Building efficient mobile systems and applications for supporting information exchange in resource limited settings

Siddhartha Asthana

January, 2017

Indrprastha Institute of Information Technology Delhi
New Delhi, India

Thesis Committee:
Dr. Pushpendra Singh, Chair & Advisor
Dr. Amit A Nanavati
Dr. Animesh Pathak
Dr. Anirudh Joshi

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: Mobile Systems, Voice Interface, Information Exchange, Human Computer Interaction

Dedicated to my parents and sisters

Acknowledgments

I am grateful to my advisor Dr. Pushpendra Singh for his continued guidance, support, and inspirational thoughts throughout Ph.D. program as well as in my early career. His words inspired me to achieve high standards for our research and encouraged me to be confident even in challenging circumstances. He has always been encouraging towards my research ideas and played a crucial role in converting them into concrete research directions. I would like to thank him for maintaining firm confidence in my capabilities.

I would also like to thank Prof. Amarjeet Singh for his insightful comments and suggestions in structuring my research. I learned a lot from my interactions with him. He always extended support and time to discuss my research work and provided detailed critique for the same. His honest and constructive feedback helped me improve my work at different fronts.

I would like to thank my monitoring committee members Dr. Ashish, Dr. Gaurav Gupta, and Dr. Vikram Singh for their time and reviews on my work that helped me correct my mistakes at the right time. I thank other faculty members and staff at IIIT-Delhi for their hard work in creating an excellent research environment and for their time to time help. I am grateful to Prof Pankaj Jalote who has been the backbone and source of positive energy in the Institute. His thoughts have been a source of inspiration, knowledge, and motivation for me in this journey.

I would also like to acknowledge all the hard work of all my collaborators and students with whom I worked. In particular, I want to heartily acknowledge Dr. Vivek Srinivasan (Stanford), Dr. Aditeshwar Seth (IIT-Delhi), Ms. Khadijah Faruqui (181-helpline), for their time and deep insights on our work. I would also like to thank Parul and Shradha for extending their support in the 181-helpline project. They have played an instrumental role in helping me with implementing and setting up 181-helpline. I would like to thank TCS Research for supporting my Ph.D. research.

I was fortunate to have professional siblings who could share and understand the hardships of Ph.D. as a rigorous process. I will like to thank Kuldeep, Garvita, Anush, Paridhi and other research scholars from MUC and Precog for being their encouraging words and support. I am also grateful to Niharika for her time and critiques on my work. From time to time, she helped me not only with technical insights but also helped me sustain through the hard times. I am extremely fortunate to have her as my supporter in this journey. I will always remember her interesting discussions, advice, and support for me.

Finally, I would also like to thank my supportive and affectionate family for its support. I am grateful to my parents and my sisters for encouraging and supporting me. I would like to express my utmost gratitude to my parents for teaching. Even in this lifetime, I may not be able to repay their sacrifice and love for me.

Abstract

Enabling support for information exchange has a great impact on the development of different sections of the society. Mobile phones using telecommunication services have enabled such support in several scenarios including healthcare, agriculture, and physical safety. In many scenarios, mobile phones are also the preferred technological medium like reaching out to police or hospital for assistance in emergency situations. In several geographies such as rural areas of developing regions, mobile phones are the only possible solution due to several human factors (illiteracy, poverty, etc.) and poor infrastructure support for other technological mediums. This motivated us to study different systems that support information exchange using mobile phones or telecommunication services.

In this thesis, we focus on building efficient mobile-based information sharing technologies for their benefits to different sections of society including developing regions. We have study three basic forms of the telephonic system that are currently used for information exchange and other telephonic systems can be realized by combining these three systems. First, we study helpline systems that are mostly used as an emergency responder and customer care systems. We developed a statistical method to evaluate and design temporal metaphors that can help to reduce the number of unanswered calls on a helpline system. We evaluated our solution using a series of real-world experiments conducted at Government Helpline. Second, we study automated voice applications that are widely used in ticketing services of airline and railways. Our proposed adaptive voice response system provides a way to reduce navigation time so that desired information can be accessed quickly on menu based voice application. We have also presented the real word evaluation of our adaptive system. Third, we study Smartphone applications that are also becoming popular in several areas including remote health monitoring and sensing. Our proposed method provides a solution for smartphone based deployments in regions where data connectivity is a substantial hurdle to such implementations. Towards this, we have build a system that can reduce the dependency on data connectivity by enabling multimodal communication in the telephonic system. We have tested our system in a rural area over a period of nine months. We have also developed and evaluated method to send data through voice when data channels are not available with the cellular network. A prototype for android platform is also developed along the same line to send SOS signal with GPS data from smartphone application when data connectivity is not available.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *"Mobile phones are misnamed. They should be called gateways to human knowledge."*
>
> - Ray Kurzweil, *Futurist (at Handheld Learning '09)*

Telecommunication services are a vital medium for information exchange and communication. They have a vast impact in developing countries where the reach of the Internet connection is relatively low as compared to developed countries. At present, there are nearly 7 billion (95.5% world population) mobile phone users [1] compared to 3 billion users of the Internet [72] in the entire world. According to 2016 statistics released by Telecom Regulatory Authority of India (TRAI), there are 1051 million land-line and mobile connections in India whereas the subscriber count of the Broadband Internet users is around 144 million only [98]. Similar trends can be seen in other BRIC[1] nations [2]. Brazil, Russia, and China have less than 42% of the total population connected to the Internet. The number of subscribers of cellular mobile in terms of total population for Brazil, Russia, and China is 90.5%, 162.4%, and 64.4% respectively. These figures clearly show the high penetration of telecommunication services in the developing countries as compared to the Internet. Thus in the developing regions, phone based medium has more reachability than the Internet. Mobile based system are also proposed as preferred technological solution in several contexts including agriculture [80], health care [93], education [62], etc.

