

Excerpt from the April's edition of the „Speech Strategy News“ 2011

## VUI Visions

# Personalizing Self Service in Voice Portals and Apps

---

Jörn Kreutel and Lupo Pape, SemanticEdge GmbH

*In this guest column, we ask designers skilled in creating Voice User Interfaces to highlight a particular aspect of VUI design inspired by actual deployments. In this issue, Dr. Jörn Kreutel (co-Founder and CTO) and Dr. Lupo Pape (Managing Director, co-Founder, and CEO), **SemanticEdge**, note that personalization and “expressive interaction” can be a key to handling multiple channels of customer service efficiently and effectively with a common architecture. Dr. Kreutel is responsible for the architecture of the SemanticEdge AIM Framework, as well as for the design and the development of voice applications built on top of it. Before the foundation of SemanticEdge, he was employed by Lernout & Hauspie in Barcelona. Dr. Kreutel graduated in Linguistics and Aesthetics at Humboldt University in Berlin. He holds an MSc in Speech and Language Processing from the University of Edinburgh and a PhD in Computer Science from the University of Potsdam. Prior to the foundation of SemanticEdge, Dr. Pape worked as a Project Manager at McKinsey & Company. Dr. Pape received a doctorate in the areas of Metallurgy and Environmental Technology from the Technical University of Berlin and studied Economics at the Université de Fribourg in Switzerland and at the Universidad de Barcelona, Spain.*

Although web-self-service has been promoted for years now, telephone customer service is still the main avenue of access to organizations with a large customer base. Additionally, with the explosion of smartphone and app sales a new dimension is opened for the management of customer relations. Customers may now potentially be addressed anytime and anywhere. Instead of treating the app-channel separately, it makes a lot of sense to develop a multi-channel customer service strategy and a joint infrastructure for the management of an enterprise’s self-service channels like voice portals, apps, and the personalized services on the company’s website. Personalization is the key innovation driver in these channels. Not only will personalized multichannel self-service raise customer service to the next level, it will also significantly support the marketing of additional products and services. In order to implement such services, a new expressive interaction management infrastructure is required that is able to handle different customer service channels and to personalize the service offering.

Our article will first outline the main tendencies that we observe on the market for VUIs and Apps, respectively, and will then sketch a scenario for multi channel customer service. We will then briefly describe the main aspects of a software architecture that is able to cope with those business requirements.

## A new quality of demand for voice portals and challenges for VUI-development

The demand for speech dialog systems is picking up and more industries like insurance companies or power suppliers are looking for ways to optimize their customer service operations and to cut telephonyrelated costs. Another reason for the increasing demand is a strong renewal cycle of out-dated touch-tone systems. The significant penetration of smartphones with touch-screens and cordless handsets is heavily questioning the usability of touch-tone-systems.

Instead of just adding voice to their IVR menus, companies are showing strong interest in open natural language dialog strategies. And, given the companies investments in CRM systems in the past years, more customer information can and will be used for better customer service in the human contact as well as in the self-service channels. In every industry, a large proportion of the customer service calls

results from past activities of the customer or the company or could be derived from the customer status.

After the 2008/2009 economic crisis, the renaissance of the demand for voice portals or portal upgrades comes with a new kind of demand, challenging the traditional development approaches. "How-may-I-helpyou?"- style dialogs including highly personalized dialog strategies can't be implemented by using plain VoiceXML-code or by applying the common tools of IVR-vendors, which usually lack the required expressiveness.

In order to ideally treat every caller differently depending on the caller's profile and their past contacts, as well as to accommodate the strategies of customer service and marketing departments, applications have to be highly flexible and configurable. A new architecture for applications and a new development approach are required.

***"...applications have to be highly flexible and configurable. A new architecture for applications and a new development approach are required."***

### **The App-Phenomenon and new challenges for App-development approaches**

Starting with the emergence of the iPhone, another trend heavily affects companies' channel and selfservice strategy, *the App-phenomenon*. In some industries and with some innovative apps, mobile applications are already changing the way companies interact with their customers.

Not surprisingly, the services offered, e.g., by banks and carriers in their first generation of apps are the same they had been addressing in their voice portals, when they were launched five to ten years ago. These are its standard services that are required to be usable anywhere and anytime.

Now apps face similar problems as voice portals: Given the limited screen size only a selection of services is offered, which raises similar challenges like the menu structures in voice portals. The wider and the deeper the service spectrum in the IVR menu gets, the higher the frustration and drop-out of callers. Another important issue of the apps is the fragmentation of the operating systems.

As it is required for the next generation of voice portals, personalization is key to mobile customer service apps in order to tailor the service and marketing offering to the specific customer's needs. Given the overlap of the service range, why not try to establish a common ground for both channels? The personalization of app-content faces the same technical challenge as the development of personalized voice applications.

