



Traditional IVR and Visual IVR – Killing Two Birds with One Stone

Dmitry Sityaev, Praphul Kumar, Rajesh Ramchander

Genesys, United States of America

dmitry.sityaev@genesys.com, praphul.kumar@genesys.com, rajesh.ramchander@genesys.com

Abstract

This paper describes a novel solution which allows to quickly build and develop multi-channel applications. Due to the popularity of a smartphone, a new paradigm of applications called Visual IVR has been emerging recently, where visual navigation replaces the traditional DTMF or voice-enabled dialogue control. The described solution brings a unified approach for the creation of traditional IVR systems as well as Visual IVR systems.

Index Terms: IVR, Visual IVR, multi-channel dialogue systems, SaaS

1. Introduction

Interactive Voice Responses (IVR) systems have been around for decades now. Historically, IVR solutions used voice prompts and menus to present information and options to callers, and touch-tone (DTMF) to gather responses. Modern days IVR solutions also enable input and responses to be gathered via spoken words using voice recognition (ASR). Users can access their bank balances, flight schedules, gain information about product details, order status, movie show times and so on from any telephone. Companies have been using IVR to automate their calls as well as collect some basic information from callers to help them get routed to the right agent.

The proliferation of mobile technology has changed the way we interact with the world. We (as users) expect to be able to receive service and handle our requests any time, anywhere and from any device. This forces companies to change the way they interact with their end users too: not only are companies expected to provide service 24/7, they are also expected to communicate with their end users across various channels and provide solutions that cater to the multi-tasking behavior of the end user.

Visual IVR is a new breed of solutions that extends the traditional IVR experience to a variety of experiences ranging from multi-channel to omni-channel and multi-modal. At the core lies an instant interaction over a smart device (mobile phone, tablet, etc.) that is driven by visual navigation. End users are presented with a choice of items or a form to fill on their mobile device, which unlike a traditional IVR is faster, leads to increased accuracy and enhanced customer experience: customers are no longer hampered by voice recognition errors, long winded menus and options, and inability to go back to any previous point of their interaction with the system.

The user is thus presented with an option to manage their interaction across both channels – Traditional IVR and Visual IVR. An application developer can decide which channel

would be optimal for solving a specific problem and direct the user to that channel without losing any context.

The benefits associated with the use of Visual IVR over Traditional IVR are numerous, however, building a Visual IVR from scratch can be a challenge. This paper presents a solution that is aimed at creating both Traditional and Visual IVR applications within minutes. Moreover, the applications created are driven by the same flow and fully cloud-based.

Section 2 provides an overview of the solution architecture. Section 3 outlines further directions for research and development. Section 4 summarizes the key points of this paper.

2. High-level architecture

The high-level solution architecture is presented in Figure 1 below. At the heart of the solution lies a tool called CX Builder. CX Builder is a web-based on-demand tool that allows users to build the application flow, upload prompts, create recognition grammars (both speech and DTMF), facilitate transactions with other systems over RESTful APIs, and so on. An example of what a flow built with the CX Builder looks like is shown in Figure 2.

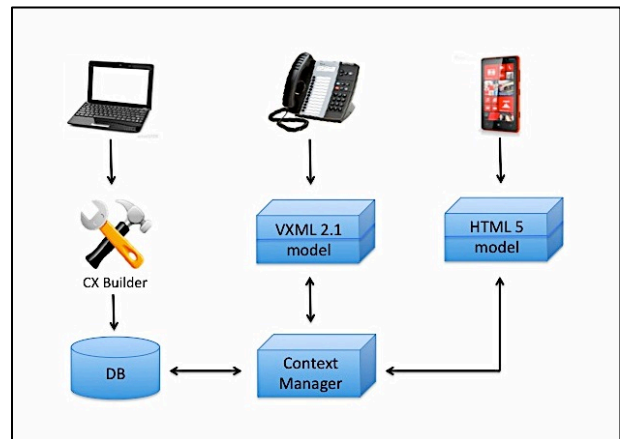


Figure 1: High-level solution overview.

Once the application flow has been built in the CX Builder, its representation is pushed into a database and stored there. When a call is placed, the Context Manager (which has context awareness) determines the platform needed to execute the request and then routes the call to a voice platform. A special component is used to fetch the application representation from a database and transform it to a series of VXML pages dynamically. These pages are written using the VXML 2.1 formalism [1] and are the voice platform is used to interpret and execute the VXML, thus driving the call flow.