Several research studies have shown that the people are already acquainted with mobile usage and hence they quickly adapt to the mobile-based intervention. Another advantage of mobile phone based services is their ability to deliver and generate content through voice at an incredibly low cost that helps to overcome the problem of illiteracy and poverty with such user groups.

In addition to remote and rural areas, mobile-based solutions also dominate several scenarios such as police, women and child welfare department, hospitals, and multinational companies in the urban communities. These organizations prefer to use call center like solution accessible through mobile phones as their first point of contact, especially during emergencies. Services where human intervention is not needed, callers can also access information through automated voice applications. With advancement in technologies, it has opened another mode of intervention through mobile phones i.e., smartphone applications. These smartphone applications can enable information exchange in several modes including voice, text, and graphics. These appli-

[1] BRIC refers to Brazil, Russia, India, and China.

cations make use of mobile phones as a powerful computing platform that has made technology portable and remotely accessible as a solution for many problems including health care and education.

1.1 Mobile phones for supporting Information Exchange

Mobile phones are established as the means to support information exchange in several scenarios by connecting people over far distant places. An efficient information exchange system using mobile phone requires building a well designed system such that the information seeker does not worry about whom and when to ask their query. The whole management of information exchange lies on the system and not on the information seeker. Several researchers have proposed such system that enables information exchange for a particular scenario or community. Patel et al. have proposed the use of voice menu based system for farmers where farmers can post their query related to the agricultural practices to get an expert opinion or can listen to best practices from the peer farmers [80]. In a similar scenario, Government of India has deployed several call centers for farmers that enable immediate query resolution in their local language [97]. Also, researchers have developed the systems for supporting information exchange for a healthcare scenario where they utilize capabilities of Smartphone [53].

Thus, based on existing usage of mobile phones, it can be said that mobile phone acts as the information sharing technology in three broad perspectives: a) Telephonic Helplines, b) Automated voice applications and c) Smartphone Applications. These technologies either serve as a cheap and affordable platform for communication and/or low-cost portable computing device. In this thesis, we address challenges of each system type to make information exchange more efficient.

1.1.1 Telephonic Helplines for information exchange

Telephonic helplines are the simplest yet the popular form of mobile system that is used to support information exchange. Government organization setup telephonic helpline to aid people to gather or share information about a topic such as Kisan Call center for agriculture, women helpline for counseling and legal support to distressed women, Railway helplines to get scheduling information, etc. Helpline in the form of call centers is also popular among commercial organizations to help their customers.

Helplines are an important mobile based ICT for developing region as knowledge sharing can be done in local languages and without the need of smartphone capability. Further, expert advice and information can be provided by Helpline agents to information seeker in a much more personalized manner than any form of technology. Figure 1.1 shows information seeker and provider end of Helpline system. In such system, often the limiting factor for information exchange is the amount of human resource available at Information provider end. The high cost of employing human resources for providing information becomes one of the reasons that information seeker has to wait for a long time in telephonic queues before their call gets answered. This results in a situation where many information seekers hang up their call to the helpline before any of the humans at information provider end answers their call. This is wastage of resource both in terms

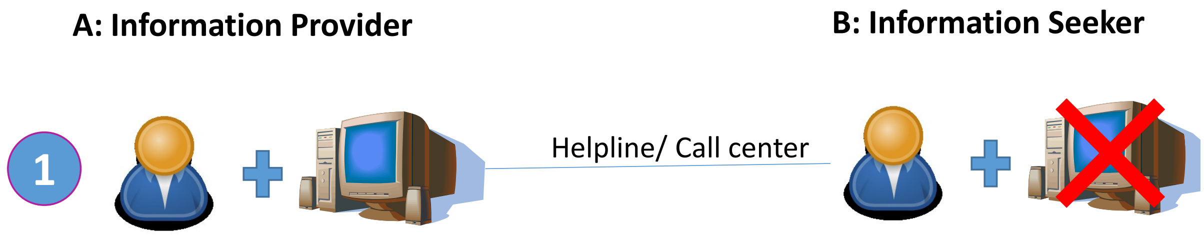


Figure 1.1: *Information provider and seeker in a Helpline System*

of time and cost, and sometime may lead to a risk of human life (distress caller helpline). In such scenarios, the objective is to maximize the reach of information exchange by reducing the number of callers hang up their call before reaching to a human agent at information provider end. Thus, we find that there is a need to focus on minimizing the number of calls getting unanswered without any additional cost to human resource.

