Generic dialogue structure for vocal access to indexed databases

Christophe Dupriez
DB Scape s.a.
dupriez@dbscape.com

Mélanie Roland
DB Scape s.a.
roland@dbscape.com

Abstract

In this article, we first describe the different steps and adaptations that were necessary to develop the VocaBase™ project. VocaBase™ is a vocal server allowing users to have dialogue with an indexed database. The original algorithm that we designed allows a dynamic adaptation of the dialogue depending on the database content and on previous choices made by the user. The flexibility of this algorithm allows the development of complex application required by the current market. Finally, we describe the VocaBase™ Studio, an Internet based development environment allowing vocal applications to be developed and tested with different databases.

1 Introduction

The aim of the VocaBase project is to realize an information service accessible through the telephone, and thus more accessible than web applications. We therefore needed to organize large amount of information in indexed databases, and to allow a phone access to these databases.

Natural language is important to access large databases, since systems accessible to a wide public cannot require users to know "commands" or to read a manual.

DB Scape s.a. has a long experience in the organization of large databases. However, the technical aspects involved in the implementation of a vocal access forced us to completely revise the way information is presented and organized. Indeed, vocal access must take the following points into consideration :

- Compared to a web application where the screen allows a large quantity of information to be displayed, a vocal application is very limited in the number of items spoken to the user. This is due to the fact that a user has to memorize an important part of what he is told.
- A standard dialogue cannot exceed a few minutes and must take into account the fact that a user may not be willing to write down all the choices offered to him.
- Finally, during the interaction, it is important to let a user understand where he is positioned in the tree of possible dialogues. Explanations must therefore be very precise and structured, in order to facilitate the user navigation.

In this article, we will use the example of an automatic call center targeted to potential tourists. Using this call center, a tourist can query a database of events and activities in France, organized by region, by type of activity and by customer type (children, elderly, reduced mobility, etc.). We are testing now this system in French and in English. Our final goal is to encompass most European languages.

2 Existing solutions

2.1 Principles

Like HTML for Web applications, VoiceXML is seen today as the "do-it-all" language to define a vocal application. Different "Wizards" or development environments (IDE) are proposed on

the market to generate VoiceXML visually based on a state transition diagram or on a form design (http://developer.voicegenie.com/IDE.php http://codecenter.telera.com/CodeCenter/ValidateLogin.asp http://www.voicewebsolutions.net/Products/studio/index.htm).

These solutions do not meet our objectives because we want to automate VoiceXML generation on the basis of our own dialogue model. This model has to take into account the needs we have identified for user interaction with catalographic databases (libraries, commercial products, governmental information, news, etc.). VoiceXML is generally used to propose "Menu" (fixed navigation trees) and "Form" (mixed initiative) types of user interactions. (http://www.w3.org/TR/voicexml20)

With "Menu" based interactions, the system switches from one level in a hierarchical dialogue to another, depending on the choice made by the user. After some interactions, the user finally reaches what interests him (or a "Form" to be filled to conclude a transaction).

With "Form" based interactions, the system offers a given number of fields to fill (for example: destination, type of activity, type of public, date of departure, etc.). These fields are fixed before starting an interaction and are often directly linked to the skeleton of an SQL query. The system asks questions to the user in order to fill up each of the fields and only then searches the database for the possibilities corresponding to the user's request. It is not a real dialogue where questions are refined following each user's answer. In such an application, a user may ask for a non-existing combination and he will have to start again his query.

2.2 Problems

In "Menu" based applications the user is obliged to follow a pre-defined order of choices to reach the desired solution. When the database includes multiple dimensions or a wide number of elements that are indexed by several values, the number of combinations of choices is exploding. Therefore "menus" become very large and difficult to manage: content adaptation requires continuous updating of the dialogues to keep the interactions reasonably short.

