



Evaluating Cost Effectiveness and Usability of Telephone User Interfaces

Bernhard Suhm, Patrick Peterson

BBN Technologies
10 Moulton Street
Cambridge, MA 02138

+1 617 873 3200
{bsuhm, patp}@bbn.com

ABSTRACT

Telephone user interfaces are the most widespread class of human-computer interfaces. However, usability evaluation and engineering methods for telephone interfaces are not well developed. This paper presents an assessment methodology that evaluates both cost effectiveness and usability of telephone interfaces based on a detailed analysis of end-to-end recordings of thousands of calls. Since agent time is the major cost driver in call center operations, we quantify cost effectiveness in a single number that measures how much agent handling time a telephone interface saves. To quantify usability, we refine task completion into a set of IVR automation rates, and we represent the complete traffic through an IVR as a tree (called a user-path diagram). Beyond evaluation, our methodology has important implications for telephone user interface design. Assessment results provide concrete guidance on how to improve the interface. Furthermore, the benefit of a redesigned interface can be estimated, thus providing the business justification for telephone interface usability reengineering.

INTRODUCTION

Introduced more than a decade ago, touch-tone interactive voice response (IVR) systems were adopted enthusiastically in many call centers to provide customer service efficiently. Consumers, on the other hand, hate touch-tone IVRs because they are difficult to use and they seem to delay the process of getting to live agents. This dichotomy is not surprising considering that most call centers focus on cutting operating costs and that usability and its impact on call center operations is poorly understood. This paper attempts to overcome the impasse by presenting a methodology for evaluating both usability and cost effectiveness of telephone user interfaces. Our assessment methodology provides usability practitioners with tools to identify and quantify usability problems in telephone interfaces, and it provides call center managers with the business justification for the cost of usability engineering.

What is the state-of-the-art in evaluating telephone user interfaces? Obviously, since touch-tone IVR systems have been deployed for more than a decade, a significant body of know-how on IVRs has been accumulated in the industry. Except for recent attempts to define a style guide for (telephone) speech applications [1] and introducing universal commands in speech-enabled IVRs [2], this body of knowledge is not well documented. The prevalence of usability problems in deployed IVRs suggests that designing good telephone interfaces is difficult and usability-engineering methods for telephone interfaces are not well developed.

Another area of related work is research on spoken dialog systems, an important application of speech recognition technology. *Spoken dialog systems* allow the caller to communicate with the system in spoken dialogs, not necessarily over the telephone. While many research spoken dialog systems have been published (e.g., [3-7]), previous work on spoken dialog system evaluation focused on quantifying the performance of the underlying technologies (e.g., [6, 8, 9]). Some studies evaluated usability of telephone interfaces based on task completion rates and post experimental questionnaires [10]. More recently, PARADISE [11] was introduced as a "consistent integrative framework for evaluation" of spoken language systems. Basically, it provides a method to identify measures that predict user satisfaction well, from the large set of measures that have been used in the field. However, this work did not address the cost for the call center, nor does it provide any guidance for telephone interface redesign.

This paper presents an assessment methodology for telephone interfaces that evaluates both usability and cost-effectiveness, and that is applicable to both touch-tone and speech-enabled IVRs. In the sections that follow, we describe the data capture process, present our methodologies for evaluating cost-effectiveness and usability, and show how those methodologies have been applied to the design of telephone user interfaces.

DATA CAPTURE

In telephone user interfaces, the only complete record of user and system behavior is in complete calls. Therefore, a comprehensive usability assessment of telephone interfaces must be based on end-to-end recordings of calls. A call typically begins with a dialog with an automated (IVR) system, called *IVR section* in this paper, which may be followed by a dialog with a live agent, called *agent-caller dialog*.

Recordings of complete calls represent a large amount of data that is difficult to analyze in raw form. To make the analysis of call data efficient, we transform the recordings into a detailed trace of significant events for each call. Significant events in the IVR section include system prompts and caller input, either touch-tone or speech. In the agent-caller dialog, we look at events such as exchange of various kinds of information (account numbers, dollar amounts, etc.) and description of the caller's problem (question about a bill, inquiry into flight schedules, etc.).

