Conspeakous VoiGen: Creating Context-Aware Voicesites

Kumar Sambhav Jain
JIIT,
Noida

kumar.sambhav.jain@gmail.com

Prabodh Prakash
JIIT,
Noida

prabodh.prakash@gmail.com

Nitendra Rajput
IBM Research,
India

rnitendra@in.ibm.com

Amit A. Nanavati
IBM Research,
India

namit@in.ibm.com

Information technology undoubtedly plays an important role in the socio-economic development of a country. Rapidly evolving ways to access information spread over the Web has benefited people in all aspects of life. However, there still exists vast untapped population at the bottom of pyramid which is aloof from these benefits. Reasons include lack of capital, infrastructure, awareness and illiteracy. Of late, we are seeing researchers going to the field to make information technology (IT) available to this class of people through simple interaction devices such as phone and interaction medium as speech. The idea of using the existing telecom network as a medium to reach information sources led to evolution of World Wide Telecom Web (WWTW). The WWTW enables access to IT services through voice interaction over phone. However due to the limitation of the sequential nature of Voicesite interactions, the cognitive load on the user can be significantly high for complex VoiceSite structures. In this paper, we present a novel system, Conspeakuous VoiGen - an architecture for creating context aware VoiceSites. Conspeakuous VoiceSites use contexts --- informative variables such as time, callerID, caller location, various caller preferences --- from the system's environment to gather useful information and present it in a flow that is more friendly and apt for the caller. The architecture has been implemented using two approaches: one that models the context as a finite state machine; and another that uses the Model-View-Controller approach to enable flexibility of managing the different contextual VoiceSites. The paper evaluates the benefits of the two approaches and presents the usability of the Conspeakuous VoiGen.

Voice User Interfaces, Design, VoiceSite, VoiGen, Conspeakuous, WWTW, Telecom Web, Spoken Web.

1. INTRODUCTION

World Wide Web has undoubtedly served as catalyst in development of human kind. It has defined the way people interact with not only people sitting in some other part of the world, but has also defined how people interact with thousands of computers across the world. As on May 2009, over 109.5 million [5] websites are operating round the globe. Web has touched almost every aspect of human life and the penetration is getting deeper by every day. The new and upcoming technology are continuously improving the way a website interacts with the users. The HCI of a website is getting better than ever before.

On one hand WWW is acting as catalyst for development of some economies but in developing countries like India there vast population at bottom of pyramid which is aloof from the benefits owing to lack of facilities. WWW is structured and efficient globally but requires considerable support from infrastructure at the user end. Internet penetration is still restricted in remote areas. In contrast, PSTN

has its influence even in weaker economies and interior as it easy to use and can be supported on lower cost and maintenance infrastructure. Voice interface, with WWW, as provided by WWTW

(World Wide Telecom Web) [1] forms a bridge between WWW and community at bottom of pyramid. WWTW is not limited to forming the bridge between WWW and people at bottom of pyramid. Using its standard, VoiceSites [2] have been made. A VoiceSite is voice driven application that can be easily deployed and run on a phone. A VoiceSite is analogous to website. Using a VoiLink [2], VoiceSite gets connected to another VoiceSite, thus making WWTW a powerful and promising source of information for people still living under covers.

Human can interact with outer environment only with the help of sense organs. The visual sense has a wider extension than any other sense. In "The Visual Literacy White Paper" [4] by Adobe, they talked about the importance of visual literacy and its powers and its benefits. Web technologies take the advantage of our visual sense in providing

the interface for interaction with computers. Lots of images, text, examples are provided to make things easier. Thus, WWW, in a way has been advantageous in terms of making better HCI. However, WWTW uses voice and hearing sense of human to perform an interaction. There are several limitations in using a voice interface: -

1. The data goes inside humans sequentially. Thus, only one thing can be interpreted at a time, opposite to the case of visual sense, where human are able to understand a complete dynamic situation with very ease.
2. Since information is processed sequentially, even to process a small amount of information can take lots of time.
3. Using Voice, the information has to be kept limited as too many of them can lead to confusion and wrong interpretations.