1.1.2 Information exchange through Automated Voice applications

Information exchanged through automated voice applications is preferred in scenarios where information access can be automated and may not need human interventions. These voice applications help many services to overcome the problem of limited human resource, complementing the helpline systems that majorly rely upon human intervention. Figure 1.2 shows information seeker and provider end of such a system where automated systems use computing power to overcome the problem of the limited human resource at provider end using voice application. In

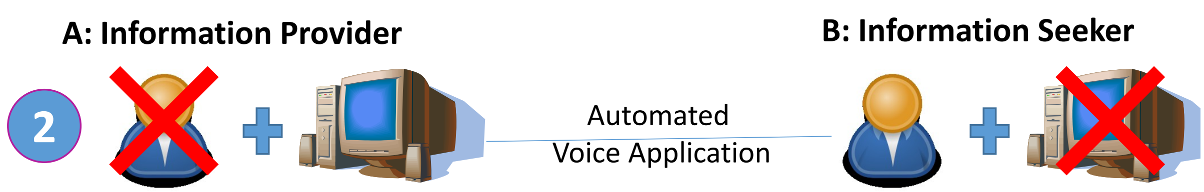


Figure 1.2: *Information provider and seeker in an Interactive Voice Response System*

automated voice applications, a user navigates through voice menu where different menu options and sub-options are arranged in a hierarchal fashion similar to a tree structure. Despite its advantage of removing the human from the loop, several studies have identified that high navigation time in accessing information makes use of voice application time consuming and frustrating. Navigation time also limits the amount of information that can be stored in the system as increasing information in the system requires longer menu structure to be traversed that leads to even higher navigation time. Thus, reduced navigation will not only improve the usability of the system but also increase the amount of information that can be stored in the system. Further, faster access to information will result in the faster release of telephonic resources occupied by each serving users.

Several studies have proposed different system designs to reduce navigation time. These techniques exert either a high cognitive load on the user or require the user to be familiar with the system. Thus, we believe there is a need of exploring system designs that are transparent to the user and thus reducing the additional cognitive load on the user.

1.1.3 Smartphone Applications for Information access

Mobile based technologies discussed above i.e. telephonic helplines and voice applications offer easy ways to exchange information through voice channels. However, these cannot support complex tasks where data transfer needed for advanced mobile solutions such as mobile banking and m-health. Having computational capability at both the end (information seeker and provider) of Information exchange (see Figure 1.3) can leverage the smartphone application to support such complex tasks such as m-health and mobile banking that are difficult to realize through voice only systems. However, the limiting factor in developing region for such mobile system is the availability of data connection. Research conducted in developing region has shown data connectivity issues with cellular services [57]. Searching for an alternate solution to send data,

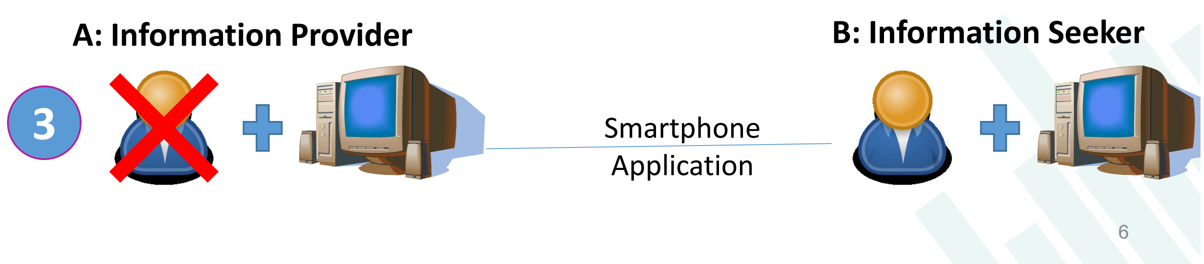


Figure 1.3: *Information provider and seeker interacting through Smartphone application*

research has proposed the use of reliable voice channel of cellular services for data transmission. We believe efficient and improved techniques need to be built for developing regions that can send data by piggybacking over the voice channel.

1.2 Contributions

In this research, we focus on building efficient mobile-based Information sharing technologies for their known benefits to the developing world. Specifically, this thesis has following contributions to mobile-based Information sharing technologies utilizing both voice and data services:

- Helplines /Call centers: We identified an open challenge in such system is to manage the waiting callers in the telephonic queue so that they do not hang-up the call without getting any human response. We designed and deployed an auditory progress bar (APB) that encourages callers to wait longer so that people responsible for answering these callers get more time to answer each call. This has led to drop in the number of calls being unanswered. We also propose an objective evaluation of APB performance using statistical measure for time and duration analysis. The proposed method makes APB evaluation free from user bias that arises due to subjective evaluation of such system in previous HCI studies. We also did a comparative study among different forms of APB design to better understand their capability in retaining the waiting caller longer in a telephonic queue.

- Automated Voice application: Menu based navigation in automated voice application are considered slow and time-consuming that results into caller frustration. This also forms the limitation to the system scalability as menu options grow while storing more information leads to higher navigation time and caller frustration. In this thesis, we propose a solution to navigation problem by developing an adaptive menu interface that reduces navigation time of a caller. Our approach is independent of user profiling or individual history of system usage. The real world evaluation of adaptive system shows statistically significant performance difference over baseline.
- Data Connectivity: To encourage the Mobile ICT solution that utilizes data services, we propose a solution to overcome connectivity issues. We have built a solution that can send data over voice channel when the data channel is unavailable. In telephony, voice services are preferred over data services, hence, utilizing voice channel for data transmission ensures better availability of data transferring mechanism. The proposed solution improves the existing method by utilizing reverse channel to achieve higher data rate and error sensitive adaptive data transmission. We have designed and deployed an Android based solution that can send the sensor data to a remote location using voice connectivity.