While most of our analyses are based on this event sequence, the ability to switch between call recording and its representation as event sequence is crucial throughout the analysis process. To actually capture the events from the different segments of a call, we use a variety of methods, which we describe next.

Touch-tone IVR Events

The preferred method for capturing the IVR event sequence would be an event log that is generated by the IVR. However, the reports that current IVR platforms generate are usually inadequate and inaccurate. They are inadequate because they are typically based on "peg counts", which indicate how many times a prompt or menu was visited overall, but provide no information on specific calls. Peg counts are unable to identify even the most basic usability problems, such as callers being trapped in "voice mail jail". To illustrate IVR report inaccuracy, consider the reporting of hangups in the IVR section. Without knowing if a specific call actually accomplished anything, these reports have to make arbitrary decisions about whether to treat a hangup as a "self-serve" or an "abandon", which have very different values to both the customer and the call center.

While it is possible to customize IVR reports to include event traces for calls, this requires that the IVR code write to an event log at appropriate states in the call. Generating such code is error-prone and intrusive to call center operations. Therefore, we have designed our process to develop the complete IVR event sequence from the call recording alone. By recording calls remotely, the event sequence can then be captured in an unobtrusive fashion.

Our IVR analysis employs three main tools, a DTMF detector, a prompt detector, and a prompt inference tool, to capture the event sequence in three steps. First, we use a commercially available DTMF detector to detect touch-tones. Next, our prompt detector detects important known prompts in the recordings. Then, whenever the IVR is so complex that detection of all prompts would be impractical, we employ a prompt inference tool to efficiently infer the complete prompt sequence.

An additional, crucial step is to determine the exit condition for the IVR section of the call. The exit condition indicates whether the call ended in the IVR with a hangup or whether it was transferred to an agent. We detect IVR transfer prompts, such as "Please wait for the next available representative", to determine whether a call was transferred to an agent. If the prompt detector detects the transfer prompt, we infer that the call was transferred to an agent. Otherwise, we assume that the caller hung up. This method of inferring the IVR exit condition fails when the caller hangs up during the hold time, before reaching an agent. However, such cases can be corrected during the transcription analysis, which is described below.

Speech-enabled IVR Events

We follow a similar process to capture events in speech-enabled IVRs, with the following two modifications.

First, the analysis must rely on prompt detection to disambiguate the event sequence after any speech input from the caller. Unlike touch-tone IVRs, where we can reliably recognize user touch-tone input, recognition of user speech input is error-prone. Thus the state transition after speech input cannot be inferred reliably from the speech alone.

Second, to evaluate speech recognition performance, all segments of a recording that contain user speech must be identified, and the true sequence of words on those spoken segments must be annotated manually, using human transcribers.

Agent-Caller Dialog Events

Our *transcription analysis* captures the sequence of significant events for anything that follows the IVR section in a call, i.e., waiting on hold and agent-caller dialogs. Significant events include start of the agent-caller dialog, exchange of information between caller and agent (such as account numbers, amounts, dates), and the reason for the call. In addition, the transcription analysis may characterize the call as a whole with attributes such as the degree to which the call was resolved, and agent courtesy. We currently employ human transcribers to perform these annotations, hence the name transcription analysis.

EVALUATING COST EFFECTIVENESS

Agent time tends to dominate costs in most call centers. The ratio between cost of agents and all other costs, such as telecom time, and IVR hardware and software costs, is at least 4:1. Therefore, our assessment methodology

quantifies the cost effectiveness of an IVR in terms of agent time. We define the *total IVR agent-time benefit* as the agent time that is saved by the IVR compared to routing all calls directly to a non-specialized "floor" agent.

An IVR "saves" agent time whenever it successfully performs tasks that otherwise would have to be performed by an agent. Tasks that typically can be performed within an IVR include identifying the caller, providing information to the caller, performing transactions, or routing the caller to specialized agents. In some cases, completing these tasks successfully may resolve the call so that the caller hangs up without any assistance from an agent. We refer to such calls as *self-serve* or *full automation*. It is important to note, however, that even if a call is not fully automated, the IVR can still provide significant savings through partial automation. Table 1 shows typical agent-time savings by automatable task. These savings can be derived from benchmark assumptions or measured in annotated agent-caller dialogs. While the emphasis in this context is on cost, we note that IVR automation rates correspond to sub-task completion rates. Hence, IVR automation is a more differentiated version of the standard task-completion usability measure.

