Telephone Voice Interfaces on the Cheap

Thomas Hornstein

UBILAB, Union Bank of Switzerland
Bahnhofstr. 45, CH-8021 Zurich

e-mail: hornstein@ubilab.ubs.ch

Traditional interactive voice response applications are based on well-known menu-like structured dialogues using DTMF. This navigation technique is application-dependent and has limitations. It cannot be improved by simply switching from DTMF to voice input. Rather, we propose an application-independent navigation method called Zap & Zoom in combination with voice and key input. Users can Zap over a list of items (subjects) and Zoom into items of interest (content of subject). A set of application-independent commands was defined for this type of navigation and trained for voice input in three languages. Design recommendations have been set up to employ the Zap & Zoom navigation in telephone information systems and to achieve an open, easy-to-use and consistent voice interface. Two different information services based on the Zap & Zoom navigation were built.

1 Introduction

Telephone-based information services have been introduced in the last decade. The interaction between the machine and the user was based on the telephone and its key pad using touch-tone (DTMF[1]) signals. This technique is fairly efficient since it is simple and people are used to seeing a similar interface on other devices like automatic seller machines etc. Unfortunately the distribution of DTMF based telephones is still not homogeneous in most countries. In some areas DTMF dialling is not yet supported. In many countries - particularly in Europe - people often keep using their old pulse-dialling telephone. Additionally ISDN has been introduced as a third standard. One way of building interactive voice applications is to support the various communication standards. Another way is to be independent of the communication standard by supporting voice input.

Supporting various communication standards raises several problems in the case of pulse detection. The response time of the application depends on the entered digit. Pulse detection is also error-prone. Additionally phones with a dial do not have a star or a hash key. These limitations could be overcome by the so-called pocket dialler (DTMF-generators) but this is an additional device and is often not accepted by, or not available to customers.

On the other hand, telephone-based information services supporting voice input are independent of the type of telephone equipment and the underlying communication standard. Another reason for supporting voice input is that speech interfaces are often more time-effective and subjectively easier to handle for novice users than DTMF interfaces [FRM93]. Since the beginning of the nineties the deployment of applications with voice input has grown slowly. There are two main reasons: first, the computing

---

1 DTMF=Dual-Tone Multi-Frequency

power needed for voice recognition is expensive; and second, the development of voice recognition vocabularies for particular applications is very time-consuming. The latter is still true while the former has become less important. This paper describes how we avoided setting up the development of expensive and application-dependent vocabularies.

In section 2 we first describe the requirements for voice navigation and summarise the traditional navigation methods and their limitations. In section 3 we introduce the domain- and application-independent navigation technique Zap & Zoom - an alternative to the traditional menu-based navigation method. In section 4 we then describe our design model for IVR applications which leads to generic telephone information services and faster development. Finally we show an example of a phone banking information service using the ideas proposed.

2 Voice Navigation

2.1 Principles

In early 1992, we started to investigate voice technology with the aim of building telephone-based information systems. After studying basic aspects, we realised that human computer interaction and the ergonomics of voice interfaces are the essential factors in the success of information services for occasional users. This leads to a user oriented system development where usability tests play an important role.

Many papers describe various kinds of DTMF-based interfaces [HAL89], [DET90], [PEL93], design criteria for telephone based applications [KLO94] and style guides [FRT91]. The described techniques do not help in the design of voice recognition interfaces. They often describe only basic interaction types such as the traditional menu navigation. We define the following guidelines which help to extend the user interfaces of telephone information systems successfully to include voice input:

- Application-independent navigation
- Suitable for selections from a large number of choices
- Easy for novices and fast for experts
- Active users instead of passive users

The following sections describe why traditional navigation techniques do not satisfy all these guidelines.

2.2 Simple Menu Approach

Traditional menu navigation is tree-based. It uses the digits zero to nine, yes and no as keywords. These twelve keywords are simply a mapping of the telephone key pad. The computer first plays a message containing all possible options of the menu. The users are then asked to select one option by speaking a digit specifying the number of the option. The advantage of this approach is that only one small vocabulary is needed for both navigating through the service and entering data (integer values). This helps to provide a high recognition accuracy and keeps the training effort for vocabularies in different languages to a minimum.