1.3 Outline of the Dissertation

This dissertation discusses all the three forms of Mobile ICT as mentioned above starting with helpline based information exchange which is the simplest form of mobile based information exchange. Then we discuss automated voice application that with the help of computing infrastructure has significant benefit over helpline system regarding human workforce required for mobile based information exchange. At the end, we discuss a more sophisticated version of mobile based information exchange using smartphones where the system can leverage the computing power available with users' phone. For each of these three types of system, we present observations from field deployment which has led us to identify the relevant problem in each system. Following the observation from field deployment, we present the relevant literature on identified problem. At the end, we present an evaluation of proposed solution through real world studies and control experiments.

The rest of this dissertation is structured as follows:

- Helpline: In Chapter 2, we present the observation and findings of our Helpline system deployed for the State Government of Delhi. Based on the findings and literature survey, we identified handling waiting callers at helpline with the limited human resource to attend them is still an open and challenging problem. Towards this, we also identified Auditory Progress bar- a temporal metaphor for Auditory Interface as potential solutions for the helpline. We discussed the shortcoming of existing methods to evaluate temporal metaphors and proposed first objective and quantitative method for performance assessment of temporal metaphors. The proposed methodology is based on Survival Analysis and helps in the judgment of statistical significance in performance difference of two or more metaphor designs. We also presented a thorough evaluation of different temporal metaphor designs for the helpline.

- Automated Voice Application: In Chapter 3, we presented observation and findings of menu based voice application deployed for IIIT-Delhi to provide admission related information to its callers (or admission applicants). Based on findings of field deployment and literature survey, we identified navigation in voice menu system is and open and challenging problem. We proposed and evaluated the performance of data-driven intelligent systems that adaptively rearranges the menu option to save on menu navigation time. We further investigate and compare different algorithm designs. This real world comparative study builds a good understanding of different attributes of data-driven mechanism to improve the performance of adaptive interfaces. In Chapter 4, we also present design and development of Maareech, a usability testing tool for voice response system.
- Smartphone Applications: In Chapter 5, we present observations and findings from the deployment of mobile-based ICT in rural areas. Findings of deployments that were consistent with other past studies done in rural areas of developing region show the lack of reliable data connectivity in this region. We propose a framework that can make use of voice channel in GSM to transmit data whenever/wherever data connectivity is not available on the mobile device.

Chapter 2

Waiting Caller: Information exchange through Helpline systems

Phones have emerged as a major medium to connect with commercial as well as non-profit organizations. Almost every organization, across the world, provides telephonic interfaces to fulfill the information needs of its customers e.g. providing information for technical assistance, lodging a complaint, or have a general enquiry. To satisfy these information needs, organizations may set up a call-center or helpline system accessible through any telephonic device including fixed-line phones. Research shows that telephonic system set up to satisfy information needs of users provide an added advantage over other digital technologies in terms of requiring minimal user training [80]. This advantage makes them easily adaptable for any section of society including rural and illiterate populations. For example, several NGOs and government organizations have set up their helpline systems to ease out information exchange for farmers in rural areas [80], women in distress [42, 77] etc.

According to recent statistics, the global market for such helpline and call center will reach USD 9.7 billion by 2019 and is growing at a rapid Compound Annual Growth Rate (CAGR) of over 9% [30]. The success of these systems is primarily governed by user/customer experiences with these helplines. There are several factors that may affect the user/customer experience, e.g., time spent by a caller while waiting for an agent to answer the call, successful resolution of caller's query, etc.

Traditionally, researchers have paid most attention to the time spent by callers while waiting for an agent to assist them since all the other factors are largely dependent on skill levels of the agent. In an ideal world, a caller will never be on hold waiting for an agent, however, in practice, this would require the number of agents to be always more than the possible number of simultaneous calls. For any organization, this is not economical and hence organizations always look for creating a balance between the number of agents required and risk of losing callers who are on hold. One can solve the problem of losing a caller in two ways:

1. By minimizing the waiting time of caller
2. By making waiting time of caller more pleasant so that they can be encouraged to wait longer.

The first approach requires statistically analyzing call center data to learn call arrival rate, abandon rate, etc. to estimate an optimum number of agents required at any time so that possible waiting time can be reduced to the desired service level. The second approach is to either improve the interaction with the system so that waiting becomes pleasant or encourage callers for waiting longer by inducing user behavior through additional information.

In this dissertation, we focus on the second approach where we try to explore methods and techniques for inducing longer waiting behavior in callers of call centers/IVR systems. Among the multifaceted aspects of usability concerning efficiency, effectiveness, engagement, error tolerant and easy to learn[1], we formulate it as a problem of improving user engagement on auditory displays. User engagement is a challenging problem as wrong attempts to engage user may result in more frustration for the user. In the context of call centers, user engagement is traditionally achieved through the use of IVR systems which can interact with a caller over the phone through touch-tone or voice recognition. The IVR system exhibits an auditory display in terms of voice menu and options to the users connected through phones. IVR systems are currently heavily used by commercial, non-commercial, social and government organizations. In call centers that experience high call volume and also suffer from the scarcity of human agents, IVR is used as a tool for engaging callers (user) to wait till any of the agents becomes available to attend the waiting caller. In general, IVR announces the delay in the form of a repeated audio message with some time gap between each repetition. The content of the audio message is mostly a variation of the following text:

