Capitulo 2 Antecedentes del Proyecto

2.1 Visión general del Proyecto.

El objetivo del presente proyecto es proveer a la comunidad universitaria de una Plataforma de Pruebas para implementaciones VoiceXML para poder crear aplicaciones que tengan como meta transmitir información a personas que no cuenten con Internet o una computadora.

La aportación que brinda los resultados de este proyecto serán una ayuda para el desarrollo educativo de la Universidad de las Américas Puebla, pues el estar a la vanguardia con las tecnología es un punto importante de las Universidades, y si este proyecto en un futuro lograra ser base de otros mucho más grandes, sabré, que le pude brindar algo a la sociedad, que en un futuro será el contexto donde vivan las generaciones venideras, sería una aportación para el crecimiento humano.

2.2. Orígenes y Evaluación

En este momento en que se tiene un panorama general del proyecto, es necesario tomar una decisión acerca de la metodología para la selección del intérprete de lenguaje de marcación VoiceXML a emplear, Las opciones posibles son OpenVXI y la otra JVoice XML. Debido a que esta decisión es de vital importancia para obtener los mejores resultados, se opto por crear un una matriz de ponderación y analizar cual sería la mejor opción, al final nos dio como resultado OpenVXI la opción mas viable y menos riesgosa de seleccionar como interprete base.

Ver Anexo1. Matriz de Ponderación.

2.3. Definiciones y conceptos básicos.

VoiceXML (Voice eXtensible Markup Language) es un formato XML estándar de W3C (World Wide Web Consortium) para especificar diálogos interactivos de voz entre un ser humano y una computadora [5].

OpenVXI es una librería de software abierto que interpreta el lenguaje de marcación de diálogo de VoiceXML. Incluye reconocimiento del habla simulado, capacidades para desplegar avisos, conversiones de texto-a-voz (TTS “Text-To-Speech”), y funciones telefónicas. [6]

La solución que este proyecto propone es el de emplear VXML se optó manejar ***OpenVXI*** [7;8] debido a que es considerado un componente importante para crear portales de voz, y además cuenta con las siguientes características importantes:

- Disponible.

- Es un software libre (Open source).
- Configurable.

2.4. Diagrama General

Ofreciendo una visión más clara de lo que representa el contexto de esta plataforma, se muestran a continuación en la Figura 1, aquí se presenta la arquitectura general de la plataforma a emplear, así como los componentes que la integran. Todos los componentes están diseñados para ser portátiles o modulares dentro de un sistema operativo.