Regardless of whether voice or key input is used this method has several limitations and drawbacks which prevent the building of more sophisticated services:

- Users are forced to listen to long menu prompts before selecting one option. That means most of the time users are inactive.
- The receptivity of users is limited when listening to prompts and this often makes it impossible to play more than four to five options at the time [ENG90], [PAA86].
- As a consequence of using digits as input tokens, the semantics of these tokens differ from menu to menu and application to application. This makes it difficult for the users to learn the menu inputs.
- It is difficult to compose menus dynamically at run time

2.3 Enhanced Menu Approach

A common method of enhancing the usability of traditional menu navigation is to extend the vocabulary. Instead of speaking a digit to select a menu item, users can speak a keyword for that item. This means that each menu item has its dedicated keyword. The navigation principle remains the same as the one described in 2.2 with all its limitations.

On the other hand, this technique makes interaction more intuitive. Dedicated keywords for menu items also allow direct access from a given menu to any other menu in an information service. But as a consequence the vocabulary becomes highly application-dependent. Once the application changes, new keywords have to be trained for a new vocabulary.

The enhanced menu navigation approach is application- and domain-dependent. This is probably the most significant obstacle for service providers employing this type of interface. Collecting the speech training data for a given vocabulary is a significant cost factor when developing an IVR service. To reduce costs and to make the vocabulary reusable, the underlying navigation technique of an information service should be application- and domain-independent.

The use of a large vocabulary can be seen as a CISC approach as it is for microprocessors. We believe that small reusable vocabularies, combined with an application-independent navigation method, are more effective. This is more like a RISC approach.[2]

3 The Zap & Zoom Navigation Approach

Zap & Zoom (Z&Z) navigation is list-based. It is designed for easy-to-use telephone applications. In addition, the method is suited for general purpose information systems. The principle of a list-based telephone interface using only key input has been introduced by [RES92] as an extension to conventional menu navigation. Experiments have also been done by Apple to store voice notes on personal devices in a list-based fashion [STI93]. We adapted the idea of the list-based principle for voice recognition and added more features in order to make it more flexible.

3.1 Principle

A Z&Z list consists of single interaction elements. These interaction elements are basic building blocks and are called Z&Z elements. In each Z&Z element the user is prompted to select that item. Users can zap over the Z&Z elements with the *next* or *back* command

---

2 CISC=Complex-Instruction-Set Computer, RISC=Reduced-Instruction-Set Computer

and they can zoom into an item of their interest with the *yes* command. It is a little like watching TV without a program listing. You *zap* through the channels until you find the program you are interested in. Figure 1 shows a Z&Z element on the left side and a traditional menu selection on the right side.

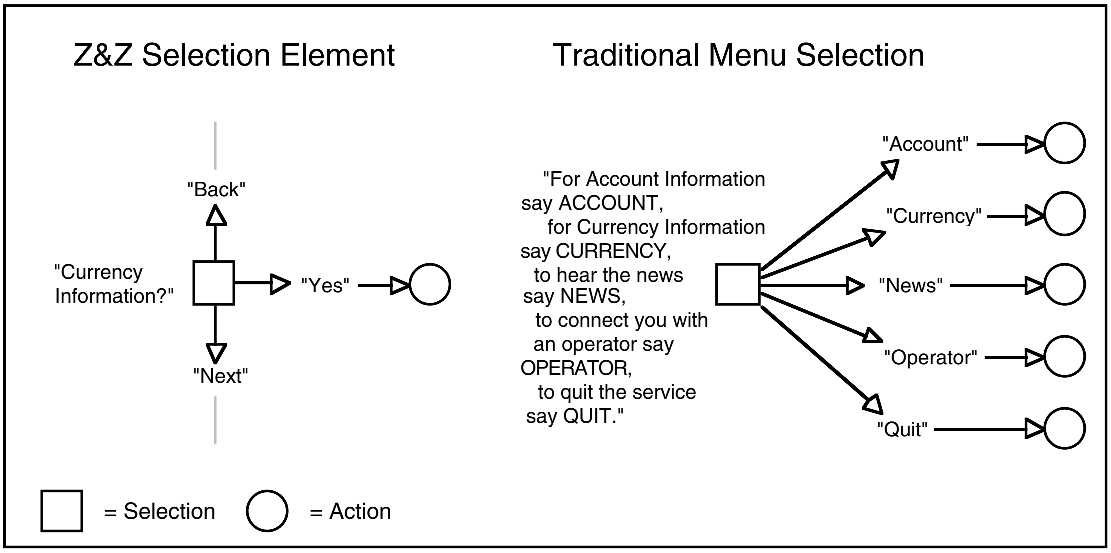

Figure 1: Difference between Z&Z navigation and traditional menu selection.

The selection elements for Z&Z and the menu in figure 1 are atomic units of their underlying navigation principle. The Z&Z element is generic and application-independent while the menu selection is not. A Z&Z list representation of the menu selection in figure 1 using connected Z&Z elements is shown in figure 2.