Automated Task	<i>Caller Identification</i>	<i>Information Delivery</i>	<i>Routing</i>
Saved Agent Seconds	15	50	40

Table 1: Typical Agent-time Savings for Automated Tasks

We implement the basic principals of measuring cost effectiveness in terms of agent time saved at the task level, by first quantifying IVR automation and then calculating an overall benefit measure, as described next.

Quantifying IVR Automation

IVR automation can be measured based on call event-sequence data. We call this process *IVR automation analysis*, and it starts with the definition of IVR-automatable tasks, as in Table 1. Typically, the completion of a task can be associated with reaching a certain state in the IVR. Thus, the set of completed tasks can be inferred directly from the event sequence data for a call, using a simple lookup table that documents which IVR states correspond to the completion of which tasks.

We make one important exception to the assumption that IVR states indicate successful task completion. Specifically, we do not assume that routing decisions made in the IVR are necessarily correct. Rather, we look at subsequent agent-caller interactions to determine, based on the annotated reason for a call, whether the call was correctly routed or *misrouted* to an agent. Calls that misroute to specialists usually need to be transferred somewhere else and, therefore, incur a cost equal to the time it takes the specialist to reroute the call, which can be thought of as a negative routing benefit.

Given the definition of tasks that can be completed within an IVR, we characterize the automation achieved in a call according to distinct combinations of automated tasks, which we refer to as a *call profiles*. Given a set of calls with their event sequence data, we accumulate counts for each call profile by annotating every call with its set of completed tasks. The call traffic into an IVR is thus partitioned into a set of call profiles, each representing a distinct pattern of automation.

Automation rates are defined as the percentage of automation achieved over all calls for each automatable task. This percentage can be calculated simply by adding the percentages of all call profiles that include the specific automatable task.

Table 2 shows an example IVR automation analysis, in which we distinguish two agent types, "specialist" and "floor". The left column lists the call profiles. The next two columns (labeled "Traffic") show the breakdown of the total data set, which consists of 3636 calls, into the various profiles. For example, 2% of the calls were fully automated, and 18.7% abandoned without getting anything done. Then, the three "Automation" columns show the automation categories for each profile. The analysis is based on three automation categories: "A" for capture of the caller's account number, "R" for routing, and "I" for delivery of information. For example, the profile "Transfer to floor after info readout" achieved capture of the account number ("A") and automated delivery of information ("I"). Note that misrouted calls incur a "negative R" benefit, or misrouting cost ("-R").

Call Profile	Traffic		Automation			Benefit [agent secs]	
	Calls	% Calls	Account #	Routing	Info Delivery	one call	Average
<i>Fully-automated, self-serve</i>	72	2.0%	A	R	I	105	2.1
<i>Transfer to specialist after info readout</i>	1	0.0%	A	R	I	105	0.0
<i>Transfer to floor after info readout</i>	38	1.0%	A		I	65	0.7
<i>Transfer to specialist w/ ID</i>	849	23.4%		R		40	9.3
<i>Transfer to floor w/ ID</i>	1008	27.7%	A			15	4.2
<i>Transfer to floor w/o ID</i>	591	16.3%					
<i>Misroute to specialist w/ ID</i>	389	10.7%	A	-R		-25	-2.7
<i>Misroute to specialist w/o ID</i>	6	0.2%		-R		-40	-0.1
<i>Abandon</i>	681	18.7%					
Totals	3636	100.0%	41.5%	14.5%	3.1%		13.6

Table 2: IVR Automation Analysis, with two agent categories (“specialist”, “floor”).

The bottom row, for the three “Automation” columns, shows the automation rates by category: 41.5% capture of account number, 14.5% routing, and 3.1% information delivery. The 14.5% net automation rate for the routing task is composed of 25.4% correct routes and 10.9% misroutes, which indicates that IVR routing is not very accurate.

