Dialogos Voice Platform

Product Datasheet

Dialogos Speech Communication Systems S.A.

September 2007

Contents

1 Dialogos Voice Platform

Dialogos Voice Platform is an open, standards-based VoiceXML and CCXML platform that is optimized for the development, deployment, hosting and operation of voice automation applications.

Dialogos Voice Platform offers support for the latest speech technologies and application design tools. Instead of relying on dated speech technologies, application development tools created for touch-tone – not speech – applications, and require investment in costly proprietary infrastructure, Dialogos Voice Platform's open, standards-based architecture allows customers to leverage their investments in highly scalable IT infrastructure.

1.1 DVP features

1.1.1 Standards-based Architecture

Dialogos Voice Platform supports VoiceXML v2.0[1] and its v2.1[2] extensions, along with CCXML and the Speech Recognition Grammar Specification – the open, industry standards for speech applications and services developed through broad industry participation in an open forum managed by the World Wide Web Consortium.

With respect to voice application development, and since VoiceXML is based on Web architecture, businesses deploying voice services on Dialogos Voice Platform can leverage their existing expertise and investment in Web infrastructure such as application servers, business logic and rules, and customer and corporate databases. Not only does this reduce investment requirements in new infrastructure, but the familiarity of Web architectures also reduces costs associated with training IT staff and implementation time.

1.1.2 Optimized for speech

Dialogos Voice Platform is uniquely optimized for speech solutions with a tightly integrated application design, development and run-time environment, speech centric reporting and system administration capabilities and advanced speech technologies, designed to increase automation rates, improve caller satisfaction and maximize ROI.

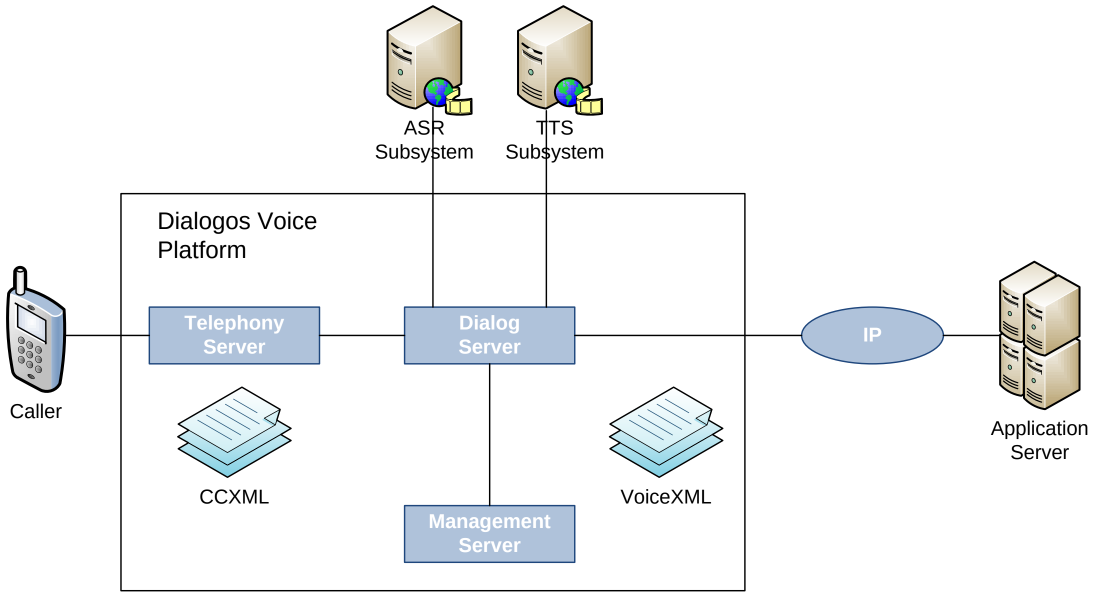
1.2 DVP architecture

Dialogos Voice Platform (DVP) comprises of four (4) basic subsystems:

- telephony server
- dialog server
- management server

[1] Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C Recommendation 16 March 2004 (http://www.w3.org/TR/voicexml20/)

[2] Voice Extensible Markup Language (VoiceXML) 2.1, W3C Recommendation 19 June 2007 (http://www.w3.org/TR/voicexml21/)



DVP exposes a series of APIs that can be used for developing applications and interfaces to an extensive set of third-party subsystems.