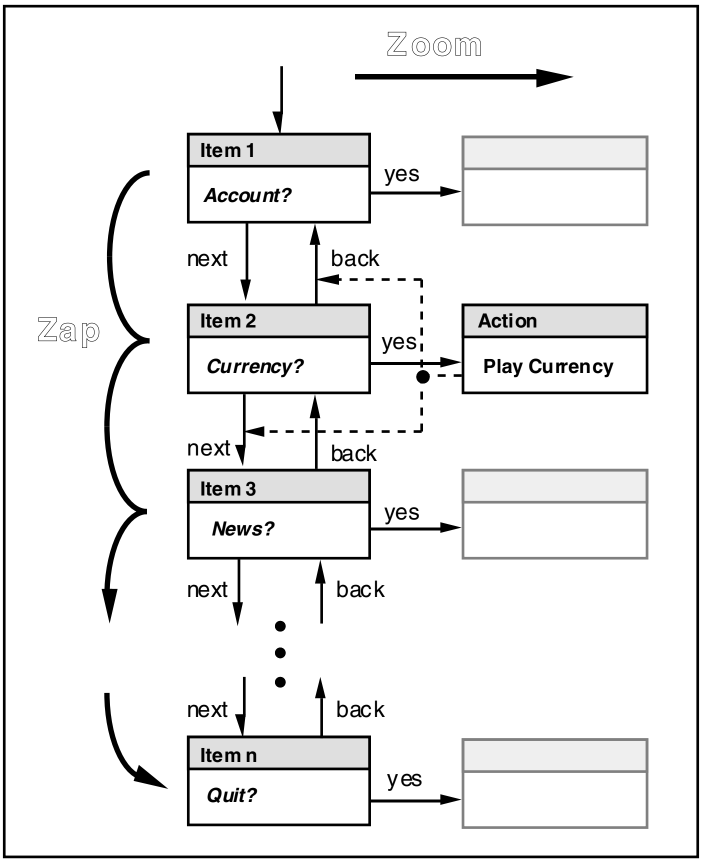

Figure 2: Principle of a Zap & Zoom list.

A selected action automatically moves to the next item when it has finished. To make the dialogue intuitive it is important that next means the next item relative to the user's navigation direction. This avoids unnecessary repetitions of prompts. When the user zaps forwards, the action moves to the "next" item. When the user zaps backwards, the action moves to the "previous" item.

Forcing users to prompt after each item gives them more initiative and allows them to explore the service on their own. Items can be easily connected together at run time.

When moving from traditional menu-based navigation to the Z&Z navigation, three main differences can be summarised which characterise a Z&Z:

- Only one item at the time should be played and users have to answer after each item
- Users can move forward and backward
- The system knows the direction the user is moving

3.2 Commands

The next, back and yes commands do not fulfil all requirements of Z&Z navigation. To add the missing functionality a set of twelve commands is defined. All twelve commands are application-independent and are listed below.

Commands are printed in italic letters. They represent an exactly specified meaning and are place holders for keywords of the voice recognition. A service that supports different languages has exactly the same Z&Z lists and uses simply different keywords as commands. Keywords must be chosen very carefully when training a vocabulary for any language so that they are unequivocal to users [KLO94]. Good keywords can be found only through repeated usability tests [TOG91]. Depending on the performance of the recogniser, different keywords with the same meaning can be trained in order to make the interface more flexible for the users.

Yes	Confirm/Select an item
No	Reject/Go to next item
Next	Go to next item
Back	Go to previous item
Repeat	Repeat last prompt
Overview	Restart at the top level of the service
End/Cancel	Terminate the call/ terminate the current task
Help	Play help message
Key/Voice	Switch from voice input to key input and vice versa
First	Go to first element of a list
Last	Go to last element of a list
Shortcut	Go to a specific item in a list

The *Key/Voice* command is used to select the input mode. This is essential when the service supports key input in addition to voice input. In this way users can choose their most convenient mode or they can switch off the voice recognition in noisy environments.

The *Shortcut* enables direct access to information. It is a compound command starting with the keyword *Shortcut* and followed by an entry that specifies the target item to access. The entry is implementation-dependent. It could be an index for a list element or any application-dependent command as mentioned in 2.3. The shortcut feature makes the interface open for extensions and enables a so-called "expert mode" for experienced users.

It is not necessary to implement all the commands in a Z&Z element. The first eight commands are mandatory for any list based navigation while the last four are only recommended when building more complex information services.