"Form" based applications are not well fitted to access databases with hierarchical classifications or with dependent dimensions (not all choices remain for some fields when one or another has been filled), because it is difficult for the user to further refine his search after the form has been completed. There is indeed no pre-established structure of dialogue that would allow the user to simply express his wishes after listening to the previous search results.

3 The structure proposed by the VocaBase project

3.1 A little history

Since 1986, members of our team have been working on information indexing, classification and retrieval based on thesauri.

In 1997, we started to study the optimum dialogue structure between a "catalogue" database (as defined in 3.2.1) and its users.

Since 2000, we are working on the VocaBase™ project and we have been using the results of our previous researches in the field of vocal applications.

This year, our objective is to test the results of our research on « real life » applications, and allow a very large number of projects to test our ideas.

3.2 Principles

3.2.1 Data modeling

The applications that we are addressing are "catalogues" (bibliographic catalogues, products catalogues, FAQs, events catalogues, etc.). Users of such applications hope to find some solution to their needs by consulting such a "catalogue" type of database. Users want to identify the suitable solutions and, eventually, start a transaction based on some of the identified solutions. The goal of the VocaBase™ project is to identify the best dialogue model to query such databases and to bring a generic solution to this class of applications.

The "objects" included in a "catalogue" database can be of various types: books, legal documents, CDs, tourist activities, commercial products, etc. Each of these objects must receive a name, a

description, and an indexation by concept from different domains: subject, location, period of time, targeted users, etc. A user can select some objects and then make a transaction (obtain information, purchase, contact through SMS or by E-mail, etc.).

The object description must be representative of the object itself : the user must have enough information to take a decision and make a transaction (most probably ask for more information or establish a contact).

The indexation of each object must be precise enough (few objects should share the same indexation combination). Each indexing term must have enough synonyms to allow the user to use his own words. Hierarchies of indexing terms (with obvious generalization concepts) enable users to express broad needs and explore other indexes to narrow down their searches.

For example, in the case of a database about books, these books may be indexed by author, type of book, publisher, price level, but also by different subjects that could interest the readers. Authors can be grouped by countries or by centuries. Subjects can be grouped along different classifications; they can be assigned different synonyms.

These concepts are familiar to "catalogue" databases builders. But they must be adapted to the usual guidelines.
(http://www.loquendocafe.com/index.asp
http://developer.voicegenie.com/tutorials_VoiceGenie.php) for vocal applications :

- Since descriptions must be read to users, they cannot exceed a few sentences (from 40 to 100 words). They must be simple and complete. Sentences should be short and written with simple words. Text to Speech (TTS) systems do not yet have all the warmth and intonation of human beings. Tags may have to be included in the texts to guide TTS especially around proper nouns of different cultures.
- Each object may need different size of names and descriptions for different uses in the dialogues : choice (very short but discriminating text to make a choice in a list) or explanation (concise but complete text to help the user confirm his choice).
- To facilitate the user's understanding, the indexation must be organized in hierarchies of concepts that are both clear and natural to users. The user must indeed quickly understand the meaning of the concepts (at whatever level they are), and should be able to guess the next branches of the hierarchical decision tree. For example, if we take a database of books, the concept "novel" could logically be divided into "thrillers", "spy stories", "love stories", etc. In order to ease the interaction with a user, this hierarchical structure should not include too many levels, or too many choices by level. 3 to 4 levels including 4 to 8 propositions (Miller, 1956) each should be the ideal dialogue structure. Ideally, indexation types ("dimensions") should be limited to a maximum of 4, in order not to have a too complex dialogue. However, the existence of different indexation types (for instance activity type and region) allows diversification and refinement in the search: it is better to have more than one indexation type.

3.2.2 Data Management

GenIdex®, the indexation and query engine developed in a previous project, allows us to propose a dynamic search to the user. At each level in the dialogue, the user is guided depending on what he has previously selected and on the content of the corresponding part of the database.