Thus, it becomes very challenging to create a model having interaction system closer to the way human interact. A static and predefined system cannot be the solution as with static system there is always a chance to get busy in dealing with non-relevant things uselessly. Thus, even from initial observation one can say, intelligence has to be embedded efficiently in any voice driven system to get a usable interaction way.

Earlier we have seen the architecture of VoiServ and Conspeakuous. VoiServ was the first architecture to be proposed for making a VoiceSite. The VoiceSite it made was completely static in nature. In all cases and situation, it made same interaction with the user. This way of interaction obviously has its limitation. To overcome this limitation, the architecture of Conspeakuous was proposed. While VoiServ concentrated on making an architecture for VoiceSite and its deployment, Conspeakuous concentrated on making an architecture that improves the HCI of a VoiceSite. In our paper, we present a novel system that takes into account the creation of a VoiceSite with improved human interaction methods. We present an architecture to create intelligent VoiceSite. Since lots of information is present while interaction with users, Conspeakuous VoiGen takes into consideration two central questions

1. What: What needs to be presented to the user is very important. Since, a lot of information source exists, it is not necessary that every information remains important every time. Thus selection of correct information is very essential.
2. When: A information is useful only when it is presented at right time to the user. Thus, it is again very important to present correct information at the right time.

We have followed to approach to design the Conspeakuous VoiGen. One uses the template based approach to achieve efficient and flexible template of a voice site and other deals with context using FSM.

2. RELATED WORK

IVR systems provides voice interface to an information system. Conventional IVR are mainly DTMF based or have restricted speech recognition support. WWTW [1] gives platform for interconnected voice based applications termed as VoiceSites (similar to websites). VoiceSites can be interconnected using 'VoiLinks' (analogous to hyperlinks) which are links between two voice applications within the web. VoiLinks can span across different enterprises enabling cross-organizational workflows driven by a voice interface over an ordinary phone.

Conspeakuous –architecture for modeling, aggregating and using the context in spoken language conversational systems. Since Conspeakuous is aware of the environment through different sources of context, it helps in making the conversation more relevant to the user, and thus reducing the cognitive load on the user. Additionally, the architecture allows for representing learning of various user/environment parameters as a source of context.

3. PROBLEM STATEMENT

HCI is an integral part of any software. A software with excellent performance is of no use if it is not presented well. For a system with limited and only relevant source of information, the work is easier. But, in case where the source of information varies and not all the information is relevant every time, the way to provide interaction between computer and humans become more tough. The reason for such cases being complex are: -

1. Since only a subset of all information is relevant, a good interaction design would be one which interacts with users only considering those information. Thus, an extra effort has to be put in handling the relevant and irrelevant informations.
2. As the working condition changes, the relevant information may change and it is possible that the interaction part needs to be changed completely.

Thus, an effort has to be put in the collaborative working of the back-end, where all processing is being done and front-end, to which user interacts. We have lot of information around us, many of which are contextual. For a particular task, we need

to use a finite set of relevant information, while others can be easily ignored. However, in the classical way of VoiceSite Creation that VoiGen used, we were not able to limit the source of information for performing tasks. Thus the whole process of interaction with user was overburdened as time was uselessly spent in answering to questions that were of no use in the current situation. The only way to make it better is to limit the source of information only to relevant ones. We saw Conspeakuous, providing us with an architecture to limit the source of information and perform task in a better, reliable and efficient way. However, both of them are discrete system. It is very important to make a new system that has the capabilities of both VoiGen and Conspeakuous. If we ignore the importance of having a such a system, then VoiGen would continue to work in old fashion and Conspeakuous is not of much use in itself. Also, due to high load of interaction both the system, eventually, would be a failure if they continue to work alone. Individually, they lead to less throughput. A user would spend lots of time and resource answering to irrelevant questions and would get output out of a certain subset from the set of questions.