3.3 The Telephone Key Pad as a Remote Control

In order to enable key input, the voice recognition commands have to be mapped to the telephone key pad. In telephone-based information services the key pad is a remote control similar to the ones for television sets, CD-players etc. A common characteristic of such devices is often that each key has its dedicated function. This is the basis idea for the design of the Z&Z key layout too, because an invariant key layout for the navigation makes it easy for the users to learn how to manipulate services. We define the layout on the telephone key pad as shown in figure 3.

1 Overview	2 Shortcut	3 End
4 First	5 Voice/ Key	6 Last
7 Back	8 Repeat	9 Next
* No	0 Help	# Yes

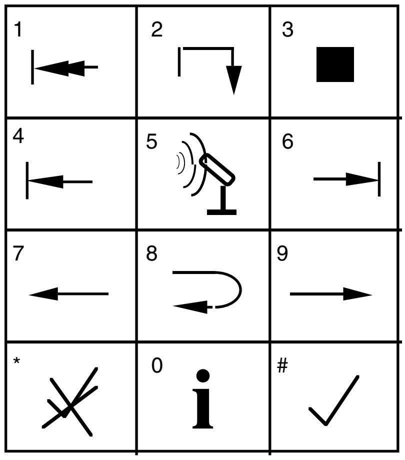

Figure 3: Mapping of Z&Z commands to the telephone key pad.

The layout shown is consistent with other user interaction types. The reason for this layout is to allow consistent data entry interaction where users enter an integer value like e.g. a personal identification number. In a data entry interaction all the digit keys are occupied to enter the value except the star and the hash key. This means that only the latter ones can be used to either commit or reject the data entry phase. We chose the hash key for committing and the star key for cancelling entered values. The remaining commands are then assigned to the digit keys 0 to 9.[3]

Users often can keep a graphical representation of an interface better in mind than a textual representation. Therefore we defined icons representing the Z&Z commands and assigned them to the keys as shown in figure 3 on the right side.

3.4 Design Recommendations

To take full advantage of the Z&Z navigation and to make it easy to use, some design recommendations have to be respected.

- The effectiveness of the Z&Z depends on how the text for the items is formulated. Prompts are shorter and clearer if they are formulated as questions and not as invitations. Using invitations, users must always be told *how* to select an item. A typical example of invitations are menus. Single questions, as they can be used in lists, are obvious to answer - also for novices: they tell only *what* to select. Here is a good and a bad example of a Z&Z prompt:

  Good: "Balance of your second account?"
  Bad: "For the balance of your second account say yes, for the next accounts say next!"

- When reaching the end of a list, mark the top and the end clearly by playing an appropriate message and do not wrap around automatically from the bottom of a list to the top or vice versa. Wrapping around confuses the users and they will lose their way in the service.

---

3 The layout recommendations given in [FRT91] have been respected for commands like Yes, No, Help, and Repeat.

- The system should always provide feedback when the user jumps because this is often a context switch. This is particularly important for commands like Overview, Voice/Key, First, Last, End and Shortcut.
- When using lists that consist of a number of elements of the same type, e.g. a list of currencies or cinemas, always announce the number of items at the beginning of the list like e.g. "Please select from the following 9 cinemas". This provides an overview and makes it easy for the users to decide how to access the list or approximately where in the list the item that they are interested in could be found. Users may access a list linearly through zapping or direct through a shortcut.
- A system with voice input is always error prone. It cannot be guaranteed that what the system recognises is actually what the user said. This should be taken into account when designing error messages [HEL88].
- Users very rarely invoke the help command when they run into problems. Therefore automatic invocation of help messages may increase the usability and acceptance of a system. A typical situation may arise when the system prompts for an item and the user does not remember the command. The system should help the user automatically after a short time-out (typically 2-3s) with a message like: "For yes please press the hash-key, for next press 9" in the case of key input.
- Building complex services is a challenging task. More than two hierarchical levels are too difficult to understand for most occasional users. We avoid hierarchical levels by splitting the information into several services or by separating multiple language information services into a service for each language. Separate telephone numbers are used for different services.
- It has been shown that lists with up to twelve items can be handled without difficulty [RES92]. Lists with less than four items offer no advantages in terms of access time against menu navigation when using the service for the first time. Individuals also expressed a strong preference for the list-based navigation method over the traditional method in an experiment comparing the two methods.

These recommendations are the result of more than one year of experimenting with the Z&Z and a series of usability tests. The aim was to build information services for occasional users, which implies users cannot be trained or supervised. This strongly affects the design and limits the complexity of a service.

