema utilizado por los teléfonos para asignar un tono (frecuencia) a cada tecla de un teléfono. Es por esta razón que uno escucha un sonido distinto cuando presiona las teclas de un teléfono.
3. PSTN: Public Switched Telephony Network
Nombre otorgado a la red de telefonía clásica

Referencias

- Voice Extensible Markup Language (<http://www.w3.org/TR/voicexml20/>)
- VoiceXML Forum (<http://www.voicexml.org/>)
- Intel Telecom Products
(<http://www.intel.com/design/network/products/telecom/index.htm>)