Calculating Overall IVR Benefit

For each call profile in Table 2, we can calculate the agent-time saved for a single call by adding up the agent-time savings for each automated task, using the data from Table 1. The two columns labeled "Benefit" in Table 2 show the IVR benefit for each call-profile. The left column shows the benefit for one call and the right column the average benefit, relative to all 3636 calls that were evaluated. Because the benefits in the last column have the same basis (i.e. all calls) they can be added to derive a total IVR benefit of 13.6 agent seconds saved, shown in the lower right-hand cell of this table.

EVALUATING USABILITY

Evaluating usability typically encompasses quantifying usability, identifying usability problems, and evaluating subjective usability factors. Our assessment methodology currently quantifies usability and provides methods to identify usability problems, but we do not (yet) formally evaluate subjective usability factors, such as user satisfaction.

Common usability measures include task completion rates and task completion times. As shown above, our IVR automation analysis provides task completion rates in a form that is suitable to the problem of evaluating telephone user interfaces. Clearly, automation analysis can also be used to quantify usability of telephone user interfaces. More specifically, low automation rates point to usability problems. In the example above, we can see that low routing accuracy points to a severe usability problem in this IVR.

In addition to automation analysis, we have developed a number of other tools for evaluating usability, specifically *user-path diagrams* and *life-of-call timing diagrams*, which we describe next.

User-path Diagrams

User-path diagrams effectively visualize user behavior in the IVR by representing event sequence data as a tree. To manage the complexity of user-path trees, we define *IVR states* as IVR tasks or subdialogs, such as ID entry or menu selection, that may encompass many events and multiple IVR-caller interactions in the captured sequence data.

The nodes of the tree correspond to IVR states, arcs correspond to state transitions, and leaves correspond to end-conditions of calls. Each node and leaf is marked with the count of calls that reached the node or leaf, and with two percentages: the absolute percentage, relative to all calls in the data set, and the percentage relative to all calls that reached the parent of the node. In addition, arcs may be marked with the user input that causes the corresponding state transition, such as pressing a certain touch-tone in response to a prompt. We found it helpful to distinguish at

least three end conditions. “Self-serve” are calls that resolve in the IVR. “To-agent” are calls that transfer to a live agent. “Abandon” are calls where the caller hangs up, either in the IVR without obtaining any useful information, or on hold before reaching a live agent. If the call center operates with distinct categories of agents, the “to agent” category is typically broken down into various subcategories, each representing a distinct routing destination from an operational point of view.

Figure 1 shows an excerpt from a user-path diagram. Rectangular boxes represent IVR states, arrows represent call traffic, and circles indicate places where calls leave the IVR. For example, of the 4319 calls that this data set contains, 234 calls (or 5.4%) abandon at the greeting. At the opening menu, 311 calls (or 7.2%) are transferred to a "floor" agent, claiming they want to establish a new account. At the same menu, 4.3% of the calls reach a "floor" agent for other reasons, and 1.1% abandon the call.