4 Design Model for Applications

A real information service cannot be built only with the proposed Z&Z navigation. Rather it consists of different types of interactions and system actions as building blocks where Z&Z is one of them. We define and classify a set of such building blocks for building complete services.

4.1 Building Services with Nodes

Any telephone-based information service is a sequence of user interactions and system actions and can be modelled as a finite state machine (FSM). The states of the FSM represent black boxes of complex user interactions such as a Z&Z list or an integer input. We call theses black boxes nodes. A service can then be described by a two dimensional transition table containing the source nodes in one direction and the target nodes in the other. From each node users may move to multiple targets (other nodes) according to their input (user event). Other conditions for moving to specific nodes can be the evaluation of database queries or any exception handling like user errors, time-outs, hardware or application errors (system events). The FSM of a service can be represented as a directed graph in which the vertices are the nodes and the edges are the user or system events. This representation is suitable not only for a graphical representation of a service but also for checking the correctness of dialogue flows through connectivity.

4.2 Classifying Nodes

A service graph consists of different types of independent nodes. These nodes are classified into types according to their use and their behaviour as shown in figure 4.

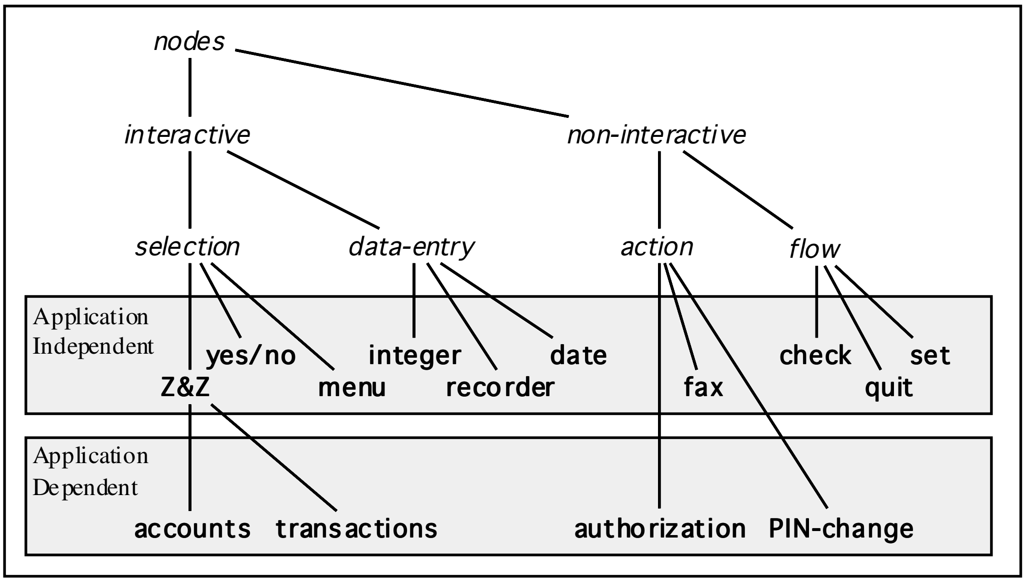

Figure 4: Node Classification.

Nodes printed in **bold** letters are actual implementations and nodes printed in italic letters are abstract nodes for classification only. According to their behaviour, abstract nodes are split into *interactive* and *non-interactive* nodes. From the point of view of the user interface the interactive nodes are the most relevant. Interactive nodes are classified according to two fundamentally different types of interaction: *selection* and *data-entry*. Non-interactive nodes perform actions and control the flow of a service and are "invisible" to users. All actual nodes are either application-independent or application-dependent. Application-independent nodes can be reused in different services.

The selection consists of three basic types. The most important for us is **Z&Z** navigation. Others are the traditional **menu** selection and the so called **yes/no** dialogue which is actually a mutilated menu selection that represent a simple boolean selection.

Data entry nodes allow the users to enter numerical or speech data. Three types have been implemented: **integer** value input, **date** input and a message **recorder**. Entering integer values is the basic data input interaction used in nearly all IVR applications. The date input is a more complex version of the integer input and can be seen as a structured entry form [RES93a],[RES93b] combining three integer inputs for the month, day and year value. The message recorder is used to capture non-structured spoken information from users. All three types can repeat the entered data for confirmation and correction.