1.2.1 Telephony Server

The telephony server is an independent software subsystem, which controls the operation of the telephony hardware. One or more application servers communicate with the telephony server over a TCP/IP data network, and receive notifications for a set of events, such as:

- Incoming call
- Call completion
- Call transfer
- Start/Finish audio playback
- Start/Finish audio recording
- DTMF usage

The telephony server supports the following protocols:

- T1 / E1
- ISDN PRI
- SS7
- VoIP (SIP/RTP)
- Analog

The telephony server is connected to the telephone network. This connection is handled either through inexpensive, off-the-shelf telephony network interface cards (supporting T1, E1 and analog networks) or a software subsystem (supporting VoIP networks). DVP's distributed architecture allows the simultaneous management of a large number of telephony channels (120 channels per card on an E1 network). The telephony server supports multiple telephony cards on the same server, allowing for higher call densities.

The telephony server also provides support for conferencing. The conferencing feature is available both in software and in hardware-accelerated modes. Software-only conferencing support takes advantage of the telephony server's sophisticated signal summing and echo cancelation algorithms, in order to provide a high-quality user experience. The number of participants taking part in a conference can be extended even further by utilizing the hardware acceleration offered by certain telephony cards, allowing for the creation even of extremely high-density conferences.

The telephony server includes a CCXML interpreter, which can be used to control the underlying, incoming or outgoing, telephony connections, and the voice applications associated with them. By fully supporting the latest Working Draft of the Voice Browser Call Control specification[3] the telephony server allows the voice automation solution developer to:

- provide application support for multi-party conferencing, with advanced conference and audio control
- handle and control multiple calls, using sophisticated, application-dependent logic
- handle incoming and place outgoing calls, that can be associated with an automated voice application

1.2.2 Dialog Server

The dialog server acts as a shell for all dialog units that support end-user interaction. The telephony server forwards all telephony events to the dialog server. The dialog server associates every active call with a dialog unit that handles interaction with the end user. Upon call termination, the dialog unit is freed up and can be reused in order to serve a new incoming call.

All dialog units are based on open standards, facilitating interconnection with third party systems, and allowing reuse of any pre-existing infrastructure. Some of the standards supported by DVP are the following:

- Network communication: TCP/IP, HTTP, HTTPS, POP3, SMTP, RTP/RTCP
- Database communication: SQL, ODBC
- Information management: UNICODE 3.0, XML 1.0, DOM 3.0, SAX 2.1
- Dialog management: VoiceXML 2.0 & 2.1, Speech Recognition Grammar 1.0

The conversation server includes a VoiceXML interpreter, which can be integrated with the industry's leading speech recognition, voice authentication and text to speech synthesis software. Using standard Internet protocols, the application server fetches VoiceXML applications generated dynamically by external application servers that employ JSP, Java servlets, ASP.Net, PHP or equivalent server-side technologies.

The dialog server controls audio input (recording) and output (playback) through the voice platform's active telephony lines. It is also responsible for the platform's interconnection

[3] Voice Browser Call Control: CCXML Version 1.0, W3C Working Draft 19 January 2007 (http://www.w3.org/TR/2007/WD-ccxml-20070119/)

with speech recognition and/or synthesis systems. Subsystems providing support for Nuance Recognition server and Nuance RealSpeak synthesis server, allow

The audio playback subsystem supports a wide variety of codecs, allowing the decoding and subsequent playback of almost any audio format, regardless of the underlying hardware capabilities. Some of the supported codecs are the following:

- PCM 16-bit / 8-bit
- CCITT a-Law
- CCITT u-Law
- OKI (Dialogic) ADPCM
- IMA ADPCM
- Microsoft ADPCM
- GSM 6.10
- MPEG-1 layer 1, 2 & 3

Furthermore, advanced re-sampling algorithms allow the playback of audio files recorded with sampling rates, ranging between 6kHz and 48kHz.

1.2.3 Management Subsystem

The management subsystem handles the configuration, initialization, monitoring and termination of all of DVP's subsystems.

The management subsystem can be accessed through a web-based graphical user interface. System administrators and operators can use the graphical Management Console to efficiently manage and maintain all aspects of their speech systems to help ensure high service availability.

