HEISON CHAK

Media Resource Control Protocol

Heison Chak is a system and network administrator at SOMA Networks. He focuses on network management and performance analysis of data and voice networks. Heison has been an active member of the Asterisk community since 2003.

heison@chak.ca

WITH TODAY'S TELEPHONY SERVERS and applications, speech capability and intelligence have been key differentiators between a superb Interactive Voice Response (IVR) system and its ordinary counterparts. Text to speech (TTS) and Automatic Speech Recognition (ASR) are means of assisting self-service IVR systems in directing calls to appropriate destinations via spoken speech. This is the most natural form of input and becomes especially convenient when callers do not have easy access to a keypad for DTMF input. Voice applications written in VoiceXML (VXML), such as address/ZIP code recognition, email collection, or alpha-numeric input, may ease the pain of performing these tedious and difficult tasks, achieving higher automation with improved accuracy. In this column we will examine how it is that clients (e.g. ,the application server, the PBX) make text requests to speech servers and get spoken speech in return.

Synthesizer/Recognizer, Then and Now

Text-to-speech (TTS) and automatic speech recognition (ASR) technologies have been around for quite some time. The first computer-based speech synthesis systems were created in the late 1950s, and the first complete text-to-speech system was built in 1968.

Research on ASR began in 1936, but the problems of speed and accuracy were not overcome until 1982, when Covox launched the first commercial product. Another company founded in the speech recognition market the same year was Dragon Systems. Dragon Systems was acquired by Scansoft Inc. in 1999. A company that had grown through acquisition, Scansoft acquired Speechworks in 2003, to become an enterprise speech solutions company, a direct competitor for Nuance, which had been established in 1994 and was by 2003 a leader in computer telephony applications and automated call steering. In 2005, a de facto acquisition of Nuance by ScanSoft took place and the combined company changed its name to Nuance. (See Figure 1.)

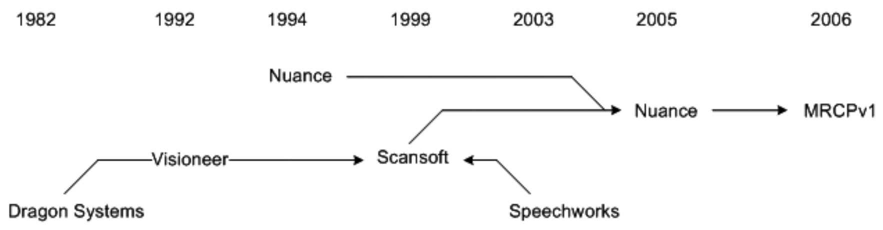

FIGURE 1: HISTORY OF MRCP AND NUANCE

RFC 4463: MRCPv1

While vendors (e.g., Nuance, IBM WebSphere Voice Server, AT&T Natural Voices) are competing for market share of TTS and ASR products, there are many proprietary systems for performing synthesis and recognition over the network. However, these nonstandard approaches have often caused migration problems and become major support issues as a result of vendor mergers and acquisitions. In 2006, Cisco, Nuance, and Speechworks jointly developed RFC 4463: Media Resource Control Protocol (MRCP), a protocol which controls media service resources such as speech synthesizers and recognizers over a network. MRCPv1 is an Informational RFC and therefore not a candidate to become an IETF Internet Standard (it was published for its historical value as an ancestor to the MRCPv2 standard). That said, MRCPv1 has been widely implemented by both speech server vendors and network equipment providers and is often viewed as a subset of MRCPv2.

A fundamental difference between MRCPv1 and MRCPv2 is seen in the mechanisms used for media session management and protocol transport. With MRCPv1, Real Time Streaming Protocol (RTSP) is used for session setup and RTP is used for media streaming. MRCPv2, in contrast, uses SIP instead of RTSP for session setup (RTP is still used for media streaming).

Consider the MRCPv2 architecture diagram (Figure 2) and the MRCPv2 message (Listing 1). When speech is sent from the client to the speech server via a SIP connection, the speech server will synthesize the speech using an installed voice (here, "Samantha") and return the speech ("Hello, my name is Samantha . . .") over an RTP stream negotiated during the SIP call setup phase.

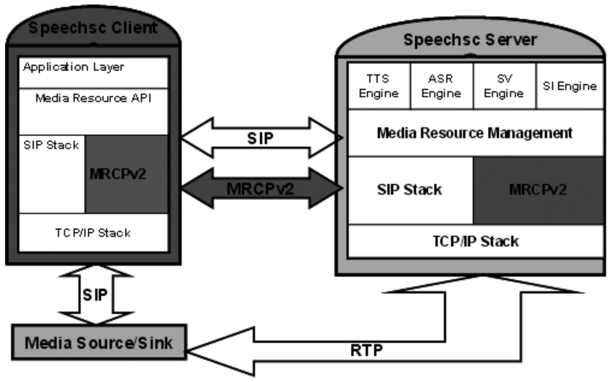

FIGURE 2: MRCPV2 ARCHITECTURE

```
MRCP/2.0 732 SPEAK 543257
Channel-Identifier:32AECB23433802@speechsynth
Voice-gender:neutral
Voice-Age:25
Voice-Name: Samantha
Prosody-volume:medium
```

```
Content-Type:application/ssml+xml
Content-Length:850
<?xml version="1.0"?>
<speak version="1.0"
  xmlns="http://www.w3.org/2001/10/synthesis"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
    http://www.w3.org/TR/speech-synthesis/synthesis.xsd"
  xml:lang="en-US">
  <p>
  <s>Hello, my name is Samantha. The time is now
    <break/>
    <say-as type="time">0345p</say-as>.
  </s>
  </p>
</speak>
```