The confirmation for numerical values is done by repeating the value(s) followed by a Yes/No dialogue. When repeating any value, the system can speak it as:

- an ordinary concatenated number
- a spelled integer
- a currency with its units
- a date with weekday, monthday, month and year
- a phone number consisting of a sequence of one- and two-digit numbers

The number of possible corrections of a value in a node is limited to three iterations

5 Application Experiences

Two information services were implemented. One is a banking service for account information inquiry and another is a service for inquiring about the Chinese horoscope combined with fax output. Both allowed us to test the usability of Z&Z and the voice recognition vocabularies. Only the banking service is discussed here.

5.1 Implementation of Nodes

Nodes are implemented as finite state machines themselves. They are modelled as independent generic units in order to provide reusability. A generic interactive node is built to handle all possible user scenarios of that particular type. These scenarios determine the model of the node. It incorporates input prompts, input capturing, input confirmation, local help, user errors and time-outs.

The run-time behaviour of nodes is determined through parametrisation of the generic behaviour. Parameters are defined statically and loaded at run time. They parametrise target node names, messages, user input formatting and message output formatting. User input formatting contains the number of input tokens, the type of input device (microphone or key pad), the vocabulary, time-out times, type and range checking of input values, etc. Message output formatting contains message IDs, the volume of messages, the format of messages (text-to-speech, digitised speech) the behaviour on user inputs (interruptible) and so on.

All interactive nodes support three different levels (1-3) for user errors and user time-outs. An occurrence of an error or time-out leads to an increment in the error or time-out level. The first level invokes a feedback message corresponding to the error or time-out and the user is then prompted for input again. After a second error or time-out a help message is automatically played as well as a feedback message as in the first level. At the third level the system plays a final message and moves to an error or time-out target. A correct input resets both levels to 1. This method provides a simple and efficient way to re-enter or correct inputs when user inputs are either unexpected or missing [FRA93].

5.2 Voice Recognition

Our voice recognition is done with a commercial speaker-independent recogniser for isolated words. Vocabularies have been trained for the three languages, German, French and Italian, on a homogeneous distributed Swiss population of about one thousand native speakers for each language. The individuals had to read the words from a list in given time intervals synchronised by playing a beep before each word. The words were recorded over the telephone network on a digital audio tape during the session. They were then copied to disk for pre-processing, indexing and training. Individuals were not supervised during the recording session. The individuals had to read the words from a list. Each word occurred twice on the list and the words on the list were not in any particular order because words at the end of a sampling session are often spoken more accurately than at the beginning of a session due to the learning effect. Each vocabulary is subdivided into two sub-vocabularies: one for the navigation mode containing the Z&Z commands and one for the data entry mode containing the digits zero to nine, ok and cancel. Only one sub-vocabulary can be active at any given time.

