Service Creation Using SIP

Paul Doyle
Director Of Strategic Solutions for Product Management

Ubiquity Software Corporation
Suite 345
3 Lagoon Drive
Redwood City
California
CA 94065
www.ubiquity.com
info@ubiquity.com
Ubiquity develops and markets SIP-based communications software for service providers, Independent Software Vendors (ISVs) and OEM partners worldwide. The company has corporate offices in three continents (NA, Asia Europe).

Ubiquity offers two products

**SIP Application Server**
A carrier-class deployment and programmable platform

Key Features:
- Standards-based SIP Servlet API and Application Creation Environment
- Pre-built Application Building Blocks (ABBs) and non-SIP connectors
- Clustered high availability and non-HA deployment models

**Speak Conference Director**
Highly-scalable, carrier-class, SIP IP conferencing application

Key Features:
- Browser-based solution
- Complete conferencing application feature set
- Web portal supports scheduling, initiating, managing and terminating multi-party conferences
Agenda

- The Case for Standards
- Call Processing Language
- SIP Common Gateway Interface
- SIP Application Programming Interfaces
- SIP and VoiceXML
The Case For Standards

- Open standards drive choices
- Prevents customer lock-in
- Open standards are the central message of SIP and the Internet
- The Internet Engineering Task Force (IETF) == “demonstrated interoperability”
CPL: Call Processing Language

- IETF XML scripting language developed for SIP
- Used for describing and controlling *Simple* call services
- Intended for end-user service creation
- CPL interpreter easily parses and validates, guarding against malicious code
- Implemented on network or user agent servers
- Easily edited by graphical clients, and independent of operating system or signaling protocol
- Uses *primitives* for making decisions based on call properties, e.g. time of day, caller, called party
CPL: Call Processing Language

- Suitable for running on servers as it has no variables, loops, or ability to run external programs

**Issues to consider**

- Services are executed by chaining scripts together
- Complex services are not really possible

The SIP-CPL draft can be found on the [IETF website](http://www.ietf.org)
<?xml version="1.0" ?>
<!DOCTYPE cpl PUBLIC "-//IETF//DTD RFCxxxx CPL 1.0//EN" "cpl.dtd">
<cpl>
  <subaction id="voicemail">
    <location url="sip:pdoyle@voicemail.ubiquity.net">
      <redirect />
    </location>
  </subaction>
  <incoming>
    <address-switch field="origin" subfield="host">
      <address subdomain-of="ubiquity.net">
        <location url=" sip:pdoyle@ubiquity.net ">
          <proxy timeout="10">
            <busy> <sub ref="voicemail" /> </busy>
            <noanswer> <sub ref="voicemail" /> </noanswer>
            <failure> <sub ref="voicemail" /> </failure>
          </proxy>
        </location>
      </address>
    </address-switch>
    <otherwise>
      <sub ref="voicemail" />
    </otherwise>
  </incoming>
</cpl>
SIP-CGI: SIP Common Gateway Interface

- CGI scripts allow websites to interact with databases and other applications.

- Similarities between SIP and HTTP make CGI a candidate for service creation in a SIP environment.
  
  - Scripts reside in the server and pass message parameters through environment variables to a separate process.
  - This sends instructions back to the server through a standard output file descriptor.
  - SIP CGI is suitable for services that contain substantial web components.
  - Written in Perl, Tcl, C, C++ or Java, making it accessible to a large community of developers.

- Two Main issues to consider:
  
  - Security
  - Efficiency - 10 people using a CGI script – 10 instances of script processes

- The draft standard is on the IETF website.
Application Programming Interfaces

- SIP Servlets
- JAIN for SIP
- PARLAY
Background to Servlets

- Servlets are Java applications that run in Web or application servers providing server-side processing.

- Portable between servers and operating systems.

- Replacement for CGI scripts, Active Server Pages (ASPs) and proprietary plug-ins written in C and C++.

- Similar to CGI concept but, messages are passed to a class that runs within a JVM (Java Virtual Machine) instead of using a separate processes.

- Servlets run efficiently in memory:
  - 10 people using a CGI script – 10 copies
  - 10 people using a servlet – one instance
SIP Servlet Specification

- Endorsed by the IETF site
- Developed under Java Community Process JCP.org

- Provides standardized platform for delivering SIP based services.

- Stated Goals:
  - Support for UAC, UAS and proxy behaviours
  - Application composition
  - Converged applications
  - Third party application development
  - Carrier grade
  - Simplicity
  - Security
  - Expressiveness
  - Familiarity
Why SIP Servlets?

- Based on HTTP Servlet API (one of the best known APIs)
- Third-party developer accessibility (no proprietary API’s)
- Established and understood mechanism for rapidly creating applications
- High-level powerful API (Abstracted, not as low level as JAIN SIP API)
- Decoupled state and processing (similar to EJB session beans in J2EE)
Differences between SIP and HTTP servlets

- Requires some knowledge of SIP
- HTTP servlets are predicated only on responding to incoming requests
- SIP servlets must respond, proxy and initiate requests, and receive responses
- Concept of ‘session’ (call legs, multiple call legs and multi-protocol)
- SIP servlet model manages complex relationships between incoming requests and invoked servlets
Deploying a SIP Servlet

- Incoming SIP message is parsed
- Servlet container looks up XML rule file for matching DD (Deployment Descriptor)
- Container holds state information
  - SIPSession
  - SIPApplication Session
- Identifies a servlet within an application
Programming Interfaces: JAIN

The JAIN APIs are specified as a community extension to the Java™ platform.

