

Interoperability between HanPhoneXML and VoiceXML

1. Introduction

HanPhone Server is a powerful and flexible voice application platform developed by KanHan Technologies. It encapsulates computer telephony hardware and various TTS (text-to-speech) engines, allowing developers to build voice applications using a subset of HTML as well as a simple markup language called HanPhoneXML.

HanPhoneXML, an application of XML, is designed for creating audio dialogs featuring synthesized speech, digitized audio, DTMF key input recognition, spoken input recording and telephony functionalities. Its major goal is to provide a markup language that is simple but powerful enough to build voice applications on the HanPhone Server that can supplement or even replace existing proprietary IVR systems.

Another key design goal of HanPhoneXML is to make the language easy to learn for web developers familiar with HTML. Therefore, programming features such as variable manipulation and function calls are not included. The overall application logic is left to be controlled by the content server, which provides the audio dialogs to be executed by the HanPhone Server through static or dynamically generated HanPhoneXML using any server-side applications such as PHP, ASP and JSP:

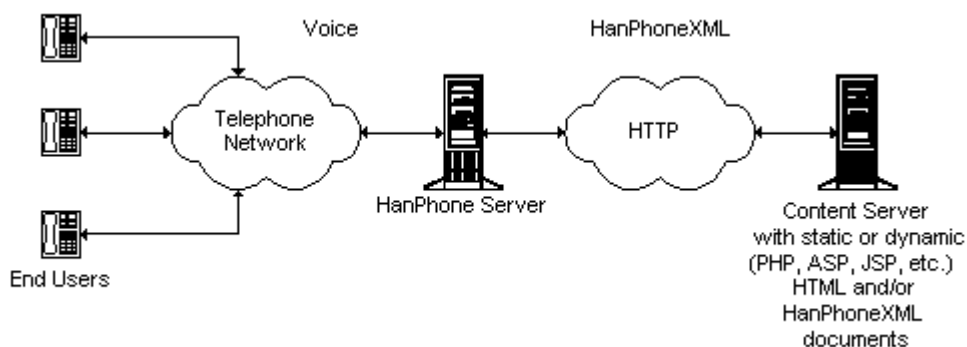


Figure 1. Delivery of HanPhoneXML via HanPhone Server

2. Differences between HanPhoneXML and VoiceXML

HanPhoneXML is similar to VoiceXML in many ways:

- Both are applications of the XML 1.0 specification. This means any standard-compliant XML parser can retrieve the content inside HanPhoneXML documents easily.
- Being an XML application, HanPhoneXML documents can be transformed to other types of XML documents through the use of XSLT.
- Many HanPhoneXML elements closely resemble VoiceXML elements since both languages support telephony and common functionalities available in existing voice applications. Therefore HanPhoneXML can be considered a variant of a subset of VoiceXML.

Besides the similarities, HanPhoneXML has the following advantages:

- Fewer elements and the absence of programming language features resulting in a simpler syntax, making the learning curve less steep.
- The HanPhone Server has a built-in HTML-to-HanPhoneXML converter.
- HanPhoneXML allows playback of pre-recorded audio file starting at an arbitrary position, not just the beginning.

3. Interoperability between HanPhoneXML and VoiceXML

As mentioned earlier, most HanPhoneXML elements can be converted/transformed to VoiceXML elements by various means. The following is a table of HanPhoneXML elements and their near equivalents in VoiceXML:

HanPhoneXML	VoiceXML	Remarks
card		No equivalent. Forms and menus in VoiceXML are direct child elements of the root element.
content	prompt	Similar.
select	menu	Similar.
option	choice	Similar.
disconnect	disconnect	Identical.
call_transfer	transfer	Similar.
record	record	Similar.
random		No equivalent.
form	form	Similar.
input	field	Similar.
sleep		No equivalent.
dial		No equivalent.
if	if	HanPhoneXML: determine whether a new URI should be retrieved after dial / call_transfer. VoiceXML: determine which block of content to execute.
wait_call		No equivalent.

Appendix I. HanPhoneXML Elements**1. <card>**

attributes	value	meaning
id	any alphabet	a card's identifier
language	numbers (0 - English only, 1- Cantonese & English, 2 - Mandarin & English	TTS's language speaking
defaultmenu	1- enable (default), 0 - disable	play "press 00 to xxx, ** to xxx"

2. <content>

attributes	value	meaning
src	sound file name	use external pre-recorded sound file
href	other template filename	use external template
startpos	numbers (unit of seconds)	sound file's starts at X seconds
keystop	0 - no, 1 - yes (default)	press key to stop playing sound
spellnum	0 - no (default), 1 - yes	spell the numbers
language	numbers (0 - English only, 1- Cantonese & English, 2 - Mandarin & English	TTS's speaking language
charset	numbers(1 - BIG5(default) , 2 - GB)	Character encoding
duration	number (unit of second)	how long it plays

3. <select>

attributes	value	meaning
retry	numerical number	how many times to retry when encounter error input
language	numbers (0 - English only, 1- Cantonese & English, 2 - Mandarin & English	TTS's speaking language
charset	numbers(1 - BIG5(default) , 2 - GB)	Character encoding
default	number	default option if timeout
timeout	number, in unit of sec. (default 3)	
maximum	numbers	maximum digits of user input