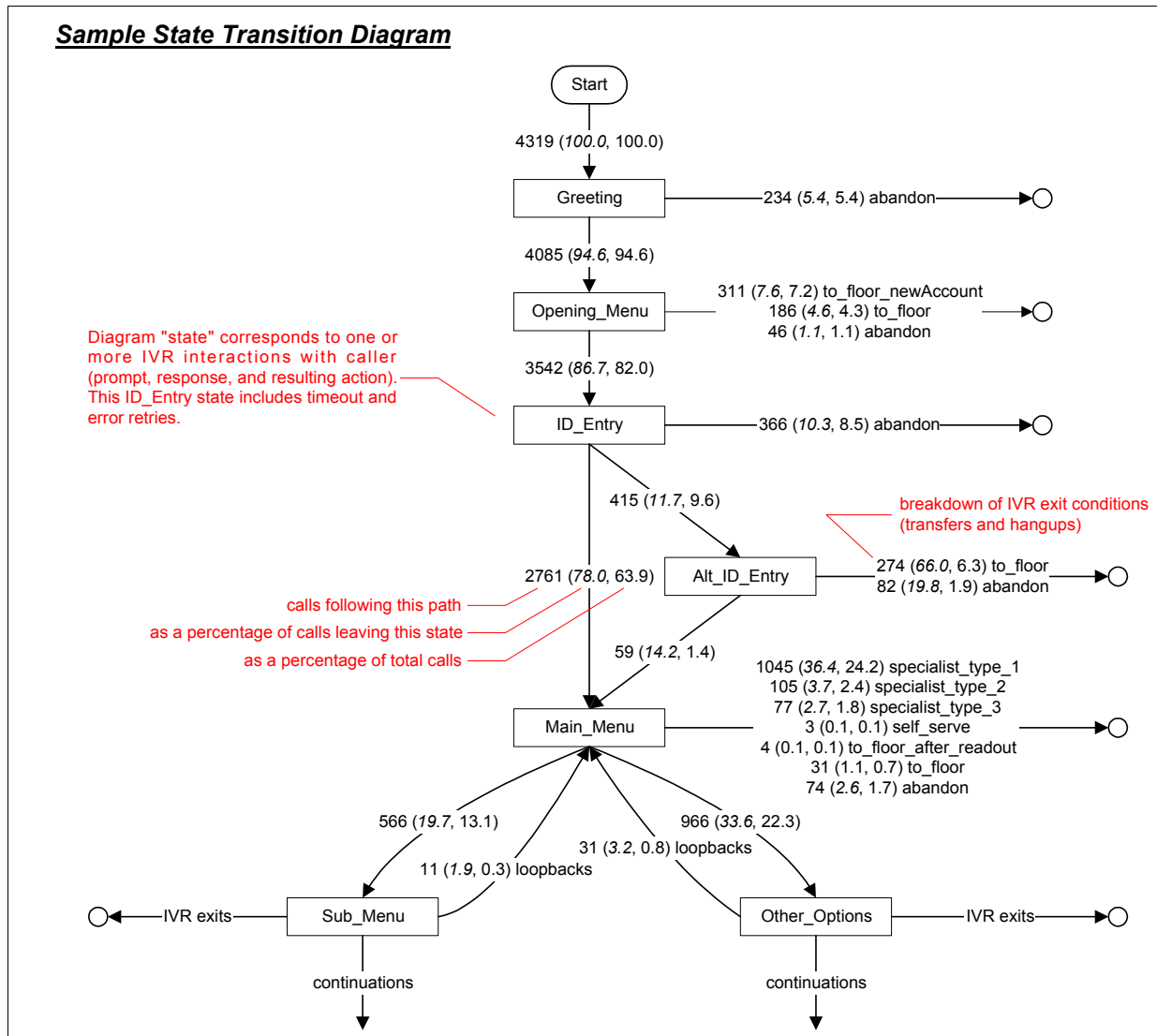


Figure 1: Excerpt from a User-path Diagram

Inspecting user-path diagrams can identify usability problems. Parts of the tree that receive little or no caller traffic, or states with high rates of abandoned calls or transfers to an agent, point to usability problems. In Figure 1, for example, the state cluster named "ALT ID Entry" receives 9.6% of all calls, but 86% of these calls either abandon or are transferred to a floor agent, and the account number is correctly entered in only 14%. Obviously, this part of the IVR is ineffective.

User-path diagrams also lend themselves to analyzing the effectiveness of sections of an IVR, such as the section that identifies the caller. We found that the success rate (or *yield*) of the IVR on specific sections is a useful usability measure. The yield of an IVR section is defined as the ratio of incoming to outgoing calls for the cluster of states that represents the IVR section. For example, the yield for identifying the caller for the IVR shown in Figure 1 is $(2761+59)/3542=79.6\%$, by adding appropriate counts for the “ID Entry” and “ALT ID Entry” states.

Life-of-Call Timing Diagrams

To conduct timing analyses of telephone user interfaces, we measure completion times from the call event traces, which contain time stamps for every significant event in a call. Timing analyses can be conducted at various levels of detail, ranging from timing broad sections of a call, such as IVR, on hold, and agent-caller dialog, to measuring task completion times for specific tasks, such as entering an account number.