LISTING 1: CONTENTS OF MRCPV2 DEMO _ SPEECH.MSG

Voice- (e.g., Voice-gender, Voice-Name) parameters may be sent to define values for the entire session. These parameters may also be sent to affect a request in progress and change its behavior (from a woman's voice to a man's voice) on the fly, if that functionality is supported by the synthesizer.

OpenMRCP

Despite the lack of free commercial-grade speech servers, the open source community is striving for significant improvement over what is currently available (e.g., Festival, GVoice). With MRCP-enabled speech servers (e.g., Nuance, IBM WVS), the open source project OpenMRCP becomes a perfect tool to bridge the gap. Sponsored by Cepstral, a TTS voice provider, OpenMRCP provides a full stack of MRCP which can be used as the basis of either an MRCP server or an MRCP client. The MRCP client is available in Freeswitch as a module. It may also be built against a stand-alone Apache Portable Runtime (APR).

OpenMRCP Against Stand-Alone APR

To test OpenMRCP, I built three virtual machines under Xen 3.0.3:

- Nuance RealSpeak 4.5 with Nuance Speech Server 5.0 on CentOS 5
- Nuance Recognizer 9.0 with Nuance Speech Server 5.0 on CentOS 5
- OpenMRCP with apr, apr-util 1.2.12, and sofia-sip 1.12.8 on Debian Etch

In order to build OpenMRCP against a stand-alone APR (Apache Portable Runtime), several dependencies must be met. First, I need apr and apr-util version 1.2, which are available from the Apache Portable Runtime Project (http://apr.apache.org). For OpenMRCP to support MRCPv2, a SIP stack is required. This is where the Sofia-SIP library fits in. Sofia-SIP (http://sofia-sip.sourceforge.net) is used by OpenMRCP to set up MRCPv2 sessions.

Once all the dependencies have been built, OpenMRCP can be downloaded and built:

```
# cd /usr/src
# svn co http://svn.openmrcp.org/svn/openmrcp/trunk openmrcp
# cd /usr/src/openmrcp
# ./bootstrap
# ./configure     --with-apr=/usr/src/apr-1.2.12 \
           --with-apr-util=/usr/src/apr-util-1.2.12 \
           --with-sofia-sip=/usr/src/sofia-sip-1.12.8
# make && make install
```

To test the Nuance TTS synthesizer using the MRCP protocol, I launched OpenMRCP in client mode. From the OpenMRCP CLI, new sessions can be created, followed by allocation of TTS/ASR channels where MRCPv1/v2 commands can be sent. For example:

```
# openmrcpclient -c 10.155.1.29:5060 -s 10.155.1.27:5060
# Connect to server with a new session
> create 0
# Create a TTS channel (1st 0 - session slot, 2nd 0 - channel type of TTS)
> add 0 0
# Send MRCPv2 message (as in Listing 1)
> msg 0 /path_to/demo_speech.msg
# Tearing down session (when the synthesis completes)
> destroy 0
# Leaving OpenMRCP
> quit
```

For the duration of the speech, an output file, synth_result_${session}.pcm, will grow in size as RTP packets arrive from the speech server. To verify that it worked, the raw PCM can be converted to .wav for playback:

```
# sox -t raw -r 8000 -c 1 -w -s synth_result_0.pcm demo_speech.wav
```

Next Steps

The newly resampled .wav file can now be played by most media players and is ready for use in Asterisk. It may be used as a pre-recorded IVR menu file or greetings for voice-mail boxes.

Ideally, it would be best if real-time operation of MRCP could be integrated into Asterisk. Instead of hacking up an expect script or Python to maneuver the openmrcpclient CLI and passing the output file to Asterisk for playback, many would like to see OpenMRCP as an Asterisk module, as it is in FreeSwitch. Unfortunately, the author is currently tied up with a reimplementation of MRCP that will be free from corporate sponsorship, so integration of OpenMRCP into Asterisk may not happen anytime soon.

The new project, UniMRCP, the successor to OpenMRCP, will probably continue to depend on Sofia for SIP stack as well as APR (apr.apache.org) for low-level, cross-platform implementation of threads, mutexes, and sync objects. UniMRCP is currently available for Windows and UNIX via Subversion.

REFERENCES

http://en.wikipedia.org/wiki/Speech_synthesis

http://www.dragon-medical-transcription.com/historyspeechrecognition.html

http://tools.ietf.org/html/rfc4463

http://daveburke.org/downloads/SP-Appendix-A.pdf

http://wiki.freeswitch.org/wiki/OpenMRCP#Configure_OpenMRCP_against_standalone_APR

http://www.unimrcp.org/

http://www.cisco.com