In a "Form" based application, type and region would have to be specified even if non-existing combinations are possible (surfing in Alsace for instance!). In a "Menu" based application we may oblige the user to narrow down a dimension before switching to the other (for instance, having to choose a given region of France before having the opportunity to specify "surfing").

This is where all the originality of our system resides: at each stage of the interaction, the system will calculate what is the best selection of choices it can propose to the user. Using hierarchies, the choices proposed are just broad enough to encompass most of the database. This selection is (for now) based on a weight calculated depending on the number of activities linked to each entry in the index. This will evolve to take into account the

weight of each activity (for instance, inventory value of each product in a catalogue).

Therefore, instead of listing all the indexes that are available at one level of the hierarchy, VocaBase™ will calculate, at each specific step of a dialogue, the most representative choices that will be proposed to the user.

Those calculations are done continuously during dialogues with users and are therefore always up to date with the database content.

VocaBase™ also offers the opportunity for the user to select something that has not been proposed by the system. Indeed, if the user wants to go to a specific city that was not stated by the system, he will simply name the city, and the system will adapt to this new choice. This shortens the interaction time by allowing the user to skip many choices that have nothing to do with what he wants. This feature can be carefully tuned depending on Automatic Speech Recognition (ASR) capabilities: only the most pertinent terminology for the current selection in the database is loaded in the ASR.

We also allow a user to express complex requests, by linking several choices with “and”, “or”, "not". For example, if a user says : “I would like to ski in the Alps.”, the selection will be the intersection between the two criteria (activities that are indexed both by “Alps” and “ski”). If a user says : “I would like to go to Alsace or Brittany.”, the selection will be activities either in Alsace or Brittany. If a user says : "I want to go in Provence but not for kayaking", the selection will be activities in Provence apart from kayak boating.

3.2.3 Dialogue Modeling

At each step of a dialogue, VocaBase™ has to propose possible choices to the user. If the number of propositions is too small, the number of iterations to reach the solution will increase. But if the number of propositions is too high, the user will forget what propositions were made.

Suppose that the maximum number of proposed choices is set to 5 (as suggested by Miller, 1956. This number is a tunable parameter).

At each step of the dialogue, the system will search the possible activities satisfying the requirements already stated by the user. If 5 activities or less are found, the system will list them and the user may then select one or another to hear its description and to possibly start a transaction.

However, in the majority of cases, a larger number of activities correspond to the user's requirements. It is of course not possible to propose all the activities. The system must therefore present new criteria to allow the user to refine his search.

Procedure :

Suppose that the user has selected "A" as a first choice and that "A" corresponds to hundred activities (set denoted D(A)). These activities are linked to "A" but also to other index entries that we can denote S(A) (In our test database, an activity is always linked to a region and to one or more types.) The GenIdex® algorithm will then select amongst S(A) the entries that represent sizable subset of activities. The goal is to supply to the user a list of about 5 choices that well describe the selected part of the database.

Some of these entries can represent more than 80% of the set D(A). For example, a majority of the activities corresponding to a user’s choice (boating for instance) could well be located in the South of France. Index entries like "South of France" will not be considered when making proposals because they are too frequent, not enough discriminating and therefore do not help to narrow down user's needs.

On the other hand, many index entries could be linked to too few records in the set D(A). They will not be presented to the user because they are too many to be spoken.

Finally, indexes can be ranked. For instance, "Regions" could be considered as the most important index, "Activities Types" as the second one and "Target Customers Types" as a less important one : this last index is never presented in the generation (but always available for the recognition).

Untold index entries remain understandable by the ASR if the user spontaneously mentions them ("I want to go with my children in Alsace."). For example, the ASR keeps understandable too frequent entries like "South of France" (more than 80% of the set D(A)) and a selection of less frequent entries is also kept, respecting the ASR capabilities.