4. CASE STUDY

Shyam, who is a small taxi vendor, receives many calls from his customers to get taxi. There are many queries from the customers that Shyam needs to answer every time. Also, answering to calls on phone keeps him busy for a significant part of day. Although he maintains a web site for his company, he observed that people preferred to call him to get taxi booked. So, he decided to make a VoiceSite for his company, so that when customer call him, they interact directly with the VoiceSite and get their taxi reserved. He discusses the importance of making such a VoiceSite with his branch offices in different cities. Once all of them agreed, he asked them to pen down their requirements for the VoiceSite. After he received the requirements, he saw there was a great variation in the way the officials from different branches wanted the VoiceSite to be made. He observed difference in the services they want also noted the vast amount of information that was needed to by user to answer before he could reserve a taxi. Already aware of VoiGen and Conspeakuous, he realized that on one side VoiGen was unable to make a site that makes intelligent decisions and on other hand it would be a hard task to make Conspeakuous work for them, as that required lot of administrative efforts to modify Conspeakuous according to their requirements. Thus, he decided to ask his phone subscriber to prepare a system that would consider the service requirement of all the taxi vendors and

would enable them to choose from services list , the service they would like to enroll for their VoiceSite. Also, he requested them to make the VoiceSite intelligent so that the end users who call do not have to spend long time on phone answering redundant questions every time. The context that could be used in intelligent manner are :-

1. **User Type** : An information about new user and old user, would help the taxi vendor to add relevant welcome message for the person who calls.

2. **Time of Day** : Time of Day can be used to greet user on time basis. He felt, a greeting saying good-evening in evening and good-morning in morning would be generous to users.

3. **User History** : The user history would enable the VoiceSite to know about the user preferences like, no of person, taxi type, AC preference for taxi, favorite travelling destination. Shyam felt he could use these information to suggest car and travelling destination to the users. This way the load on user to answer a set of question was reduced to choosing car, travel location from the suggestion which were made from his personal preference.

4. **Weather** : Shyam also felt the need of adding current weather information, which along with user preference can be used to suggest a more relevant car.

5. **Travel Records** : Shyam categorized travel record in three types : Past Travel, Present Travel, Future Travel. He wanted these records to be used in a better manner. For e.g. if a user has future record, the VoiceSite could instantly ask him, if would like to make changes with that travel. In case a user is in present journey, the VoiceSite asked about present journey, how well is it going ?

The requirement specified by Shyam are the salient features of the Conspeakuous VoiGen.

5. OUR APPROACH

We have followed two different approaches to make the Conspeakuous VoiGen. While on one hand we used MVC to enable flexibility in managing contextual VoiceSites and on the other hand we followed the concepts of FSM to achieve the easiness in the handling of large amount of context during the processing of VoiceSite. In the next sub-sections, we discuss both the approach individually.

5.1 Context Management using FSM

A lot of information is present while interacting with user thus it is very important to have clearly defined

context, context boundaries and transition among context. To maintain all data efficiently, FSM has been used, which provides with following benefits: -

1. States can be easily defined using FSM. The concept of defining states, for different conditions while interacting with user, gives us a way to define the context to be used and functions that needs to be implemented in a box and keep it aloof, and thus safe, from the world.
2. States also associate with itself the information of next state to come. The transition of a state to another state depends on the response of user. We can also implement conditions in FSM to redirect the current state to next correct state.
3. Using FSM, there is an ease to add new states, delete old one or alter a transition. Thus it becomes easier to handle new situations dynamically.

Using FSM, we are able to make a VoiceSite using which, context was handled very efficiently. At every stage the VoiceSite knew, where it is and where has it to go from now. This approach helped us to manage large number of context in discrete fashion. For all states, the VoiceSite considered only relevant context and ignored all other form of informations. The response of user along with the current contextual information helped the VoiceSite to move from one state to another.

Figure 1: FSM with three states and redirection lines

In the figure 1, a FSM is shown. Blank circle shows the state, directed lines shows the direction of transition. In the figure 2, a FSM shows the easiness of adding a new state.

Figure 2: The above figure depicts how to add a new state in between two states. To add a new state the redirection lines have to be modified. In the same fashion, states can be deleted, moved or modified by doing proper operation on redirection lines and states.

