# Lenguage VoiceXML

## Abstract

Este documento especifica VoiceXML, la Voz Lengua de Margen de beneficio Extensible. VoiceXML es diseñado para crear diálogos de audio que presentan el discurso sintetizado, digitalizó el audio, el reconocimiento de la entrada clave dicha y DTMF, la grabación de entrada dicha, telefonía, y mezcló conversaciones iniciales. Su objetivo principal es traer las ventajas de desarrollo Basado en la web y entrega contenta a aplicaciones de respuesta de voz interactivas.

# OVERVIEW

Este documento define VoiceXML, la Voz Lengua de Margen de beneficio Extensible. Su fondo, conceptos básicos y uso son presentados en la Sección 1. Las construcciones de diálogo de forma, menú y relación, y el mecanismo (Algoritmo de Interpretación de Forma) por que ellos son interpretados son introducidas entonces en la Sección 2. La introducción de datos por el usuario usando DTMF y gramáticas de discurso es cubierta en la Sección 3, mientras la Sección 4 cubre la salida de sistema usando la síntesis de discurso y registró el audio. Mecanismos para manipular el flujo de control de diálogo, incluso variables, los acontecimientos, y los elementos ejecutables, son explicados en la Sección 5. Los rasgos de ambiente, como parámetros y propiedades así como manejo de recurso son especificados en la Sección 6. Los apéndices proporcionan la información adicional incluso el Esquema VoiceXML, una especificación detallada del Algoritmo de Interpretación de Forma y cronometraje, formatos de archivo de audio, y declaraciones que se relacionan con conformidad, internacionalización, accesibilidad e intimidad.

Los orígenes de VoiceXML comenzaron en 1995 cuando una lengua de diseño de diálogo XML-basada tuvo la intención de simplificar el proceso de desarrollo de aplicación de reconocimiento de voz dentro de un proyecto de AT&T llamado la Lengua de Margen de beneficio Telefónica (PML - Phone Markup Language). Como AT&T reorganizado, los equipos en AT&T, Translúcido y Motorola siguieron trabajando solos PML-como lenguas.

En 1998, W3C recibió una conferencia sobre navegadores de voz. Para estas fechas, AT&T y Translúcido tenía variantes diferentes de su PML original, mientras Motorola había desarrollado VoxML, y la IBM desarrollaba su propio SpeechML. Muchos otros asistentes en la conferencia también desarrollaban lenguas similares para el diseño de diálogo; por ejemplo, como HP's TalkML y PipeBeach's VoiceHTML.

El Foro VoiceXML fue formado entonces por AT&T, IBM, Translúcida, y Motorola para reunir sus esfuerzos. La misión del Foro VoiceXML era definir una lengua de diseño de diálogo estándar que los reveladores podrían usar para construir aplicaciones conversacionales. Ellos eligieron XML como la base para este esfuerzo porque estaba claro para ellos que esta era la tecnología de dirección iba.

En 2000, el Foro VoiceXML lanzó VoiceXML 1.0 al público. Dentro de poco a partir de entonces, VoiceXML 1.0 fue presentado al W3C como la base para la creación de un nuevo estándar internacional. VoiceXML 2.0 es el resultado de este trabajo basado en la entrada de compañías de Miembro W3C, otros Grupos de trabajo W3C, y el público.

Los reveladores familiares con VoiceXML 1.0 son en particular dirigidos a Cambios de la Versión Pública Anterior que resume como VoiceXML 2.0 se diferencia de VoiceXML 1.0.

## INTRODUCCIÓN

VoiceXML es diseñado para crear diálogos de audio que presentan el discurso sintetizado, digitalizó el audio, el reconocimiento de la entrada clave dicha y DTMF, la grabación de entrada dicha, telefonía, y mezcló conversaciones iniciales. Su objetivo principal es traer las ventajas de desarrollo Basado en la web y entrega contenta a aplicaciones de respuesta de voz interactivas.

Aquí están dos ejemplos cortos de VoiceXML. El primer es el venerable "Hola Mundo":

<?xml version="1.0" encoding="UTF-8"?>

<vxml xmlns="http://www.w3.org/2001/vxml"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.w3.org/2001/vxml

http://www.w3.org/TR/voicexml20/vxml.xsd"

version="2.0">

<form>

<block>Hello World!</block>

</form>

</vxml>

El elemento de alto nivel es <vxml>, que es principalmente un contenedor para diálogos. Hay dos tipos de diálogos: forms y menús. Los forms presentan la información y juntan la entrada; los menús ofrecen opciones de que hacer después. ¡Este ejemplo tiene una forma sola, que contiene un bloque que sintetiza y presenta "Hola el Mundo!" al usuario. Ya que la forma no especifica un diálogo de sucesor, los finales de conversación.