"All our call executives are busy in talking to other customers. Please wait for sometime"

OR

"Your call is important to us. Please wait while we connect you to our call representative"

Apart from these, some IVR systems also estimate the waiting time for the caller in the queue by measuring congestion level in the call center and announce it:

*"Your estimated wait time is **X** minutes."*

Different audio messages can induce different behavior in waiting callers. Unlike a queue in the physical world, the telephonic queues remain invisible from the callers as the callers do not know their positions in the queue. This may result in anxiety and irritation at the callers' end since they can not see or judge the progress to make an informed decision about waiting or not. Hence, a good use of these audio messages should be in providing a sense of progress towards the end goal, i.e., end of waiting. These audio messages or other temporal metaphors can also affect the perceived waiting time of the caller which may be different than the actual waiting time. In this research, we focus on designing temporal metaphors for auditory displays as provided by helpline to induce longer waiting time among callers.

2.1 Background - Helpline, Waiting-Time and Temporal Metaphors

Managing waiting time of callers at helpline has attracted the attention of many research community including Operations Research (OR) and Human Computer Interaction (HCI). While OR

[1] http://www.wqusability.com/articles/more-than-ease-of-use.html

community focused on modeling call traffic so that correct estimates of required work force can be employed to keep the waiting time lower than desired service level, HCI community tries to manage caller through pleasant user experience so that callers are encouraged to wait more than normal so that they can be served with minimum work force. For this purpose HCI researchers has shown the usefulness of temporal metaphor that can help reduce perceived waiting time of a user. Thus, we review literature related to Helpline, waiting time and temporal metaphors.

2.1.1 Modeling call traffic of helpline

Operational research in the area of call centers and IVR systems with respect to above mentioned problem is largely focused on building a statistical model which can estimate the number of trunk lines, number of agents required to meet a certain service level etc. Srinivasan et al. [95] built a model based on Poisson process where call arrival rate was a constant λ. If all trunk lines were busy then the call was either lost or otherwise it would spend some time in IVR and would be answered by an agent with probability $\boldsymbol{p}$ or abandoned with probability $\boldsymbol{1\text{-}p}$. Several improvements for this model have been proposed including customer impatience in the model in terms of call abandon rate of the queue [52, 54, 68, 100]. The customer impatience has also been studied to modify queuing system of call centers. Jouini et al. have compared different queuing systems for multiple type of impatient customers [50]. However, we aim at proposing techniques and methods for effective user engagement on audio only medium rather than proposing queuing system or building statistical model for estimating resources for call centers.

2.1.2 Temporal Metaphors: Managing callers perceived wait time through feedback

Waiting Time in HCI is an important aspect of system design to enhance User eXperience (UX) when the interaction between the system and its user is temporarily interrupted (e.g., loading of web page, file download, or waiting for the arrival of live agents at call centers) [61]. Literature suggests that providing feedback to a user on the system state, especially during these waiting periods, increases system usability [89]. Depending on the type of system, several temporal metaphors (visual progress bar [44], auditory progress bar [58], non-speech waiting cues [43], etc.) have been designed to provide such feedback to the user. The objective of introducing a temporal metaphor in the system is to mitigate the adverse effect of wait time so that the system can successfully hold the user until the waiting process completes.

Peres et al. have suggested an interesting approach for engaging the user with audio progress bar which is analogous to visual progress bar on GUI [81]. The audio progress bar tries to reduce the perceived waiting time, thus making callers wait for longer time. Engaging users on an IVR system is usually achieved through time fillers like background music, apologies message (i.e., Sorry, all our agents are busy. Please hold the line), delay announcements (Your expected waiting time is 2 Min) etc. Many research works focus on caller behavior as affected by audio message related to delay announcement in the call centers. Feigin conducted an experiment that announces an expected waiting time (<1 min, 1 min, 2 min etc.) at high level of congestion in IVR and recommends the callers to return to IVR [37]. It was shown that approximately 1.3% of the total callers returned to IVR. This study suggests that the audio message can change the waiting behavior of caller but we require more research on content and presentation of audio messages to have impact on larger number of callers. Armony et al. have studied the equilibrium associated with the delay announcement [10]. The equilibrium exists in a sense of cyclic dependency of the 3 factors i.e., caller's response to the delay announcement, the system performance depending upon caller's response, and the announcement depending upon system performance. Several other researches have shown keen interest in developing ideas around delay announcement. Allon et al. have studied the impact of postponing the delay announcement in comparison to providing the delay announcement immediately [8]. They have shown that this postponement can actually help the call center maintain more credibility among callers and augment the resulting equilibrium in certain settings. In all the studies, the delay announcement from call center is assumed to be of high credibility at caller's side. However, research has also explored if call centers can reap benefits by giving false information about delay to lure the customers [9]. Rafeli et al. have explored psychological aspects of different time fillers like delay announcement, music, and apology message and their effects on the cognitive process of a waiting caller [74]. They conducted a lab experiment and compared the delay announcements with two other time fillers like music and apology message. Niida et al. have also showed the effectiveness of different time fillers on visual display [78]. From the literature survey, we conclude that the delay announcement is an effective mechanism for engaging user on auditory displays.