Access to the management console can be restricted. It is possible to access the console either locally, from the server hosting DVP, or over the network, from a pre-specified list of workstations that will be granted access to it.

1.3 Application Development

Dialogos Voice Platform offers a rich set of application programming interfaces (APIs), each one providing a different level of flexibility. The developer is free to choose his API of choice, depending on the requirements, the complexity and the peculiarities of each application.

1.3.1 Native C++ API

The Native C++ API provides the highest possible flexibility compared to any other supplied API. User interaction is modeled after the object oriented features of the C++ programming language.

Applications are decomposed into a set of appropriately configured interaction components. A transition matrix identifies the component that will handle user interaction, depending on the current state of the application. Interaction components may be further decomposed into less complicated ones, providing a hierarchical dialogue organization and great opportunities for reusability within the same or other applications.

In order to reduce the development cycle of new applications, Dialogos Voice Platform provides a set of ready-to-use common dialogue interaction components, which can easily be configured and then combined to form fully functional voice or DTMF driven applications.

1.3.2 JavaScript API

The JavaScript API allows the configuration and execution of any component or application that was developed, through an interactive shell. These applications may have been developed using any combination of the following APIs:

- Native C++ API
- Java v1.4
- JavaScript v1.5

Components developed using the Native C++ API or the Java programming language, can be transparently reflected into the common runtime environment of the Dialogos Voice Platform, and then manipulated through a high-level scripting language such as JavaScript.

Developing simple applications may be as simple as typing a few lines of JavaScript code. Applications with a higher degree of complexity may take full advantage of the ready-to-use Native C++ common dialogue components, or even invoke the VoiceXML interpreter.

1.3.3 VoiceXML v2.0 / v2.1

VoiceXML is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of Web-based development and content delivery to interactive voice response applications.

VoiceXML is a markup language that:

- Minimizes client/server interactions by specifying multiple interactions per document.
- Shields application authors from low-level, and platform-specific details.
- Separates user interaction code (in VoiceXML) from service logic (e.g. CGI scripts).
- Promotes service portability across implementation platforms. VoiceXML is a common language for content providers, tool providers, and platform providers.
- Is easy to use for simple interactions, and yet provides language features to support complex dialogs.
- Reuses existing back-end infrastructure.

DVP’s VoiceXML interpreter component is fully compliant with the W3C Recommendations for the VoiceXML v2.0 and v2.1 standards. The VoiceXML Interpreter is provided in the form of a Native C++ interaction component. This component may be used in stand-alone configurations or as part of larger and more complex applications.

While the VoiceXML standard does allow for a variety of compelling applications, there may be certain specific capabilities or functions that are not yet available through the standard. Dialogos Voice Platform offers the ability to invoke through the VoiceXML runtime environment components implemented using the Native C++ API or the Java programming language. This way, complex operations can be performed locally, without the necessity to contact an application server.

1.3.4 CCXML v1.0

CCXML provides declarative markup to describe telephony call control. CCXML is a language that can be used with a dialog system such as VoiceXML. CCXML can provide a complete telephony service application, comprised of Web server CGI compliant application logic, one or more CCXML documents to declare and perform call control actions, and to control one or more dialog applications that perform user media interactions.

CCXML offers an abstract layer for call control, independent from the underlying telephony system and completely “unaware” of the protocols used for the signalling and transport of voice calls. CCXML is the key factor for new cost-effective integrated voice services platforms easing the convergence process from traditional telephony (TDM) to voice-over-IP (VoIP), as it provides the same level of service to both networks.

Independence from telephony is accompanied by independence from the developing environment, as XML does not require specific tools and compilers, it is widely acknowledged and can be run on any platform. CCXML applications are fast and easy to develop, portable onto any platform and can be run by any CCXML compliant interpreter.

Combining the advantages of standardization and of an XML web-based approach, CCXML is becoming a de facto requirement because it allows portability and reusability, maintainability and cross-vendor interoperability, eliminating any awkward call control customization associated to specific vendors.

1.4 Connectivity with External Subsystems

Internal communication between Dialogos Voice Platform’s subsystems, as well as external interconnection with third-party systems, is made possible using the TCP/IP network protocol. In the case where enhanced security communications are required, TCP/IP can be combined with SSL to meet such requirements.