Useful insights about the customer experience within a telephone user interface can be gained by tabulating the timing of broad sections for different call types. We use the term *life-of-call diagram* to refer to these call timing profiles presented in bar-chart form. Figure 2 shows a typical diagram. We find it most useful to identify call type based on the reason for the call as annotated in the caller-agent interaction. In Figure 2 calls going to an agent are broken down into four call-type categories, based on the type of agent best suited to handle the callers true problem. The diagram shows that the call center can service calls of type A much faster than type C. The profile “Hangup during transfer from IVR” shows that callers' patience during a transfer is typically exhausted after 90 seconds of waiting on hold.

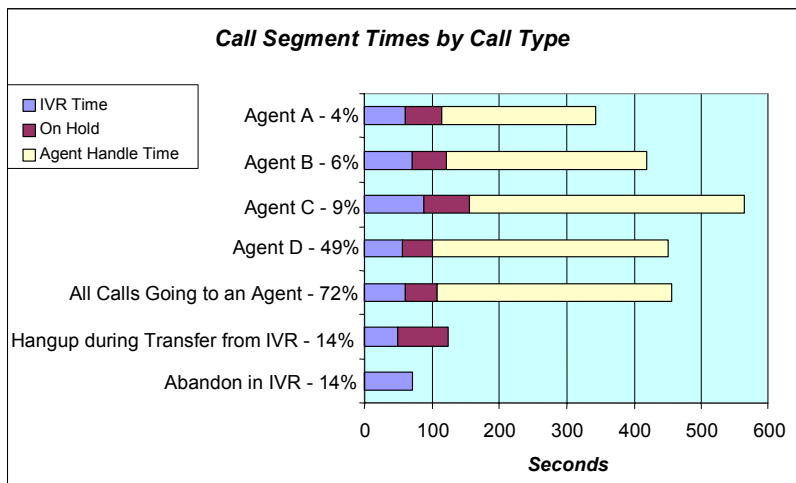


Figure 2: Life-of-call Timing Diagram.

APPLICATION TO TELEPHONE USER INTERFACE DESIGN

In the previous sections, we introduced our telephone user interface assessment methodology along with supporting analysis tools, such as IVR automation analysis, IVR benefit analysis, user-path diagrams, and life-of-call diagrams. Beyond evaluation, our assessment analyses also provide guidance for improving the interface design, which we will discuss next.

Assessment Quick Hits and Reengineering

In the course of evaluating existing telephone user interfaces, we usually observe a number of usability problems in enough detail to diagnose the problems and recommend solutions. We refer to solutions that are clear-cut and non-controversial as *quick hits*. When the diagnosis is clear but the solution is not, we frequently recommend call flow *reengineering*, where alternative designs are tested side-by-side for efficacy.

We have encountered many obvious quick hits, some more than once. For example, in a few cases we observed a suspiciously large proportion of callers being bumped out of a touch-tone numeric entry task with numbers that had too few digits. By listening to call recordings, it became obvious that some callers were struggling to enter long digit strings and were being cut off before they were able to complete their entry. The solution was to increase the interdigit timeout parameter. Another quick hit example is in touch-tone menu design, where our detailed analysis

of routing can point to particular menu choices that are not effective and can be improved with simple changes in prompt wording.

There are also a number of typical problems where reengineering is called for. Touch-tone menus are always problematic, but we have found that it is usually possible to make significant, quantifiable improvements in performance with careful design. We also find that speech and natural language technology, which lets callers avoid menus by describing problems in their own words, can outperform touch-tone. Another problem that can be addressed with reengineering is call flow "leakage", or the tendency for callers to bail out of the call flow at the earliest opportunity.

Comparative IVR Analysis

As part of the reengineering process, we typically evaluate alternative designs side-by-side with real traffic. For each design we measure automation rates and calculate IVR benefit. Differences in automation rates indicate which IVR design is better for each automatable task. For overall comparisons, differences in total IVR benefit reveal which design is superior on the whole. Comparative IVR analysis can thus validate that a new IVR design is indeed better, and furthermore, it can quantify the cost savings.