2.1.3 Temporal Metaphors and evaluation methodology

Traditionally, researchers have studied the effectiveness of temporal metaphors through several subjective measures including user preference [29, 58], time perception [43, 44], satisfaction [47], acceptability [29], and appropriateness [38]. At the same time, the use of objective measures has not been explored much to evaluate the effectiveness of such systems. We discuss the observation that existing subjective measures are susceptible to several evaluation conditions (like cognitive load, the paradigm used for estimation [24], etc.) that can bring variations in the outcomes of the experiments. Further, there are several important objective outcomes that cannot be inferred from subjective measures like *"for what time duration can the proposed temporal metaphor hold users, on average"*. Thus, we emphasize the need for objective measures to generate objective outcomes in the study.

An experiment on the evaluation of temporal metaphor involves a users' reporting about the time perceived by them (also known as subjective time). Subjective time is a complex cognitive process, and user assessment depends on several factors. Lallemand et al. have discussed the contribution of cognitive psychology to better understand subjective time [61]. Here, we discuss how these different cognitive processes and assumptions can bring variations in results and their interpretations for a waiting time experiment.

Mode of Experiment

There are two paradigms to conduct a waiting time experiment for subjective assessment, viz., prospective and retrospective paradigms. In a prospective paradigm, users are informed at the beginning of the experiment that they have to estimate the duration of a given time interval. In the retrospective approach, users are not aware of the time estimation task until the end of the experiment. Studies have shown that retrospective and prospective paradigms have opposite effects on the duration judgment [21, 25]. Block et al. have tried to investigate the reason for performance differences due to different paradigms used for subjective assessment and concluded that in prospective paradigm, a greater cognitive load on user decreases the subjective-to-objective time duration ratio whereas in retrospective paradigm a greater cognitive load increases the subjective-to-objective time duration ratio [24]. This performance difference due to the choice of paradigm used for conducting an experiment and the lack of standard methods to translate results from one paradigm to another makes it difficult to compare studies conducted in the past.

Cognitive Workload on the User

The subjective assessment of the time depends a lot on the attention given by a user on the passage of time. Thus, time is perceived as passing slowly when attention is focused primarily on time [23]. The amount of attention on temporal information is less if attention on processing non-temporal information is more, due to a high cognitive demand of a specific task [20, 104]. Thus, the subjective time judgment depends a lot on cognitive workload on the user. Further, users differ from each other in terms of cognitive capabilities, which means that for the same task, different users may perceive different cognitive loads. Thus, for the same experiment, different users report different perceived (subjective) times. Thus comparing results for performance difference of temporal metaphors based on different studies in literature is difficult, as it is not certain whether the difference in results is achieved due to a different system or different cognitive capabilities of their respective user set. One way to avoid such problems is by having a properly sampled large number of users in each experiment, but that is costly and not always feasible. Another approach may be to report cognitive load on the users during the experiment along with other experimental results. There are different psychometric scales to measure cognitive load on the user, such as the Subjective Mental Effort Questionnaire (SMEQ), also referred to as the Rating Scale for Mental Effort (RSME) [106], and NASA-TLX [45]. However, little is known about the representativeness of these scales in terms of the actual cognitive load on the users and the distribution of observed load on the user again in terms of how much is due to the system and how much is due to the difference in cognitive capabilities of a user.

Retention Delay

Retention delay is the duration of time beyond which information cannot be retained by the short-term memory of the human brain. A waiting time experiment conducted with or without retention delay may produce different results. As the human brain does not remember past events in a consistent and linear manner [7, 28], event recall happens with selectivity and bias, where it is easier to remember the first or last moments of an event than those happening in between. Initial moments are stored in long-term memory, and the last or recent moments are stored in short-term memory. Short-term memory is less stable and can be affected by a delay of more than 15-30 seconds (or retention delay); it can also be affected by an interfering activity [39]. Results for a retrospective evaluation after retention delay will solely be based on the long-term memory [61]. However, if the evaluation is done without retention delay both long-term and short-term memory affect the subjective assessment.

Psychological Models for Subjective Time

Most of the literature on waiting time is focused on explaining the distortion in subjective time perception (under or overestimation). Several models have been proposed to study the perceived time based on the existence of the internal clock [6, 32, 103].The attentional gate model proposed by Zakay et al. [105] is considered the best to explain subjective time in HCI [61]. This model consists of a pacemaker (internal clock) and an accumulator connected by a switch as described in the model proposed by Triesman [99]. Zakay's model assumes an additional gate (in comparison to Triesman's model) between the pacemaker and switch. The subjective time depends upon the way a person divides his/her attention between information encoded by a temporal information processor and non-temporal information processor. The gate takes temporal information into account and co-ordinates activation of the switch. Thus, different researchers may make use of different models to explain their results and develop different theories. For a comprehensive literature review of the models, readers are referred to [23, 41, 65].

2.2 Women Helpline: A Motivating Scenario

In 2013, we began working with a government organization that manages a Helpline system to provide information to women in distress. We collected data on the usage of Helpline system to better understand the type of callers and the context of the calls. This helped us in designing further experiments for studying the effect of temporal metaphors on caller waiting behaviour.