Figure 3 shows our working FSM model that we used to implement the scenario. We start with TWMS and then move on to NUWMS or OUWMS, depending on a check condition that verifies, if the user is new or old and if welcome messages are available or not. If only this portion of FSM is considered, the important features of this part is: -

1. The context information that was being considered was only the type of user. Any other form of context information was completely ignored.
2. Since the only importance of the both the state is to make user listen welcome message, a state for them could have been simply ignored. The FSM could have moved to next relevant state and could have shown these state on redirection lines easily. But, a state has been dedicated to it, to recompile all the relevant form of information, even when the VoiceSite only present a static output to the user. In doing so, the FSM not only drop the type of user information, which becomes irrelevant now, but also picks up new relevant information. The use of this state shows that using FSM model, we can very easily keep track of all, even small, processing of VoiceSite. Thus, every minute detail can be easily tracked and any change in relevant source of information can be detected, to make a further change.

We have continued to make FSM in same fashion, till we reach an end. Thus FSM provided us with efficient methods through which we were able to reduce load of too many informations, to only relevant ones. Also, with FSM, the any states can be easily handled for any modifications.

Figure 3: The above figure shows the working model of the FSM we used to implement our working scenario. A blank circle shows come check condition, which is then redirected to proper state. In this working FSM, we have assumed that the initial state is always constant and user always choose timed welcome service.

5.2 Template Based

5.2.1 Overview

This approach is inspired from earlier version of VoiGen. After receiving all the inputs, it parses through the data to actually generate a VoiceSite. The generated voice site either employs pure VoiceXML or adds dynamicity using JSP. Using this methodology it becomes difficult to flexibly add intelligence through context sensitivity in generated voice site.

Conspeakous [2] architecture efficiently separates context part from the conversation part. The context part is not hard coded so that application is flexible to changing contexts. Nature of conversation is controlled by Situation computed from various context variables. Contexts are fetched from separate module context composer that fetches context values to Situation Composer.

In this approach, we have tried to combine features of Conspeakous to a new MVC based design that can generate context sensitive application. Rather

than generating code for a voice site, the approach advocates a complete template for a type of voice site. In template based approach, every voice site that solves same purpose uses one physical code only. Each voice site of same type (like Travel Voice Site A, B...) is identified by the number it is associated with. This way the process of generating a voice site is simplified to just creating new model (database and preference parameters) whenever creating a new voice sites. This approach significantly simplifies the task of creating a voice site by a common man as depicted in results.

At any point of time, behavior and content of conversation is controlled by Situations. A Situation can be understood as function of 'n' context parameters.

$$S=f(c_1,c_2,c_3...,c_n)$$

where the logic of function 'f' and data type of c_i is formulated by the developer. For understanding, consider possible situations that may persist when a call is initialized.

The number dialed by the caller tells the template which voice site's model (database items and preference parameters) to be used. Caller's number distinguishes between old and new caller as reflected by database. Using caller id of an

existing caller one can also query database to conclude whether the caller is just an old customer, customer with pending booking or a customer with a currently running booking (using time also as context variable)

Situation	Context Variable Used	Impact on conversation
NEW_CALLER	CalledID, CallerID	Present standard menu and add caller to database model
OLD_CALLER_NOT_CUSTOMER	CalledID, CallerID	From the knowledge of last call, like if called queried something, suggest him apt options.
OLD_CUSTOMER	CalledID, CallerID	Application now knows about his various item preferences. Present options that he is likely looking for.
CUSTOMER_WITH_PENDING_BOOKING	CalledID, CallerID	The caller is customer with a pending booking. He is likely to be interested in cancelling or modifying his booking.
CUSTOMER_WITH_RUNNING_BOOKING	CalledID, CallerID, Time, Caller Location	The caller is currently in between a journey. He is likely to share his experience, book further trip or looking for facilities like hotels, rest rooms etc. Appropriately direct him to required voice site.