As an example, Figure 3 compares four IVR designs: an initial touch-tone baseline; a quick hit touch-tone design; a reengineered design representing a practical upper limit for touch-tone; and a speech-enabled design that uses BBN's Call Director natural language processing. The height of the columns indicates the total IVR benefit.

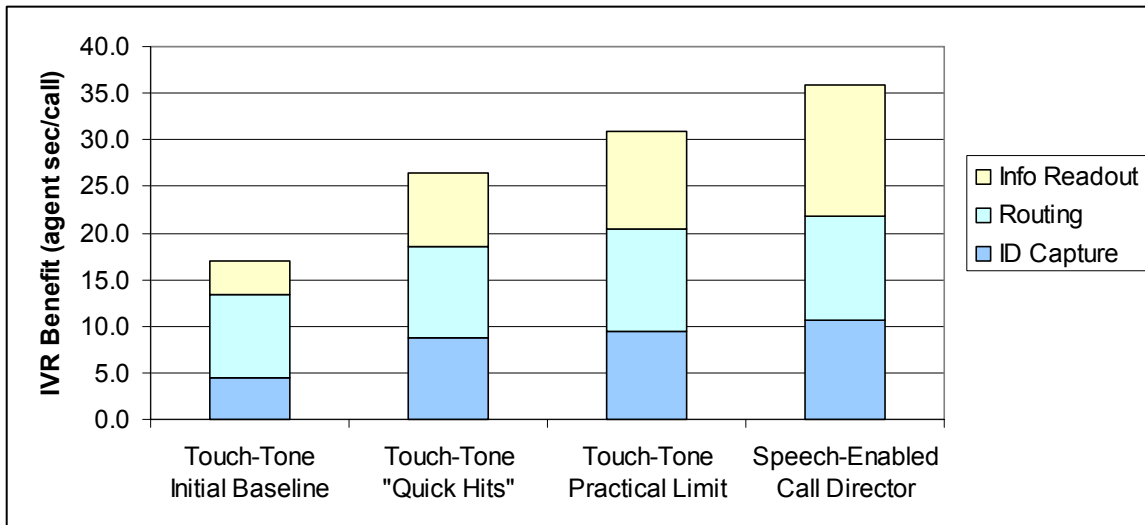


Figure 3: Comparative IVR Analysis

It can be seen that the quick hit design increased total IVR benefit from 17 to 26 agent seconds. By comparing the three automation categories, "ID capture", "info readout", and "routing to specialist", it can be seen that the increase is due to improved capture of account number and information delivery. Hence, this comparative IVR analysis validated that the quick design is indeed superior to the original system, and quantified the savings as 9 agent seconds. Touch-tone reengineering produced another 5 agent seconds of benefit, but is probably close to the limit of what can be done with a purely touch-tone interface. In contrast, reengineering with speech produced 10 agent-seconds of benefit beyond quick hits.

Benefit Projections

Due to the cost of IVR changes in large call centers, the redesign of telephone user interfaces must be justified by a business case. Our IVR automation analysis and benefit calculation can provide the necessary business justification for IVR redesign because the cost savings of a redesigned IVR can be estimated. Based on an automation analysis of the existing IVR and knowledge of usability problems, we can derive bounds for improvements in the various automation categories. From these bounds, we can project total IVR benefit to determine upper limits on annual cost savings, which are then used to justify reengineering effort

Our reengineering methodology, which is based on evaluating designs with real callers, eventually produces very tight benefit projections. In Figure 3, for example, the numbers for reengineered touch-tone and speech-enabled systems are based on such benefit projections.

SUMMARY

Telephone interfaces, an important class of human-computer interfaces, have been neglected by researchers in the field of human-computer interaction. Usability evaluation and engineering methods for telephone interfaces are not well developed. Decision-makers in call centers, under strong financial pressures, strive to cut costs without being able to assess the significant impact of usability on customer satisfaction and the financial bottomline. To remedy this situation, we have presented an assessment methodology for telephone user interfaces that evaluates both cost effectiveness and usability. Moving beyond previous laboratory studies of research spoken dialog systems, which evaluate only task completion rate and time, our methodology allows practitioners to evaluate usability of telephone interfaces in a systematic and comprehensive fashion. Furthermore, our methodology can be applied to production call centers that service millions of calls per year.