2.2.1 181-Women in Distress Helpline

The state of Delhi is the national capital of India. It has seen a steep increase in the crimes against women in recent years. A major problem that women have registered is the lack of quick access to police or other law enforcement agencies in case of violence against them. To address women's concerns, the state of Delhi has established helplines such as 1091 (Delhi Police Women Helpline) and 23379181 (Delhi Commission for Women). Despite these helplines, however, women continue to find it difficult to gain quick access to information regarding police or other law enforcement agencies. To handle these issues, the state government of Delhi decided to start its supplementary helpline, named the *181 - Women in distress* helpline, to provide assistance to women in distress. The helpline started on 31st December 2012 with 24 x 7 availability and a toll-free number. On its first day, the helpline received around 15,000 calls, however, since then the usage has reduced significantly. The call center is managed by women call executives who are trained in handling women's issues and some legal advisors. The calls vary from seeking counseling over petty matters (e.g., verbal arguments within the family) to criminal assaults (dowry death, rape, etc.) as said to us by agents attending to the calls. The helpline provides advice and consultation to women and coordinates with other state agencies (police, legal aid, medical service, etc.) whenever needed for the quick resolution of women's complaints. Overall, since its launch, the helpline has become very popular among the Delhi women due to its effectiveness in getting issues resolved.

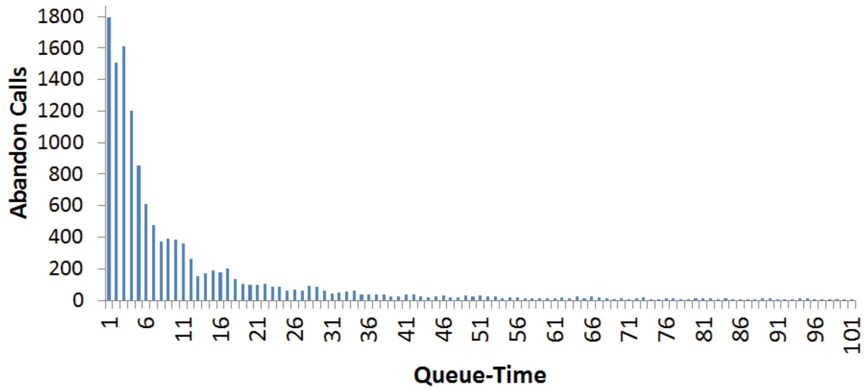


Figure 2.1: *Distribution of Queue-time (in seconds) of abandoned call in pre-experiment data collection*

2.2.2 Helpline inspection

To understand the waiting behavior of callers, we analysed the helpline prior to experiment design. We analyzed 29,548 calls that the helpline had received in the past three months before the experiment. Out of the total calls, 13,428 were abandoned calls, i.e., calls that were not answered by any agent, and 16,120 were answered calls. We also observed that most of the callers try again, so the actual number of unanswered callers is not as high as it looks prima facie. To understand the system dynamics, in order to design our experiment better, we performed some basic analysis. We analyzed waiting time (or queue-Time) for the abandoned calls and the answered calls separately. Figure 2.1 shows the distribution of abandoned calls based on queue time. The Y-axes shows the number of abandon calls and the X-axes shows their respective queue time (or wait time). For abandoned calls, we found that 95% of the calls had a queue time (waiting time) of less than 97 seconds. It shows that 95% callers on such helplines do not even wait for100 seconds, which reflects the impatience of callers due to stress. On the other hand, the waiting time in a regular call centers could be as high as 40 to 50 minutes (as reported in a leading newspaper of the UK [2]). This shows that the nature of helpline callers is different from a normal call center.

We want to reduce the number of abandoned calls as these calls are important for the purpose of the helpline. Though many callers return, it will still be better to serve them at the first instant. One solution is to increase the number of call executives at the time of a rush hour, but due to several reasons, the number of executives cannot be increased. Thus, we would like callers to wait more so that they can talk with a helpline executive and get their problems resolved.

2.2.3 Problems Identified

Based on above literature survey and analysis of data collected on helpline usage, we identify following research challenge to make Helpline system more effective.

[2]http://www.mirror.co.uk/news/uk-news/big-six-hold-times-energy-3168050

- Alternate evaluation method for Auditory Progress Bar to minimize bias arises due to subjective assessment.
- Effective Auditory Progress Bar (APB) Design to encourage helpline callers to wait longer than normal.

2.3 Survival Analysis for quantitative evaluation of caller waiting behaviour

As discussed previously, we identified how evaluation based on subjective measures can be influenced by contextual variables like cognitive load, retention delay, paradigm used, etc. The same experiment, when conducted by different researchers, may produce different results based upon different configurations that they choose for evaluation. We do not argue which one is better, but we emphasize that even when researchers explicitly mention the evaluation methodology they have chosen, it is not easy to compare the results with other experiments conducted with different evaluation methodology. As an implication of this, the literature may seem to have inconsistent findings and conflicting theories. A similar observation was made by Kortum et al. while designing an auditory progress bar (APB) for a telephonic system [58]. They mentioned that ten different studies conducted for an auditory progress bar had mixed performances (over and underestimation of time) and listener preferences. We believe that subjective analysis is a good method for understanding the attitude and preference of a user, but researchers also need to incorporate an objective analysis method for a variety of reasons. First, objective methods are less influenced by individual users, i.e., they can capture the actual behavior of a user with the system rather than user perception about the system. Second, it is less time-consuming to collect and analyze objective data for a large set of users. Third, there may be some system aspects that users cannot assess or report–for instance, *how long a system is able to hold users to wait*–and various probabilistic and statistical insights, like *the probability that a user will wait for y seconds*, cannot be assessed by users. Hence, an objective evaluation of a system needs to be integrated with the traditional subjective usability assessment methods. In this section, we address this gap by introducing an objective evaluation technique that can generate such insights.