5.2.2 Implementation

The complete template of a voice site is as web application in itself is deployed on a web server like Glassfish or Apache Tomcat. To achieve above stated features we have followed Model-View-Controller paradigm. While implementing MVC paradigm, model is commonly represented by entity beans, the view may be represented by a Java Server Page and controller may be represented by a servlet. This separation of presentation from model and control supports ability to add and modify new contexts and situations without starting from the scratch. Initially the call is handled by a Java Servlet that uses callerID and callerID to compute Situation using database and preference parameters. Then it sets Situation bean as session variable which is available throughout call duration.

```
// Situation.java
public class Situation {
    protected SituationCodes code; // Java enum
    protected int ordinal;
    protected String description;
}

// SituationCodes.java
public enum SituationCodes {
    NEW_CALLER,
    OLD_CALLER_NOT_CUSTOMER,
    CUSTOMER_NO_PENDING_BOOKINGS,
    CUSTOMER_FUTURE_PENDING_BOOKINGS,
    CUSTOMER_RUNNING_BOOKING,
    CALLERID_DISABLED,
}
```

After saving Situation as session variable, the control is transferred to an appropriate JSP page (the presentation part). The JSP page has VoiceXML tags for prompting and accepting user inputs. The grammar (SRGS- Speech Recognition Grammar Specifications) used for recognizing user input is also generated on the run by implementing it as a JSP page rather than static .grxml file. This also improves speech recognition efficiency as only relevant items are included in grammar specification. The JSP page for presentation is divided into switch-cases where each case corresponds to a SituationCodes enum value. The page uses Situation session variable to know which part of JSP page to be executed.

5.2.3 Results

We implemented Conspeakuous VoiGen using both the methodologies. Below we present the

sample conversation between the user and the system

```
VoiceSite: Hello, how can i help you?
Caller: Book a taxi.
VoiceSite: What will be your destination?
Caller: Delhi
VoiceSite: When would you like to travel?
Caller: 15th december
VoiceSite: Which car you prefer?
Caller: Indica
VoiceSite: Your booking on 15th december is confirmed. Thank you.
```

Now caller calls while he is travelling.

```
VoiceSite: Welcome again. How can I help?
Caller: Book further.
VoiceSite: What will be your next destination?
Caller: Noida
VoiceSite: Your trip to noida is confirmed. Thanks
```

As we can see that in second case the conversation is context sensitive and thus simplifies and more useful.

Some of the strong features that we are able to achieve through use of above architecture are:-

1. Reduced Time of Interaction with user: - Through the use of relevant context at proper time, we were able to reduce the time of processing of a VoiceSite considerably. In our scenario, we tested VoiceSite, once using Conspeakuous VoiGen and also by running classical VoiGen. We observed that we are able to reduce time by 20 seconds. While VoiceSite made by classical VoiGen took 1 minute 3 seconds to process our scenario. VoiceSite made by our architecture completed the whole work in 43 seconds only.
2. Standard of efficiency and reliability: - The output made by Conspeakuous VoiGen was same as that made by classical VoiGen, thus we are able to make a system that produced same standard of result in lesser time. This was our throughput was also increased significantly.
3. Improved HCI: The HCI of telecom web was suffering before this. Through our architecture, we bring telecom web more near to the way humans, thus making it more real time.

To conclude, through our improved way of interaction between the system and humans, telecom web now has strong potential to come out of papers and make a strong impact in life of every people and especially help people still under covers to come at par with the world

6. REFERENCES

- [1] Kumar A., Rajput N. , Chakraborty D. And Nanavati A.(2007) WWTW: The World Wide Telecom Web. *In ACM SIGCOMM NSDR, 2007.*
- [2] Kumar A., Rajput N.,Chakraborty D.,Agarwal S., and Nanavati A.(2007) Voiserv: Creation and delivery of converged services through voice for emerging economies. *In Proc of the 2007 International Symposium on a World of Wireless, Mobile and Multimedia Networks. Finland, June 2007.*
- [3] Arun Nair S., Nanavati A. and Rajput N., Conspeakous: Contextualising Conversational Systems. IBM India Research Lab
- [4] Bamford Anne
http://www.adobe.com/uk/education/pdf/adobe_visual_literacy_paper.pdf (14th Dec' 2009)
- [5]http://en.wikipedia.org/wiki/World_Wide_Web (14th Dec' 2009).