Overview of XML Communication Technologies

Executive Summary

A significant shift has occurred in the communications infrastructures deployed today. This shift is the result of the acceptance and growth of communications technologies, such as 4G cellular and Voice over IP. As these technologies have gained widespread acceptance, customers' expectations have also increased for a more media-centric, information-rich caller experience. This white paper provides an overview of some of the XML communication technologies rising to address these expectations, and briefly discusses the Dialogic® products that can offer complete XML-based server platforms as well as Dialogic products that can be used to build XML communication solution components.

Table of Contents

Introduction

Today, more than at anytime in the past, the world has migrated to an information-driven, always connected world. Emergent communications technologies at the consumer level have largely fueled this trend. Cellular phones and Voice over IP (VoIP) provide real-time access to information, which, when combined with the enhanced use of media for interpersonal communications, increase the complexity of meeting the consumers' needs and expectations. The eXtensible Markup Language (XML)-based protocols presented in this white paper provide a means to develop applications and services that can help keep pace with technology advancements and customers' expectations.

A section is also provided that briefly discusses Dialogic® products that can help address customers' needs for XML-based server platforms as well as for products that can be used to build XML communication solution components.

XML Capabilities and Benefits

The XML protocols based on the web model provide that the same technology and expertise used to develop web applications can be used to develop voice applications. XML protocol markup languages make building communications solutions easier, in the same way HTML simplifies building visual applications. Adopters of the XML-based protocols are no longer locked into rigid, pre-configured communication applications, nor are they locked into a single communications vendor with product-specific APIs. The protocols presented in this white paper abstract the need to know the technical details about the underlying communications products used.

The capabilities provided by the XML protocols discussed in this white paper offer much of the same basic functionality of traditional vendor-specific communication solutions. IVR, fax detection, and multi-party conferencing solutions can be easily implemented using the script-based development environment found in these XML-based protocols. Replicating existing communication solutions is not the goal of these protocols; providing support for expanded media capabilities used in rapidly emerging communications technologies is. XML protocols were designed to simplify the deployment of emerging communication trends ranging from speech recognition and synthesis to multi-party video conferencing.

The benefits offered by implementing XML protocol-based solutions are shortened time to revenue/market and lower development costs because the protocols have been abstracted and have a protocol scripting interface to provide ease of use for protocol development and maintenance for a large population of skilled XML developers.

Authoring Environment

A number of tools are available for developing XML-based communications solutions. These tools range from complete Integrated Development Environments (IDEs) to open source development tools. Commercial development environments are offered as either self-contained development environments or as subscription-based hosted services. Hosted services allow subscribers to use their web-based development tools and XML test environments to verify solutions as they are developed. A majority of the self-contained development environments are operating system independent and can be used with Windows®, Linux, or Sun Solaris operating systems. One development environment vendor provides a graphical tool that provides the ability to rapidly assemble applications using pre-built functional blocks.

Protocol Summary

Each of the protocols summarized in this section represents a significant shift in the development of telephony solutions. The protocols share the common traits of abstracting the "heavy lifting" of vendor-specific APIs away from the developer, and have XML scripting interfaces that make them easy to use.

Voice eXtensible Markup Language

Voice eXtensible Markup Language (VoiceXML) is the language that extends web access to the telephone. Voice recognition and speech technologies used in conjunction with VoiceXML provide voice browser functionality that mimics the traditional web browser. Instead of a mouse, monitor, and keyboard, users interact with a VoiceXML solution by first listening to pre-recorded or computer synthesized speech and then selecting options using spoken commands or pressing keys on their phones (DTMF tones). Like a web browser, a VoiceXML browser can dynamically create content during the course of a call. Dynamic content generation provides a flexible, customizable way to address users' expectations to find information when they need it.