For instance, DVP connects to Application and Web Servers using HTTP/HTTPS protocol, and to email servers using the SMTP and POP3 protocols to receive and send email, respectively. DVP also provides interfaces for connecting to SMS service centers, for sending and receiving SMS messages.

1.4.1 Database Management Systems (DBMS)

Dialogos Voice Platform has adopted the open database connectivity standard (ODBC), offering support for a wide range of database management systems, including:

- Oracle Database
- Microsoft SQL Server
- IBM DB2
- Sybase
- MySQL (open source)

Database support is made possible through a set of Native C++ and JavaScript objects. In addition to these interfaces, the Java common runtime environment also offers implicit support for the Java database connectivity standard (JDBC).

Finally, the inherent client-server architecture of the VoiceXML standard provides access to database management systems through application servers that employ JSP (servlets), ASP, PHP or similar server-side technologies.

1.4.2 Automatic Speech Recognition Systems (ASR)

Dialogos Voice Platform supports speech recognition interfaces to Nuance Communications' Speech Recognition engine, covering at least twenty (20) languages and many more dialects. Some of the statistical recognition models offered by Nuance, such as those for the Greek, Turkish, Arabic and the Catalan language, have been developed and are supported by Dialogos.

Dialogos Voice Platform also exposes a series of interfaces that can be extended to support other, third-party speech recognition engines, which are currently not supported.

1.4.3 Speech Synthesis Systems (TTS)

Dialogos Voice Platform can interface with external audio sources either through a TCP/IP network or through the SAPI 5.1 standard. These interfaces are also exposed by most new generation speech synthesis products, such as Nuance's RealSpeak, rVoice and Vocalizer speech synthesizers. DVP's audio subsystem also allows mixing pre-recorded and dynamic (synthesized) prompts in a transparent way.

Nuance's speech synthesizers (RealSpeak, rVoice and Vocalizer) incorporate the latest technologies in the field of speech synthesis, offering large densities and producing almost natural sounding synthesized speech. More than 20 languages are supported, including English, German and Russian.

1.4.4 Computer-Telephony Integration Systems (CTI)

Dialogos Voice Platform, having been tailored for the needs of large call centers, supports all necessary interfaces for interconnection with a Computer/Telephony Integration server. Using this technology, calls that cannot be fully serviced by the automated system, can be forwarded to a live agent, who will continue the conversation with the client from the point where the system left off, as all information that had been gathered by that time would be available on the agent's screen.

Dialogos Voice Platform already supports one of the most widely deployed CTI systems, the Genesys T-Server.

1.4.5 Other Information Sources

Supporting the latest draft of the VoiceXML v2.1 standard, DVP offers the option of using the `<data>` tag for direct information exchange between the client (VoiceXML browser) and the server (data source) using XML messages. XML messages are processed on the VoiceXML browser, using JavaScript and the DOM, allowing the use of several "AJAX-like" programming paradigms.

This technique minimizes communication with application servers, retains dialog progress at the client, and greatly simplifies the structure of a large family of applications.

2 Nuance Speech Recognition

Nuance 8.5 is the latest release of the market's leading speech recognition software optimized to enable highly accurate, scalable, and easy to deploy voice solutions. Nuance-powered voice systems allow end users to enjoy faster and more efficient phone-based interactions, while enabling carriers and enterprises to improve customer satisfaction and lower costs.

Among the key features and benefits of Nuance's Speech Recognizer are the following:

- Proven performance with deployed accuracy measurements as high as 97 percent
- Large vocabulary speaker independent recognition capable of handling over 100 million listings
- Exceptionally high accuracy in noisy environments – ideal for wireless and hands-free operation
- Seamlessly integrated voice recognition, verification and text-to-speech for improved performance
- High port density for efficient use of system CPU and memory resources
- OA&M framework and SNMP support for easy integration with off-the-shelf OA&M, billing and provisioning systems
- Grammar Hotswap enables online changes to the user interface allowing 24/7 availability
- Barge-in allows users to interrupt application prompts in midstream
- Dynamic Grammar Recognition enables applications involving user-defined, voice-enrolled, or text-enrolled grammars (for example, Voice Activated Dialing)
- Hotword recognition allows the application to "listen" for specific control words
- Statistical Language Models that support natural language interaction (Say Anything™ & AccuRoute™)
- Dynamic language detection for multilingual dialog systems (Dynamic Language Detection™)
- Support for 28 languages allowing for deployment in local markets around the world
- Host Media Processing (HMP) support for Intel Dialogic telephony cards