### **Multi-channel use-cases and multi-channel management challenges**

In fact, mobile apps do not necessarily overlap or cannibalize the voice channel. Both channels have their specific usage scenarios. In many cases, cross-channel process design will find new synergies between the "old" voice world and the new app world.

A personalized up-selling recommendation, e.g., for a specific data/SMS/voice option, formulated appropriately in both channels of a carrier and encountered by the customer in the app or in the IVR before being routed to an agent, will significantly increase the awareness and the probability of success. In the same way, important status information can be displayed in the start page of a mobile care (mCare) app, while at the same time being stated in the IVR.

In summary, given the two channels' similarities and synergies, it makes a lot of sense to cover mobile apps and voice portals with a joint service design approach. The synergies are more obvious when natural language voice input becomes an integral part of a mobile app. The already created and tuned "How-may-I-help-you-Grammar" of the IVR may lay the foundation for recognizing users' intents within the mCare app.

## Advanced Interaction Management for personalized and multi-channel service design

SemanticEdge's *Advanced Interaction Management Framework* (AIM) has been designed to cope with the challenges posed by the requirements of a personalized and multi-channel customer interaction strategy that comprises both VUIs and Apps, as well as other channels such as the personalized segment of a website and, as outbound channels, email and SMS. The AIM framework uses an innovative model-based approach to interactive application development and provides abstraction levels to developers, which are relevant in two ways:

- On the one hand, the AIM framework maximizes the usage of shared resources for VUIs and Apps at the level of interaction control. Not only will the mere backend layer of customer interaction channels be shared, but also the business logic and resources required for the provision of a personalized customer experience and the validation of input data are consolidated over multiple channels.
- The second aspect of abstraction supported by the AIM framework is an abstraction over the particular runtime platforms at which an application is deployed. For VUIs, this will be a mere server-based solution. For Apps, both a server-driven strategy and a hybrid approach are supported, where the latter places backend-intensive processing and reasoning over personalization on the server and the rest of the interaction control on the handset itself.

The expressive means for interaction control provided by the AIM framework are further enhanced by generic configuration elements. They allow an interactive application to be developed as a reusable "application template," open for multiple configurations and customizations. Configuration elements comprise simple discrete or open-valued parameters, table structures, as well as calendar or business rules.

Using these configuration interfaces, a marketing department can, for example, create new customer profiles and change targeted product and service offerings in all apps (iPhone, iPad, Android smartphones, and tablets) at once, without the user having to download an update of the app in the app store. At the same time, the new profile and offering may appear in the voice portal and on the website after the user has been identified. The AIM framework, hence, allows realizing a maximum level of personalized user experience while minimizing the required development efforts.

The functionality of the AIM framework is put into practice by the following components:

1. The AIM SCE is an Eclipse-based Service Creation Environment for efficient application development. The SCE allows an interaction designer to design the call flow of a VUI, as well as the page flow of an App, using an intuitive graphical representation. The latter adheres to the principles of SCXML (State Chart XML, "State Machine Notation for Control Abstraction," [www.w3.org/TR/scxml](http://www.w3.org/TR/scxml)) as a standardized modelling language for interactive application development that is positioned by the W3C as a successor of VoiceXML. Both call flows and page flows may reuse the same components for the validation of user input, the execution of backend functions, and the personalization of the user interface. Also the configuration parameters for an application are shared between the modality-specific components. Given the model of an application, the SCE is able to generate code that is executable at runtime. This is java code for server-side components and platform-specific code for the supported platforms (Android, iPhone, and iPad).
2. The AIM Management Console provides extensive functionality for managing deployed applications in an agile and flexible manner. It supports the deployment and update of server-side web applications that have been created with the SCE, and comes with GUIs for setting and modifying the configuration parameters that have been created by the application developer. Most importantly, it allows the creation of customized configuration settings for different customer profiles or different brands or tenants based on a single application template. In addition to these content management facilities, it includes a reporting component that allows

the creation of consolidated analytics over different interaction channels.

3. The AIM Runtime Engine is a powerful and yet lightweight set of highly scalable components for controlling the interaction behaviour of applications developed in the AIM SCE. It consists of shared server-side libraries used for the call flow and page flow control of VUIs, GUIs, and Apps, respectively, and of particular client-side libraries for the different handset platforms supported by the AIM framework. At its core, it employs a “dialog manager” for flexibly handling user input in a way that is agnostic with regard to whether input is coming from a DTMF Application, a VUI with natural language capabilities, or from the GUI of an App. It further comprises components for accessing backend systems as well as for interpreting an application's configuration data and business rules that are provided via the AIM Management Console.

Solving critical issues of today's self-service applications, the described interaction management architecture provides significant added value for application management: Static applications such as IVRMenus and mCare-apps with a small service range can be converted into flexible, personalized, and configurable self-service solutions offering a new customer experience. At the same time, the service offering can be controlled centrally by an enterprise's business departments, in a way that is independent of the smartphone or tablet platforms as well as of the employed interaction channels.