Similarities between voice and web browsers extend beyond their functionality to include their architecture and development environments. Web browsers connect to web servers, loading and compiling HTTP content for display on the browser's screen. Voice browsers connect to the same web servers, downloading and compiling content contained in VoiceXML scripts for audible presentation to callers. Both HTTP pages and VoiceXML scripts use XML formats to create the page or script. This common format was chosen to accelerate the acceptance of VoiceXML among web developers as well as to shorten the time to market for Voice XML-based solutions.

VoiceXML solutions are deployed using a distributed client sever architecture, as shown in Figure 1.
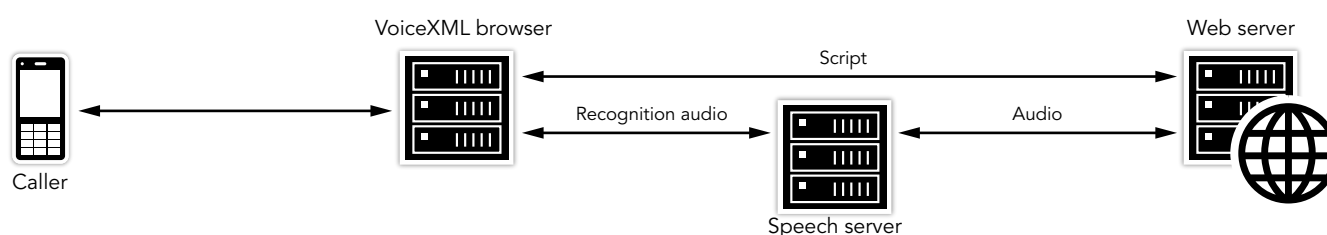

Figure 1. VoiceXML Deployment Environment

In a VoiceXML environment, calls are sent to a VoiceXML browser. The browser uses information associated with the call, for example the number the caller dialed, to obtain the associated VoiceXML script from the web server. This script typically requests that a greeting be played to a caller. The VoiceXML browser would then contact the speech server for the requested greeting. If the recording was a pre-recorded message, the speech server would request the recorded greeting from the web server. The speech server would then stream that audio back to the browser for presentation to the caller. VoiceXML scripts typically prompt the caller for input regarding what the caller wants to do next. The input can be spoken or be in DTMF tones that are sent to the speech server for recognition. A user's input controls the VoiceXML script execution. Multiple user choices can either be supported in a single script or can trigger the loading and running of multiple scripts to serve the caller.

VoiceXML can be used to implement anything from IVR solutions and voicemail solutions to account information retrieval and other self-service applications. In addition to working with incoming calls, VoiceXML browsers can be used for outbound call campaigns, conference applications, receipt of faxes, and even the presentation of videos to callers. VoiceXML however, does not operate in a vacuum, and other protocols presented in this white paper work with VoiceXML to provide added functionality, such as the Media Resource Control Protocol (MRCP). MRCP is responsible for media processing that provides the interface to the speech server.

Media Resource Control Protocol

The Media Resource Control Protocol (MRCP) defines the requests, responses, and events used to control networked media processing resources. MRCP is implemented in a client server model. Applications issue requests for media to an MRCP server, which fulfills these requests. VoiceXML is a sample MRCP client application that uses MRCP to request and receive all media needed. Abstracting the media processing away from VoiceXML frees the protocol from having to know anything about media processing. The supported media include both audio and video, and are streamed from the media server to the client application under the control of MRCP.

Common media resources include Automatic Speech Recognition (ASR) engines and speech engines. The ASR engines are responsible for providing speech recognition services for client applications. Speech is sent to the ASR engine as a stream of Real-time Transport Protocol (RTP) packets where it is compared to previously defined values for a match. Results of the match comparison are sent as MRCP command results to the client application. The match process ends when one of the following occurs:

- A match is found
- A timer expires
- The caller interrupts the session

The speech engine resources are responsible for generating computer synthesized speech or for the playback of pre-recorded audio files. Like the ASR engine, speech engines use RTP packets to deliver the requested audio through the network to the client application. Files played back by the speech engines may be stored locally on the speech engine server or be centrally located on a web server. Centrally locating the pre-recorded media on a web server provides a common source that can be used for both telephony and web server delivery.