An evaluation of a telephone user interface must be based on thousands of end-to-end calls. Calls must be recorded in their entirety to capture the complete user experience, and thousands of calls are necessary to obtain statistical significance in the analyses. We have presented methods to analyze such large amounts of audio data efficiently. Our analysis transforms gigabytes of audio data into detailed event traces. For the IVR section, the event sequence is captured in a fully automated procedure, while manual transcription is necessary to annotate events in agent-caller dialogs.

We described a bag of assessment tools for telephone user interfaces: IVR automation analysis, user-path diagrams, and life-of-call diagrams. Additionally, we have introduced total IVR benefit to quantify the effectiveness of a telephone user interface in a single number. Beyond evaluation, these tools can also provide useful guidance during IVR redesign.

The methodology currently does not formally evaluate user satisfaction or any other subjective usability measure. While the impact of user satisfaction on customer attrition can be large, most managers of call centers focus on operational savings and ignore user satisfaction, because it is difficult to quantify. We believe that standard methods developed in the human factors community are sufficient to evaluate user satisfaction of telephone interfaces.

The dependence of the analysis of agent-caller dialogs on human annotators significantly impacts the cost for an assessment. In the future, we hope that audio mining technology will lower costs of transcription analysis and allow operators of telephone user interfaces to monitor their performance in a fully-automated fashion.

Our assessment methodology is applicable not only to commercial, but also to research telephone interfaces. As with any other interface, a solid evaluation methodology is a prerequisite for usability research. Research on novel dialog strategies for spoken dialog systems could benefit from the methodology presented in this paper, by providing researchers with a detailed analysis of how dialog strategies impact customer behavior and usability.

ACKNOWLEDGMENTS

The assessment methodology presented in this paper was developed over 2 years of research and consulting with several large call centers. The authors gratefully acknowledge the contribution of all members of the Call Director team at BBN Technologies, present and past.

Please Note: Some aspects of the tools and processes described in this paper are the subject of pending patents.

REFERENCES:

1. Balentine, B. and D.P. Morgan, *How to Build a Speech Recognition Applications*. 1999, San Ramon, CA: Enterprise Integration Group. 319.
2. Cohen, M., *Universal Command for Telephony-Based Spoken Language Systems*. SIG-CHI Bulletin, 2000. **32**(2): p. 25-30.
3. Stallard, D. *Talk'N'Travel: A Conversational System for Air Travel Planning*. in *Applied Natural Language Processing ANLP*. 2000. Seattle, WA.

4. Peckham, J. *A new generation of spoken language systems: recent results and lessons from the SUNDIAL project.* in *European Conference on Speech Communication and Technology EUROSPEECH*. 1993. Berlin (Germany): European Speech Communication Association.
5. Levin, E. and R. Pieraccini. *CHRONUS: The Next Generation.* in *ARPA Workshop on Spoken Language Technology*. 1995. Austin (TX): Morgan Kaufman.
6. Bennacef, S., et al. *Dialog in the RIALTEL telephone-based system.* in *International Conference on Spoken Language Systems ICSLP*. 1996. Philadelphia, PA.
7. Lee, C.H., et al., *On Natural Language Call Routing.* *Speech Communications*, 2000. **31**: p. 309-320.
8. Chang, H., A. Smith, and G. Vysotsky. *An automated performance evaluation system for speech recognizers used in the telephone network.* in *International Conference on World Prosperity Through Communications*. 1989.
9. Pallett, D.S., et al. *1994 Benchmark Tests for the ARPA Spoken Language Program.* in *ARPA Workshop on Spoken Language Technology*. 1994. Princeton (NJ): Morgan Kaufmann Publishers, Inc.
10. Edwards, K., et al. *Evaluating Commercial Speech Recognition and DTMF Technology for Automated Telephone Banking Services.* in *IEE Colloquium on Advances in Interactive Voice Technologies for Telecommunication Services*. 1997.
11. Walker, M.A., et al. *PARADISE: A Framework for evaluating spoken dialogue agents.* in *35th Annual Meeting of the Association of Computational Linguistics*. 1997. Madrid: Morgan Kaufmann.