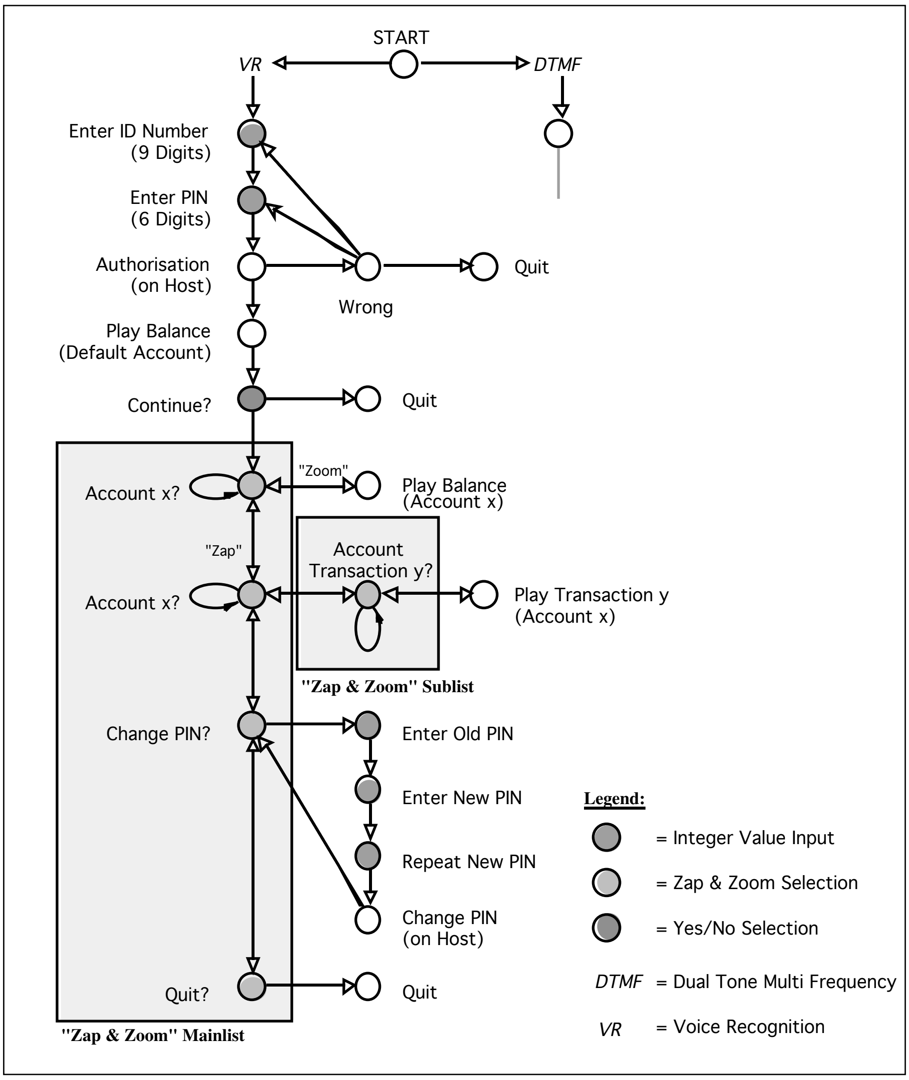

Figure 5: Simplified Service Graph of a Phone Banking Service.

5.3 Example of a Phone Banking Service

Figure 5 shows a simplified graph of a phone banking service. It contains mainly the interactive nodes and few application dependent nodes. The service is symmetrical for voice and key input. Prompt messages, time-out times, and input confirmations are different for the two input modes.

Users are firstly prompted to select either key input or voice input. If DTMF is detected the users can switch dynamically from key to voice input and vice versa. No input key (DTMF signal) restricts input to voice. Then the users are prompted for a nine-digit customer number and a six-digit personal identification number (PIN). After authorisation on a host computer the system announces the number of accounts and plays automatically the balance of the first account, namely the user's transaction account. Users are then prompted for more information with a Yes/No dialogue.

If users select "more information" they enter the Z&Z main-list. A short introduction (overview) at the beginning of the list explains how to navigate using the most important commands like yes, next, and back. The main-list then contains a Z&Z node to select from a list of a variable number of accounts to play the balance. It is followed by a same type of node to select accounts for account transactions. Zooming into account transactions of a particular account offers a list with the last five transactions. The number of account transactions available is always announced at the beginning of the sub-list. Both the number of accounts and the number of account transactions are known at run-time after user authorisation. The last two items in the main list allow the user to change the PIN-code and to end the call.

5.4 Experiences

Users encounter most difficulty at the first interaction point in the service, e.g. when selecting the voice or key input mode. It required several iterations to improve these prompt messages. Once the users arrived at the Z&Z main-list, they never had interaction problems.

In a first version of the service, most people preferred the key-input mode. This had several reasons:

- Both, customer number and PIN had to be repeated for confirmation.
- Yes/No interactions did prompt with an invitation rather than a question.
- All transaction for an account were played as one unit, no selection was possible.
- The system did not recognise the navigation direction.

After correcting the problems the preference has changed to voice input

Users tend to say *no* instead of *next* for navigating in lists even if *no* was never introduced verbally. When formulating input prompts as questions both commands should therefore have the same meaning.

In a first field test in a real environment, we had 1060 calls, 11252 utterances, and 1834 non-recognised words with the Swiss-German vocabulary. This corresponds to a total recognition error rate of 16.3% and a theoretical mean recognition rate per word of r_w=98.3%[4]. Possible word confusions are not included. We can say already that users accept error-prone voice recognition as long as the system is fault-tolerant. The accuracy of all three vocabularies is being investigated in greater detail.

All nodes have built-in logging capabilities for user and system events. Events can be individually enabled or disabled for logging either for the entire system or for a specific node. Logging information is written to files. This permits off-line analysis of how the system is used. Logfile analysis provides quantitative statements about the usage of the system [NEL94]. Event logging was an important instrument to improve and optimise our information services during pilot phases.

We can confirm that new nodes cannot be designed and implemented in one step. Implementing a specific type of node and defining the necessary parameters was an iterative task [TAT93]. Nodes must be reused in various applications and situations in

---

4 $r_w = r_t$**(n_c/n_u) where r_t = total recognition rate, n_c = number of calls, n_u = number of utterances.

order to improve their usability and to test their reusability. This required several redesign cycles as is not uncommon for the development of object oriented systems [GAM92].