MRCP sessions between client and server are established using the Session Initiation Protocol (SIP). As part of this set-up process, a command network port is defined as are network ports to send and receive audio. The command port is used to send MRCP requests for media processing and to receive results for those requests. The audio network ports are used to send audio for recognition and receive streamed audio from the media servers (see Figure 2).
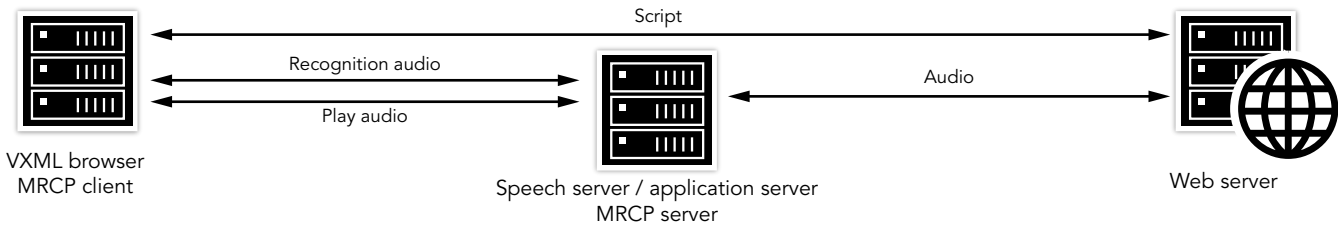

Figure 2. MRCP Interaction between Components

Media Server Markup Language

The current Media Server Markup Language (MSML) specification is a merger of the previous MSML specification and the Media Objects Markup Language (MOML) specification. MSML is used to allow media servers to extend support on SIP implementations to include functionality previously available only in the Media Gateway Control Protocol (MGCP). Like MSCML, MSML provides advanced conferencing and simple IVR functionality using media server resources. MGCP, later renamed to H.248/MEGACO, was the first widely deployed media server control protocol and is still widely used today. The popularity of SIP led to the development of MSML, which is a SIP-based way to control media servers. MSML provides the benefit of simplifying network protocols in use by using SIP to control media sessions.

MSML was defined to provide more general-purpose media server control that was more extensible than the MSCML specification provided. The MSML specification defines a strong framework that allows for extensions that ensure that the protocol can easily change to meet changing media needs. MSML is used to control media streams and conferencing resources. With the inclusion of the MOML specification, MSML is also able to define complex media processing objects that can be used on media streams and conference sessions. MSML solutions may be deployed alone or in tandem with VoiceXML to serve a wide range of application requirements. Used alone, MSML provides conferencing features such as side bar conferences, advanced audio mixing, and active speaker notification. Additionally, MSML can provide gain control for sessions controlled by either media or application servers. Gain can be adjusted using either DTMF keys or speech recognition services.

When implemented with VoiceXML, MSML's user interaction functionality is expanded. Its features include playing announcements to conference attendees, controlling record/playback, and digit and speech recognition. The power of the MSML protocol is its flexibility to implement highly tailored user interactions. Additionally, MSML provides support for mid-call interactions between the media server and application. Examples of mid-call interaction include gain control requests and conference attendee roll call reporting.

MSML was proposed as a standard after the MSCML protocol was published for review and is in some ways a competing implementation. The MSML standard provides all of the MSCML functionality, but where MSCML is primarily focused on conferencing, MSML provides expanded media handling support.