4. <option>

attributes	value	meaning
key	numerical number, '#', '*'	keys in a traditional telephone
href	start with '#' or template file name	refer to other internal card's id, or a external template file name
language	numbers (0 - English only, 1- Cantonese & English, 2 - Mandarin & English	TTS's speaking language
src	sound file name	use external pre-recorded sound file
charset	numbers(1 - BIG5(default) , 2 - GB)	Character encoding
value	string	value

5. <disconnect>

attributes	value	meaning

6. <call_transfer>

attributes	value	meaning
blind	0 or 1 (default 1)	blind transfer or not
number	string	telephone number

7. <record>

attributes	value	meaning
time	numerical number	record time in seconds, default 30 sec.
target	filename	recording filename and path
type	0 (vox format), 1 (wav format)	record file type, default 0

8. <random>

attributes	value	meaning
		random pick a child

9. <form>

attributes	value	meaning
language	numbers (0 - English only, 1- Cantonese & English, 2 - Mandarin & English	TTS's speaking language
action	http links	submit a http form
charset	numbers(1 - BIG5(default) , 2 - GB)	Character encoding

enctype	"application/x-www-form-urlencoded" (default), or "multipart/form-data"	form post encryption
---------	---	----------------------

10. <input>

attributes	value	meaning
language	numbers (0 - English only, 1- Cantonese & English, 2 - Mandarin & English	TTS's speaking language
src	sound file name	use external pre-recorded sound file
name	a string	a variable name for this tag
maximum	numbers	maximum digits of user input
stop	0-9, *, #	a stop digit for input
type	"text", "radio", "hidden", "file"	input type (HTML like)
checked	checked	default value
value	characters	
charset	numbers(1 - BIG5(default) , 2 - GB)	Character encoding

11. <sleep>

attributes	value	meaning
value	number, default 0, max 60	unit of second

12. <dial>

attributes	value	meaning
number	string	telephone number

13. <if>

attributes	value	meaning
value	string (busy, connected, noanswer, error, default)	dial result/ call_transfer result
href	start with '#' or template file name	refer to other internal card's id, or a external template file name

14. <wait_call>

attributes	value	meaning
timeout	number of sec. (default -1, infinite waiting)	unit of second

Appendix II. Examples of Equivalent HanPhoneXML and VoiceXML Documents

The following is a HanPhoneXML document that reads a short greeting and prompts the user to select an option. The user can retry twice if he/she enters a wrong value before the server disconnects. If no input is given within 5 seconds, the first option will be chosen:

```
<?xml version="1.0"?>
<xml>
  <card>
    <content>Hello guest!</content>
    <select default="1" timeout="5" retry="2">
      <option key="1" href="news.xml">News</option>
      <option key="2" href="weather.xml">Weather</option>
      <option key="4" href="others.xml">Others</option>
    </select>
  </card>
</xml>
```

The following is an equivalent VoiceXML document:

```
<?xml version="1.0"?>
<vxml version="1.0">
  <form id="form1">
    <block>Hello guest!</block>
    <goto next="#menu1"/>
  </form>
  <menu id="menu1">
    <prompt timeout="5s">
      <enumerate>
        Press <value expr="_dtmf"/> for <value expr="_prompt"/>.
      </enumerate>
    </prompt>
    <choice dtmf="1" next="news.xml">News</choice>
    <choice dtmf="2" next="weather.xml">Weather</choice>
    <choice dtmf="4" next="others.xml">Others</choice>
    <catch event="nomatch">
      <reprompt/>
    </catch>
    <catch event="nomatch" count="3">
      <disconnect/>
    </catch>
    <catch event="noinput">
      <goto next="news.xml"/>
    </catch>
  </menu>
</vxml>
```

The following is a HanPhoneXML document that implements a form to collect some user input and submit it to an HTTP server to retrieve a new document. It has a hidden variable `foo`, a 4-digit `item_id` to be input by the user and a `type` to be selected by the user:

```
<?xml version="1.0"?>
<xml>
  <card>
    <content>Hello guest!</content>
    <form action="http://www.foo.com/bar">
      <input type="hidden" name="foo" value="bar"/>
      Book ID: <input type="text" name="item_id" maximum="4" stop="#" />
      <input type="radio" name="type">
        <option key="1" value="cd" checked="checked">
          Compact Disc
        </option>
        <option key="2" value="book">
          Book
        </option>
      </input>
    </form>
  </card>
</xml>
```

The following is an equivalent VoiceXML document:

```
<?xml version="1.0"?>
<vxml version="1.0">
  <form id="form1">
    <block>Hello guest!</block>
    <goto next="#form2"/>
  </form>
  <form id="form2">
    <var name="foo" expr="'bar'"/>
    <field name="item_id" type="digits">
      <prompt>Please enter Book ID</prompt>
      <grammar mode="dtmf" src="builtin:dtmf/digits?maxlength=4"/>
    </field>
    <field name="type">
      <prompt timeout="3s">
        <enumerate>
          Select <value expr="_dtmf"/> for <value expr="_prompt"/>.
        </enumerate>
      </prompt>
      <option dtmf="1" value="cd">Compact Disc</option>
      <option dtmf="2" value="book">Book</option>
      <catch event="noinput">
        <assign name="type" expr="'cd'"/>
      </catch>
    </field>
    <filled>
      <submit next="http://www.foo.com/bar"
        namelist="foo item_id type"/>
    </filled>
  </form>
</vxml>
```