6 Conclusion

We are using Z&Z navigation as an alternative to traditional menu navigation. It has been shown that the recognition of a small vocabulary of isolated words is sufficient for building an easy-to-use and efficient voice interface when a domain- and application-independent navigation technique is employed. Voice recognition vocabularies were trained for German, French and Italian. They are reusable and this leads to a cheaper development of telephone voice information systems. Two services using Z&Z were implemented. Usability test allowed significant improvements to the user interface. Combining voice and key input provides flexibility for users and independence of telephony communication standards.

7 Acknowledgements

Jose Clarinval offered invaluable help for collecting the word samples for the vocabulary training and contributed many ideas throughout the course of this project. Phat Tran and Yvan Bourquin implemented essential parts of the system. Kai-Uwe Mätzel, Thomas Eggenschwiler, Hans-Peter Frei, Patrick Steiger, NicolasLéwy, and James Crawford also contributed to the ideas and presentation of this paper.

References

[DET90] Detweiler M, Schumacher R, Gattuso N: Alphabetic Input on a Telephone Keypad. In *Proceedings of the Human Factors Society*, 34th annual meeting. Santa Monica, CA: Human Factors Society, 1990

[ENG90] Engelbeck G, Roberts T: *The Effects of Several Voice-Menu Characteristics on Menu Selection Performance*. Technical Report ST0401, US West Advanced Technologies, 1990

[FRA93] Frankish C, Noyes J: Feedback in Automatic Speech Recognition: Who is saying what and to whom. In Baber C, Noyes J: *Interactive Speech Technology*. Taylor & Francis, pp. 121-130, 1993

[FRM93] Franzke M, Marx A.N, Roberts T, Engelbeck G: Is Speech Recognition Usable? An Exploration of the Usability of a Speech-Based Voice Mail Interface. *SIGCHI Bulletin*, Vol. 25, No. 3, July 1993

[FRT91] France Télécom: *Recommandations aux Partenaires des Services Vocaux Téléphoniques*. CNET Paris - A, 38/40 avenue du Général Leclerc, 92131 Issy les Moulineaux - Cedex, France, 1991

[GAM92] Gamma E: *Objektorientierte Software-Entwicklung am Beispiel von ET++*. Springer Verlag, 1992

[HAL89] Halstead-Nussloch R: The Design of Phone-Based Interfaces for Consumers. In *Proceedings of CHI'89*, Conference on Human Factors in Computing Systems, Austin Texas, TX: ACM, pp. 347-352, 1989

[HEL88] Helander M, Moody T, Joost M: System Design for Automated Speech Recognition. In Helander M: *Handbook of Human-Computer Interaction*, pp. 301-319, 1988

[KLO94] Kloosterman S: Design and Implementation of a User-Oriented Speech Recognition Interface: The Synergy of Technology and Human Factors. *Interacting with Computers* Vol. 6 No. 1, pp. 41-60, 1994

[NEL94]. Nelson W: Analyzing User Interactions with Hypermedia Systems. In *Computer Graphics*, SIGGRAPH, Vol. 28, No. 1, pp. 43-45, 1994

[PAA86] Paap K, Roske-Hofstrand R: The Optimal Number of Menu Options per Panel. *Human Factors*, Vol. 28 No. 4, pp. 377-385, 1986

[PEL93] Pelton G: *Voice Processing*. McGraw-Hill, pp. 186-191, 1993

[RES92] Resnick P, Virzi R: Skip and Scan: Cleaning Up Telephone Interfaces. In *Proceedings of CHI'92*: Conference on Human Factors in Computing Systems. Monterey, CA: ACM, pp. 419-426, 1992

[RES93a] Resnick P: Phone-Based CSCW: Tools and Trials. *ACM Transactions on Information Systems*, Vol. 11, No. 4, pp. 401-424, 1993

[RES93b] Resnick P, Virzi R: *The Design Spaces of Audio Menus and Forms*. MIT Sloan School Working Paper #3633-93. MIT Center for Coordination Science Technical Report #159, 1993

[STI93] Stifelman L, Arons B, Schmandt C, Hulteen E: VoiceNotes: A Speech Interface for a Hand-Held Voice Notetaker. *Proceedings INTERCHI'93*. Amsterdam, NL: ACM, pp. 179-186, 1993

[TAT93] Tate M, Webster W, Weeks R: Evaluation and Prototyping of Dialogues for Voice Applications. In Baber C, Noyes J: *Interactive Speech Technology*. Taylor & Francis, pp. 157-165. 1993

[TOG91] Tognazzini B: *TOG on Interfaces.* Addison-Welsey, pp. 79-89, 1991