An important issue is the use of hierarchies in the generation of proposals to the user. Very often, a lot of activities are in a given region, with most

of them in a sub-region. For instance, Kayaking may be possible in the North and the South of France. In the South of France, it may be dispersed in many regions with a concentration in Provence. Because we have to limit propositions to users to about five, we cannot list everything. We use "other", like in "North of France, Provence or other spots in South of France ?". We dynamically load the ASR with the fact that "other spots in South of France" means, at this specific dialogue step, all regions of South of France except Provence.

This has been extended to multiple levels of a hierarchy. If "Castle" is a "Historical Monument" which is a "Monument" which is a "Visit", and if "Parliament" is a "Monument", we may face different situations :

- "Castle" is a frequent entry (heavy weight) in S(A) and all others Visits are "light weight", VocaBase™ will generate "Castle or other Visits?". "Other visits" means here all visits except castles.
- If "Parliament" is somewhat frequent, VocaBase will generate "Castle or other Historical Monuments, Parliament or other Visits ?". "Other Historical Monuments" means all historical monuments except Castle and "other visits" means here all visits except historical monuments.

3.2.4 Example

A VocaBase dialogue can be separated in different steps :

- The very first step gives a general welcome and introductory message, a presentation of available commands (which may be skipped) and a list of choices based on the global database (or any specific part that can be pre-selected knowing the user's phone number).
- The second, central step confirms the user's last choice (A), gives the number of remaining activities (size of D(A)) and then presents a list of choices of about five index entries chosen in S(A). This step is repeated until about five activities or less are selected in the database.
- In the third step, the five (or less) activities are listed to the user.
- In the fourth step, complete information about an activity is given to the user and a transaction can be requested.
- Help can be provided at any stage of the dialogue if the user asks for it or if too many recognition errors arise.

S :
Welcome to "Outdoor is Fun".

If you're not interested by the following introduction, you can skip it by saying "main menu".

[I'm a computer. I'll better understand you if you speak clearly and simply ! But don't hesitate to interrupt me.
Say "return" to come back to your previous choice. Say "detail" to get more choices. Say "main menu" to come back to the first choices.
Say "stop" to close the dialogue. At any moment, you can say "help" to again listen to this introduction.]

We propose 237 activities in France.

Amongst these 237 activities, regions are: Brittany, Brittany coast, Opale Coast or another spot in North of France, Provence or another spot in South of France.

What would you like ?

⇒ ***U : I would like to go in Alsace.***

S :
There are 20 activities for « Alsace ».
Amongst them, Activities types are Trekking or other Sport, Castle or other Historical Monument, Local curiosity or other Visits ? Regions are Colmar or another spot in Alsace ?
What would you like ?

⇒ ***U : I prefer trekking.***

S :
There are 4 activities for « Alsace » and « trekking ».
First activity : "Beau Site" Horse Riding Centre.
Second activity : Col of Hundsruck.
Third activity : Lakes of the Ballon d'Alsace.

Fourth activity : La Bure Celtic Camp.
Please select one of these activities : "Beau Site", Hundsruck, "Ballon", "La Bure".

⇒ ***U : La Bure ?***

S :
La Bure Celtic Camp :
La Bure Celtic Camp is four kilometers to the north of "Saint Die". Arrow tips and other tools made of flint prove that this Camp was occupied during the Neolithic age. About a hundred inhabitants were living there : they used agriculture but also metallurgy as shown by anvils and other metal tools found on the spot.
You can ask for more information, return to Main menu or choose one of these activities : "Beau Site", Hundsruck, "Ballon", "la Bure".

⇒ ***U : Information***

S :
For Celtic Camp of "La Bure", the phone number is 0033 - 02 35 78 94 41 65.
Just say "Message" to receive this phone number by SMS. You can also return to the Main menu or choose one of these activities : "Beau Site", "Hundsruck", "Ballon", "la Bure".

3.2.5 Error processing

ASR errors are very frequent (Picheny and Padmanabhan, 2002) so the system must confirm the words it recognized and the user shall be able to easily cancel any system misunderstanding.