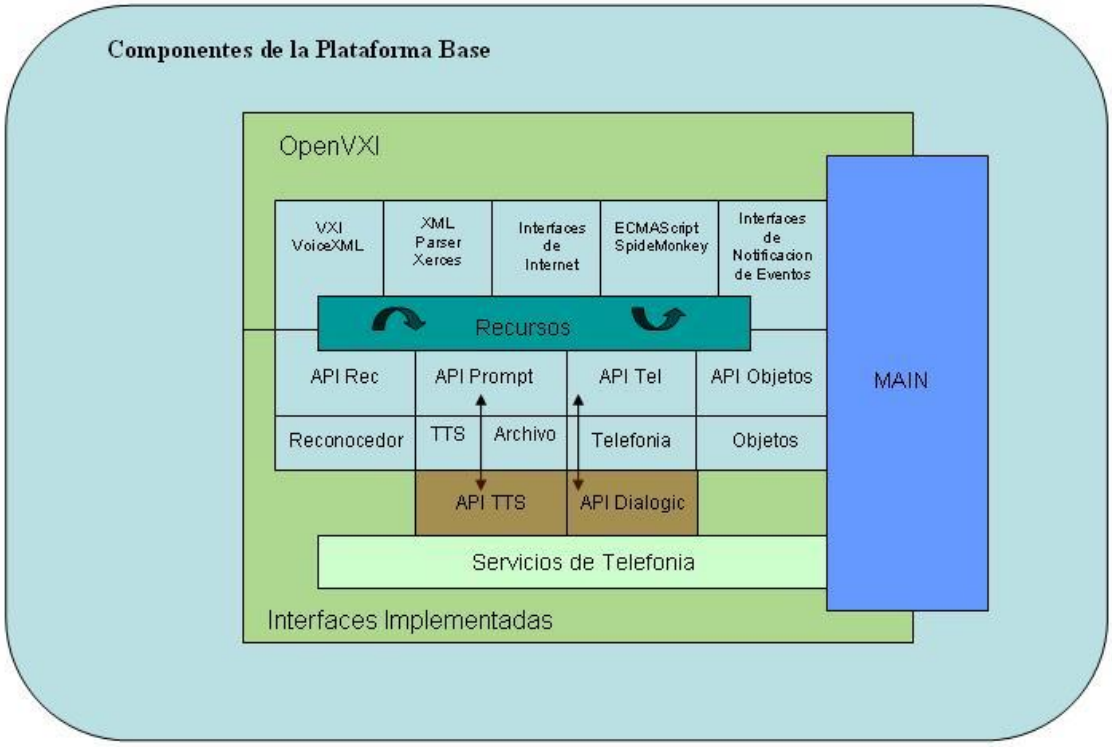

Figura 1. Diagrama General.

La descripción de los componentes que integran OpenVXI son:

a) VXI.

Interpreta el lenguaje VoiceXML y actúa como el control maestro del sistema principal. VXI implementa todas las partes requeridas de las especificaciones de VoiceXML y de la mayoría de las características opcionales.

b) XML Parser.

Su función principal es que proveer el acceso a un programa del análisis de XML, el DOM parser, siendo implementado para llamar a Apache Xerces SAX y a los APIs del programa de análisis del mismo programa DOM parser.

c) Interfaces de Internet.

Proporcionan el acceso a documentos de aplicaciones vía *http://* y *file://*, además que ayuda a fijar los datos al servidor de aplicaciones.

d) ECMAScript (JavaScript) Interface. SipiderMonkey.

Proporciona el acceso a los servicios de la ejecución de ECAMAScript. La implementación de referencia distribuida con OpenVXI integra el motor Mozilla SpiderMonkey.

e) Interfaz de Notificación de eventos.

Esta interfaz es empleada para desplegar mensajes de errores, eventos y mensajes de diagnóstico a los operadores del sistema. La implementación de referencia distribuida con OpenVXI imprime estos acontecimientos directamente en la consola, pero también permite la implementación de salidas estándar.

La plataforma de implementación debe contener las siguientes interfaces para asegurar el funcionamiento de cualquier aplicación VoiceXML:

a) Interfaz de Reconocimiento *(Rec Interface).*

Provee el manejo de la gramática y los servicios de reconocimiento que son requeridos por las especificaciones de VoiceXML, incluyendo construcción de gramática dinámica. Obtiene entrada de voz del usuario por medio de los servicios de telefonía.

b) Interfaz de Avisos *(Prompt Interface).*

Proporciona el servicio de avisos, incluyendo la reproducción de audio. Maneja archivos de audio pregrabado y provee servicios de generación de voz artificial text-to-speech, regresando audio al usuario que efectúa la llamada telefónica

c) Interfaz Telefónica.

Proporciona servicios de control de llamadas, incluyendo la transferencia y desconexión de las llamadas, además de reportar los eventos que ocurren en la misma la red telefónica.

d) Interfaz de Objetos.

Proporciona acceso a los objetos, extensiones definidas de la plataforma al lenguaje VoiceXML que son accezadas a través de objetos. Los objetos pueden ser definidos fácilmente para cumplir con los requerimientos específicos de la plataforma para el control extendido de llamadas y otros requerimientos. [9]

e) Interfaz Dialogic.

Se encarga de proporcionar el servicio de las comunicaciones en tiempo real entre el usuario (por medio del teléfono) y OpenVXI.
Es también la encargada de manejar el reporteo de eventos referentes a telefonía en conjunto la interfaz telefónica.

f) Interfaz TTS.

Provee un ambiente de desarrollo para crear tus propios mensajes y convertirlos a voz, ajustarlos, manipularlos y guardarlos para su futura utilización.
Pone al alcance un control completo sobre sus aplicaciones de voz, con lo que se puede de verdad mejorar la experiencia del usuario.

El texto se puede refinar mediante un procedimiento de análisis de nuestro archivo Vxml. Con todos estos instrumentos a disposición, se puede lograr que nuestro mecanismo generador de voz artificial TTS hable como más nos guste.