With the Visual IVR, the request could be served through an HTML5 client that knows how to translate an application into a HTML5 page. An interaction could also start with the push notification of a URL for the first page of the Visual IVR. Similarly to the Traditional IVR, there is a special component used to dynamically fetch the application representation from a database and render it into HTML format. HTML5 formalism [2] is used to represent the content of each page, and an Internet browser is used to move from one page to the next within the Visual IVR – no application download is necessary. Figure 3 below gives an example of what a page in a Visual IVR may look like.

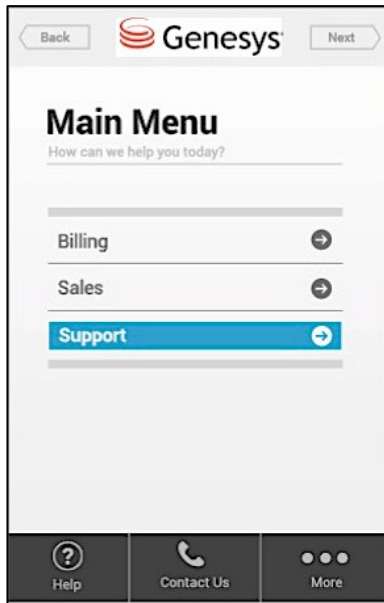


Figure 3: An example of a Visual IVR page.

It is extremely easy to make any updates or changes to the application using the CX Builder tool. Moreover, one has to make changes or updates only once in order for them to be instantly available in both Traditional and Visual IVR. The speed and ease of making changes to the application is a distinctive feature of the proposed solution.

3. Future work

Future work will be centered on making the Visual IVR more user friendly. For example, research is currently under way whether adding Text-to-Speech (TTS) for the playback of the initial prompt associated with a page has a positive effect on the user experience. Work is also currently under way on creating a new look and feel for the Visual IVR.

4. Conclusion

This paper presented a new solution of how one development tool, CX Builder, can be used for accomplishing two tasks: creation of Traditional IVR (DTMF and/or speech enabled) and Visual IVR. Although there is no standard definition of what Visual IVR entails, an attempt has been made to create a version of a Visual IVR. Much work remains to be done around carrying out usability studies for the Visual IVR.

5. References

- [1] Ohsry, M. et al., “Voice Extensible Markup Language (VoiceXML) 2.1”, W3C Recommendation, March 2007; <http://www.w3.org/TR/voicexml21>.
- [2] Hickson I. et al., “HTML5 A Vocabulary an Associated APIs for HTML and XHTML”, W3C Recommendation, October 2014; <http://www.w3.org/TR/html5>.

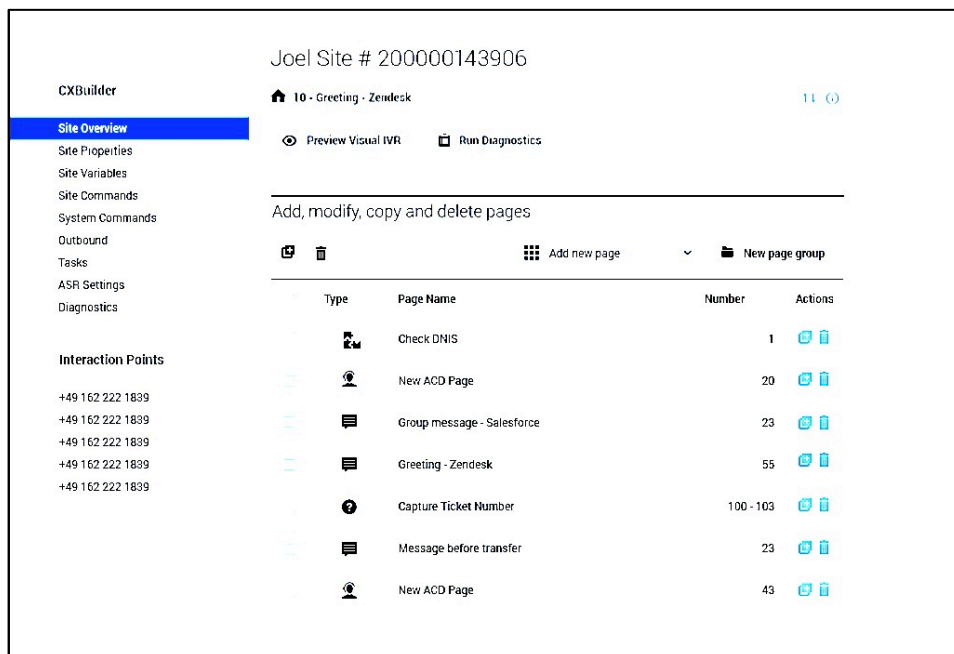


Figure 2: An example of an application flow in CX Builder.