Whenever too many errors happen in a dialogue, question generation is changed to suggest using DTMF to the user by numbered choices.

Dynamic generation (and interpretation) of VoiceXML grammar also permits to change recognized vocabulary depending on the context : a maximum of terms (which can be tuned for different ASR performances) is chosen amongst terms in S(A). This greatly helps to reduce ASR error rate.

3.3 Advantages

The main advantages of VocaBase™ results, compared to other approach, are :

- Guidance : the user may narrow down gradually his search depending on his needs and on the real possibilities offered by the database.
- Flexibility : the user remains free to state his interests in any order he wants, and not necessarily in the order proposed by the system.
- Dynamic adaptation: proposals made are function of choices the user has made. They are also functions of database content (which may evolve continuously). "Other..." is used to simplify propositions to the user and depends on the current context.
- User adaptation: the starting set in the database may be selected depending on criteria specified in the user profile (which can be associated to any user authentication mechanism like the calling phone number). Also, if the user requests “details”, the number of proposed choices may be expanded.
- Sophistication : "power users" may combine multiple criteria in one sentence with implicit or explicit Boolean operators ("and", "or", "not"). "I would like to surf in Brittany.", "I want to go to Alps or Pyrenees.", "I want to do kayaking but not in Provence.". These shortcuts accelerate the dialogue.

Updates are very easy: a new activity is easily added to the database and whenever its indexation is completed, GenIdex will recalculate the virtual hierarchy to access it. There is no manual intervention to have a dialogue continuously adapted to database content.

4 Conclusion

We have presented a strategy and an experiment opening the way toward fast deployment of "catalogue" databases through a vocal server.

We designed the VocaBase™ Studio, a development environment permitting any database author to benefit from a vocal access with very little effort. Learning Voice XML is not even necessary.

Adapting a set of data to a vocal server is done with the following steps :

As a first step, the author loads his catalogue files (descriptions, terminology, indexation) in the VocaBase™ Studio. The Studio is accessible through the Internet.

This data must follow a structure composed of three simple tables but it must also be organized in hierarchies (trees of index entries going from the general to the specific). Translation in all desired languages must be provided for all your terminology and descriptive texts. Indexation must follow the principles given above : statistics are provided by the Studio to help you.

In a second step, the author must parameterize the dialogue itself : he can decide which sentences will be told at each point of the VocaBase™ dialogue model. It may be necessary to specify, for each index entry, a preposition and/or an article to permit a better sentence generation.

VocaBase™ Studio is open to the VoiceXML services of many Voice Application Café and an application may use one or more of those to be accessible locally to users in different languages and countries.

The author can then test his application by telephone and update his data directly with the VocaBase™ Studio.

Our aim is to gain experience in the dialog management and the different users interactions. Our web site gives access to VocaBase™ Studio. Any person interested in testing the concept and the algorithms with his own data is welcome to do so.

5 References

Miller, G.A. 1956. The magical number seven, plus or minus two : Some limits on our capacity for processing information. *Psychological Review*, 63, 81-97. http://www.well.com/user/smalin/miller.html

Padmanabhan, M., Picheny, M. 2002. Large-Vocabulary Speech Recognition Algorithms. IBM T.J. Watson Research Center

Miller, M. 2002. VoiceXML. 10 projects to voice-enable your web site. Gearhead Press. Point-to-Point.

VoiceGenie IDE documentation

http://developer.voicegenie.com/IDE.php

Telera IDE documentation

http://codecenter.telera.com/CodeCenter/ValidateLogin.asp

VoiceWeb IDE documentation

http://www.voicewebsolutions.net/Products/studio/index.htm

Voice XML 2.0 specifications

http://www.w3.org/TR/voicexml20

Usual guidelines :

Loquendo Café tutorial

http://www.loquendocafe.com/index.asp

VoiceGenie tutorials

http://developer.voicegenie.com/tutorials_VoiceGenie.php