2.1 Features

2.1.1 High Accuracy Recognition in Noisy Environments

Nuance Communications' speech recognition system has the largest number of active installations than any other competitive recognizer. Among Nuance Communications' clients one can find large banks and investment companies such as Lloyds TSB Bank, Charles Schwab and Fidelity Investments. Nuance Communications' system achieves high recognition accuracy, reaching up to 97%. Special techniques, incorporated into Nuance's system, allow it to achieve high recognition accuracy results even in difficult environments, such as mobile telephony, conferencing, hands-free headsets, etc. The fact that most mobile operators have chosen Nuance's speech recognizer, testifies to the system's capabilities. A short list of such companies includes Sprint PCS, Quest Wireless, AT&T Wireless, BellSouth, Japan Telecom, Telia Mobile, Vodafone, Cosmote, and others.

2.1.2 Improved User barge-in Options

Nuance Communications' speech recognition system supports the option of barge-in, that is, of interrupting application prompts before they are completed, by directly speaking their commands. This option greatly accelerates transactions with the voice system, and makes user communication with the system more natural. Barge-in becomes even more useful as time passes, since users get accustomed to the system and learn how to respond to its prompts. Nuance Communications' system has the largest number of speech recognition systems installations with barge-in capabilities.

2.1.3 Natural Language Recognition and Understanding

With Say Anything and AccuRoute, callers can speak more naturally when interacting with voice-enabled applications. Systems can greet callers with open-ended prompts like "what would you like to do?" and users can respond with wide ranging answers such as "I have a problem with my last bill and I need some help" or "I'd like to, um, get my, um, bank balance."

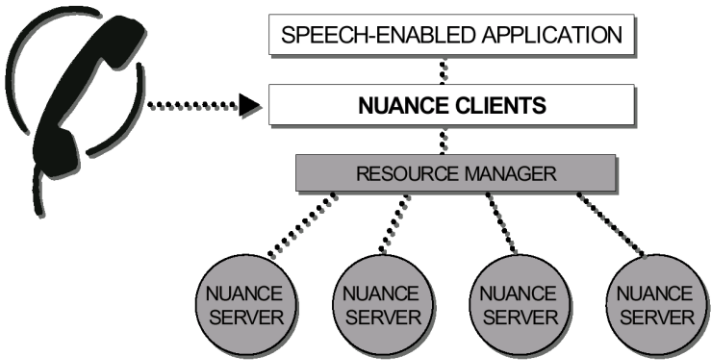
These technologies are critical for call steering applications and are designed to significantly lower the cost of customer care as well as to simplify and improve the caller experience. Large enterprise and telecommunications solutions leveraging Nuance Say Anything and AccuRoute can reduce misroutes by up to 50% and cut "zero outs" from 10-35% resulting in potential savings of up to $20 million annually. Nuance's Say Anything and AccuRoute deployments handle over 1/4 of a billion calls per year, significantly more than technologies provided by other vendors. Nuance is the only major speech vendor with significant deployment experience with these advanced natural language systems.

2.2 Deployment options

Nuance 8.5's patented distributed architecture provides scalable operation while minimizing hardware requirements and costs. Moreover, it is proven — Nuance's patented architecture has been deployed by hundreds of customer over the last 8 years. It supports simultaneous load balancing and fault tolerance across speech recognition, verification and text-to-speech operations, to ensure efficient utilization of system resources that can result in hardware savings of over 25% compared to other systems. Furthermore, the load can be distributed on a per-utterance basis so if a server goes down, the caller will not be lost, but will simply be asked to restate only the last utterance.

The architecture utilizes three primary components:

- Recognition Server
- Recognition Client
- Resource Manager



Each Recognition Server runs both recognition and verification simultaneously. The Resource Manager balances the load across all the Recognition Servers, thereby ensuring efficient hardware utilization. CPU-intensive recognition can then be performed on a machine other than the machine running the application and audio interface.

This stateless client/server architecture also allows tuning and maintenance of servers while the system is still online, making sure that you never lose customer calls. The modular approach of the Nuance architecture enables the separation of light client processing from CPU-intensive server processing. This allows for a high port density on the client side and efficient CPU usage on the server side, optimizing the use of memory and CPU resources.