Nuestro segundo ejemplo pide al usuario una opción de bebida y luego lo presenta a una escritura de servidor:

<?xml version="1.0" encoding="UTF-8"?>

<vxml xmlns="http://www.w3.org/2001/vxml"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.w3.org/2001/vxml

http://www.w3.org/TR/voicexml20/vxml.xsd"

version="2.0">

<form>

<field name="drink">

<prompt>Would you like coffee, tea, milk, or nothing?</prompt>

<grammar src="drink.grxml" type="application/srgs+xml"/>

</field>

<block>

<submit next="http://www.drink.example.com/drink2.asp"/>

</block>

</form>

</vxml>

Un *field* es un campo de entrada. El usuario debe proporcionar un valor al campo antes de seguir al siguiente elemento en la forma. Una interacción de muestra es:

C (computer): Would you like coffee, tea, milk, or nothing?

H (human): Orange juice.

C: I did not understand what you said. (a platform-specific default message.)

C: Would you like coffee, tea, milk, or nothing?

H: Tea

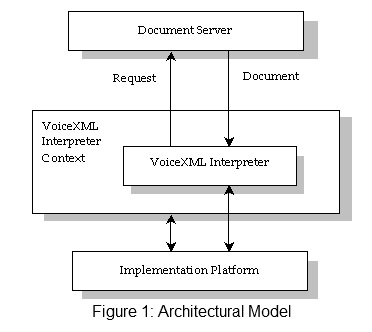
C: (continues in document drink2.asp)

## BACKGROUND

Esta sección contiene un modelo arquitectónico de alto nivel, cuya terminología es usada entonces para describir los objetivos de VoiceXML, su alcance, sus principios de diseño, y las exigencias que esto coloca en los sistemas que lo apoyan.

### MODELO ARQUITECTONICO

El modelo arquitectónico asumido por este documento tiene los componentes siguientes:



Un servidor de documento (p.ej un servidor de Web) trata peticiones de una aplicación de cliente, el Intérprete VoiceXML, por el contexto de intérprete VoiceXML. El servidor produce documentos VoiceXML en respuesta, que son tratados por el intérprete VoiceXML. El contexto de intérprete VoiceXML puede supervisar introducciones de datos por el usuario en la paralela con el intérprete VoiceXML. Por ejemplo, un contexto de intérprete VoiceXML siempre puede escuchar para una frase de fuga especial que toma al usuario a un ayudante personal de alto nivel, y el otro puede escuchar para frases de fuga que cambian preferencias de usuario como características de texto al discurso o volumen.

La plataforma de realización (Implementation Platform) es controlada por el contexto de intérprete VoiceXML y por el intérprete VoiceXML. Por ejemplo, en una aplicación de respuesta de voz interactiva, el contexto de intérprete VoiceXML puede ser responsable de descubrir una llamada entrante, adquiriendo el documento de VoiceXML inicial, y contestando la llamada, mientras el intérprete VoiceXML conduce el diálogo después de la respuesta. La plataforma de realización genera acontecimientos en respuesta a acciones de usuario (p.ej dicho o entrada de carácter recibida, desconecte) y los acontecimientos de sistema (p.ej expiración de temporizador). Algunos de estos acontecimientos son seguidos por el intérprete VoiceXML sí mismo, como especificado por el documento VoiceXML, mientras los otros son seguidos por el contexto de intérprete VoiceXML.

### OBJETIVOS DE VoiceXML

El objetivo principal de VoiceXML es traer el poder lleno de desarrollo de Web y entrega contenta para expresar aplicaciones de respuesta, y liberar a los autores de tales aplicaciones de programación baja y administración de recursos. Esto permite la integración de servicios de voz con servicios de datos usando el paradigma de cliente-servidor familiar. Un servicio de voz es visto como una secuencia de diálogos de interacción entre un usuario y una plataforma de realización. Los diálogos son proporcionados por servidores de documento, que pueden ser externos a la plataforma de realización. Los servidores de documento mantienen la lógica de servicio total, realizan base de datos y operaciones de sistema de herencia, y producen diálogos. Un documento de VoiceXML especifica cada diálogo de interacción para ser conducido por un intérprete de VoiceXML. La introducción de datos por el usuario afecta la interpretación de diálogo y es coleccionada en peticiones presentadas a un servidor de documento. El servidor de documento contesta con otro documento de VoiceXML para seguir la sesión del usuario con otros diálogos.

VoiceXML es una lengua de margen de beneficio que:

* Minimiza interacciones de cliente/servidor especificando interacciones múltiples por documento.
* Autores de aplicación de escudos de detalles bajos, y específicos para la plataforma.
* Separa el código de interacción de usuario (en VoiceXML) de la lógica de servicio (p.ej. Escrituras de CGI).
* Promueve la portabilidad de servicio a través de plataformas de realización. VoiceXML es un lenguaje común para abastecedores contentos, abastecedores de instrumento, y abastecedores de plataforma.
* Es fácil de usar para interacciones simples, y aún proporciona rasgos de lengua para apoyar diálogos complejos.

Mientras VoiceXML se esfuerza por acomodar las exigencias de una mayoría de servicios de respuesta de voz, los servicios con exigencias rigurosas pueden ser mejor servidos por aplicaciones dedicadas que emplean un nivel más fino del control.