Provides abstraction and associated Java interfaces for service creation across circuit switched and packet networks.

Key Features:

- Service Portability: Write Once, Run Anywhere
- Network Convergence: allows services to run over any underlying network architecture, IP, ATM, TDM or wireless
- Service Provider Access: specifies mechanisms to allow abstracted services direct access to network resources
JAIN in the pure SIP network

User Agents

Proxy / Redirect Server

Application Server

© 2002 Sun Microsystems, Inc. All Rights Reserved.
SIP Components of JAIN

- **JAIN SIP API** standard portable low-level interface into a SIP stack

- **The JAIN SIP Lite API** defines high level UA API to the SIP stack

- **JAIN SIMPLE** interface processes messages and presence information between a “SIMPLE” client (watcher) and a presence server (presence agent)

- **The SIP Servlet API** high-level extension API for SIP servers, based on the servlet model
Programming Interfaces: PARLAY

- Formed in 1998 to specify and promote open APIs linking IT applications with the communications world.

- Parlay API specifications are defined in Universal Modeling Language (UML) e.g. no language dependancy.

- The JAIN Service Provider APIs (SPA) define a Java technology realization of the Parlay APIs.

- Prime focus to allow applications secure access to functionality of telecoms network.

- Focused on call control and messaging.
PARLAY Architecture

Parlay APIs consist of two categories:

- **Service interfaces** offering applications access to network capabilities and information

- **Framework interfaces** providing access security and management
Other Programming Interfaces

In this lecture we have covered the main SIP based API’s and commonly known programming technologies. Others that are notable and worthy of further reading:

- ParlayX
  http://www.argreenhouse.com/parlayx/
- Microsoft’s SIP scripting language (MSPL) available in LCS
  microsoft_sip_processing_language.asp
- Microsoft’s RTC ancrtc.asp
SIP and VoiceXML

- XML: O/S and environment independent markup language
- Voice-XML (vXML) developed by the W3C [www.w3c.org/voice](http://www.w3c.org/voice) to create and negotiate audio sessions with PSTN devices
- Original goals to enhance IVR systems, for instance, and open them up to non proprietary service creation environments
- Sessions could be IVR, ASR, DTMF recognition for example
- Access to service technology is through a VXML Browser, similar to a web browser
- VXML browsers fetch VXML pages or pre-recorded media from web servers and present an interactive dialog to voice drivers for PSTN phone e.g. VXML enabled IVR
- Pages either statically stored on the web server or dynamically generated based on server side programming logic e.g. Java servlet or Java server pages (JSP)
SIP and VoiceXML

- SIP-VXML browsers are similar to VXML browsers

- The browser fetches the VoiceXML pages and interprets the relevant SIP signaling required to establish a SIP session with a SIP end point or User Agent

- Allows services previously available only to PSTN end points to be accessible IP Phones or User Agents as well as PSTN devices via IP-PSTN Gateways

- Examples of Services:
  - Call transfer
  - Conferencing Control and interaction
  - Email TT
  - Event notification
  - Scheduling

- All audio interaction via a telephone or IP User agent
Conclusion

When considering service creation technologies always examine:

- How industry supported is it? Can I get support and is there any forums or groups I can go to that are well attended?

- What is my scope for service creation? What kind of Applications do I think will be developing. What domain; Internet, Web services, core routing?

- Does my Application Creation API answer the needs of a customer or an entire channel?
Thank You!
References and Resources

References
SIP Servlets www.jcp.org/en/jsr/detail?id=116
SIP CGI www.ietf.org/internet-drafts/draft-lennox-sip-cgi-04.txt
SIP CPL www.ietf.org/internet-drafts/draft-ietf-iptel-cpl-06.txt
JAIN java.sun.com/products/jain
JAIN SIP Lite JAIN SIP Lite specification
Parlay www.parlay.org
Parlay X www.argreenhouse.com/parlayx/
VoiceXML specification www.w3c.org/voice

Papers
Programmable End System Services Using SIP (sipie.pdf)
Xiaotao Wu and Henning Schulzrinne
New Voice Services (sip-vxmlTurner-tut.pdf) Kenneth Turner

Useful Resources
The SIP Centre www.sipcenter.com
References and Resources

Referenced Material

*Programmable End System Services Using SIP* (sipie.pdf) by Xiaotao Wu and Henning Schulzrinne

*New Voice Services* (sip-vxmlTurner-tut.pdf) by Kenneth Turner

*Understanding SIP* [www.fokus.gmd.de/mobis/siptutorial/](http://www.fokus.gmd.de/mobis/siptutorial/)

*Call Processing Language (CPL) Based Service Configuration System* by Mahavir D Karnavat and Shivaji Hogale

*Integrating VoiceXML with SIP services* (SIP-vxml2272.pdf) by Kundan Singh, Ajay Nambi and Henning Schulzrinne