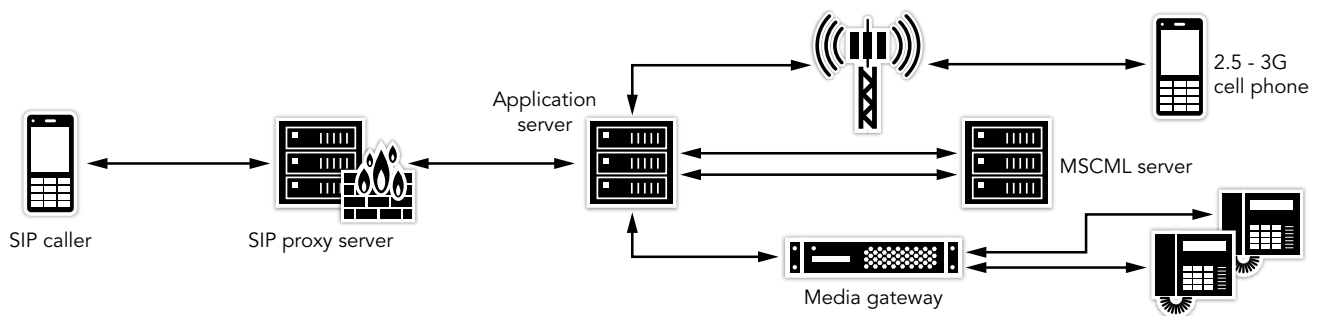Figure 3 is an example of an MSML implementation.

Figure 3. MSML Implementation

XML Implementation Considerations

Some of the notable considerations to review before implementing XML-based solutions are:

Skills Availability — The skills needed to build an XML-based communication solution are more than likely available today in a company's Information Technology (IT) department. Developers who can develop web services most likely can develop communication services.

Cost of Providing Differentiating Services — The largest cost associated with implementing an XML-based communications architecture tends be the cost of a skilled communications engineer to define a migration strategy and overall architecture. In addition, a human interface engineer is needed such that the interface is both easy to use and meets the needs of the services consumer.

Quality, Reliability, and Responsiveness of Services — A well designed XML communications solution provides a consistent quality of experience to its users. The use of Natural language processing ensures both the quality and responsiveness of the XML solution. Natural language interfaces permit the user to tailor the user experience based on phrases, and not fixed scripts and responses.

Consistency of Services Across Multiple Communication Channels — The infrastructure used to support XML-based communications solutions can largely be hosted on the same infrastructure used to support web-based access. For example, an existing web server can be used to serve media to both telephony and web users. Additionally, the application server's role can be expanded to provide call control.

Implementation of XML communication solutions is not without its challenges.

A notable challenge is security and safeguarding information used in an XML solution, which results from the highly distributed architecture and tight coupling of communication solutions to corporate data to provide increased customer service.

Another important challenge is developing a common user experience that will provide the same functionality across multiple user endpoints. The lack of standards that define the capabilities of end point devices increases application challenges. Developers will need to account for device capability differences to ensure maximum multi-device capability when delivering services.

Using Dialogic® Building Blocks

Dialogic offers complete XML-based server platforms as well as products that can be used to build XML communication solution components.

Dialogic® PowerMedia XMS

Dialogic's PowerMedia XMS is a highly scalable, software-only media server that enables standards-based, real- time multimedia communications solutions for IMS, MRF, Enterprise, and WebRTC applications on premise or in the cloud. Built on 15+ years of software media processing experience, PowerMedia XMS is trusted by world-class service providers and large enterprises to power millions of rich media sessions. The following URL has more product information: https://www.dialogic.com/en/products/media-server-software/xms.aspx

Summary

The adoption of XML-based communication solutions is growing as more businesses embrace the technology as a differentiator. Although the technologies represent a significant shift away from traditional communication solutions, they do not present an insurmountable hurdle for those wanting to adopt the XML communication technologies. Dialogic is well positioned to provide the components, both in servers and building blocks, that can enable businesses to implement these dynamic communication solutions.

For More Information

Find the latest information about the Dialogic products mentioned in this paper by visiting the Dialogic website at www.dialogic.com. Want to hear the latest on what's going on in the telecom industry to keep you one step ahead? Subscribe to the Dialogic Blog to receive great insights and advice directly to your inbox.

Dialogic

www.dialogic.com