We propose the use of actual wait time rather than perceived wait time as an evaluation metric for objective assessment. Subjective studies were largely focused around the perceived wait time, and the use of actual wait time is limited to calculate the mis-estimation (over or underestimation) in perceived wait time [58]. In subjective studies, low perceived time is desirable which is inferred as the system can hold a user for longer. But by merely inspecting the perceived wait time, it is difficult to guess how long a user will actually wait because perceived wait time is dependent on factors other than actual wait time like the cognitive load and paradigm used (retrospective or prospective) for time estimation [24]. Hence, in our proposed objective assessment we tend to measure the actual wait time of the caller directly instead of estimating it from perceived wait time or other perceptual variables. Based on the requirement, we propose the use of survival analysis (a branch of statistics) [55] as an evaluation technique for objective assessment of the wait time in HCI. It is a statistical tool for time-based analysis and has the capability to deal with censored data. With survival analysis, researchers can quantify the effect of temporal metaphors, judge its statistical significance, and generate probabilistic estimates of holding a user with the passage of time.

2.3.1 Conceptual Foundation

Here, we describe the important terms and concepts which will showcase the appropriateness and applicability of survival analysis. For a detailed description of survival analysis, readers are referred to [55].

Time Based Events and Censored Data

A time-based event is defined as an event whose time of occurrence is the interest of study. The case in which the time of an event is known below or above an observed value but the exact time of the event is unknown is called **censored observation** [55]. For instance, in the case of temporal metaphors, the outcome of interest is to find out the tolerable wait time before a user quits or cancels an ongoing process or transaction. Here, the event to observe is the canceling or aborting of an ongoing process by the user. Different users will quit or cancel the process after waiting for a certain time, which may differ among users. The wait time of different users can be used to statistically compute the tolerable wait time for the users with respect to a temporal metaphor. However, for certain users researcher will not observe any event if they wait till the completion of the process. In this scenario, researcher know only that the user waits till a certain time but do not know exactly how long she is willing to wait. In statistical analysis, such data points are termed as the **Right censored data** as the tolerable waiting time for a user is more than the observed time by an unknown value.

There are two challenges in analyzing data pertaining to a time-based event. The first challenge is to deal with censored observations and accommodate them with normal observations in the analysis. The second challenge is that researcher cannot expect data to be distributed normally as with increasing wait time lesser number of users will be willing to wait.

Survival Function and Probability Estimation

Survival function is the function that relates the probability of an event with the time estimated from time-based data discussed above. Conventionally, a Survival function F(x) is defined as,

$$F(x) = Pr(X > x) \tag{2.1}$$

where, X is a random variable denoting the time of event occurrence and Pr(X>x) is the probability that the event will occur later than the specified time x. For estimating such survival function, we will be using Kaplan-Meir [55], a non-parametric maximum likelihood estimator, which is used in several fields of science, like medical science for estimating the life of a patient, in reliability engineering to measure the time of part failure, etc. [55]. To estimate the probability for plotting the survival function, the following procedure is followed:

- Arrange the time-based data in ascending order and denote the time value as $x_0, x_1, ..., x_n$.

- At every event, calculate the probability using following recurrence relation:

$$F(x_i) = \frac{r_i - n_i}{r_i} F(x_{i-1}) \tag{2.2}$$

where r_i is the total number of events (including censored events) having time value $\geq x_i$, and n_i is the number of events (non-censored) $> x_{i-1}$ but smaller than x_i.

Size of Effect

To quantify the size of the effect of temporal metaphors on wait time behavior of users, researcher can calculate mean and median survival time from survival analysis (see descriptive measures of survival experience [55]).These terms are defined as:

- *Mean survival time:* It is calculated as the area under the survival curve [55]. It is one of the measurements to estimate the central tendency of the data distribution of wait time and represents how long a user waits on the system on an average.
- *Median survival time:* It is calculated as the smallest survival time for which the probability of the survival function is less than or equal to 0.5 [55]. Similar to mean, it is another factor that measures the central tendency of data distribution but is less susceptible to extreme values.

Statistical Significance

Logrank [19], a non-parametric test, is widely used in survival analysis to estimate whether the effect has statistical significance or not. It is a form of the Chi-square test and the hypothesis testing tool to compare two survival distributions. It calculates a test statistic for a null hypothesis (*survival curves for different groups are the same*) based on the following relation:

$$Statistic(Logrank) = \frac{\sum(Expected - Observed)^2}{\sigma^2(Expected - Observed)} \tag{2.3}$$

where $\sum(Expected - Observed)^2$ is the summation of the square of the difference between the expected number of events and observed number of events for each time point, and $\sigma^2(Expected - Observed)$ is the variance of difference between expected and observed events.

2.3.2 Field Experiment: Objective Assessment of Auditory Progress Bars

We designed one APB that gives a periodic delay announcement and compared it with the base line system (BL).

1. *Baseline (BL):* This system is the existing system of the 181 helpline, which does not provide any delay announcement to the caller. It has a pre-recorded audio message that repeatedly plays a welcome message to the caller in the Hindi language. The message translates to: *"Welcome to the Chief Minister's Women Helpline!!"*
2. *APB:* This is our proposed system, which announces an expected initial waiting time of 300 seconds to each caller. System keeps updating waiting time after every 25 seconds.

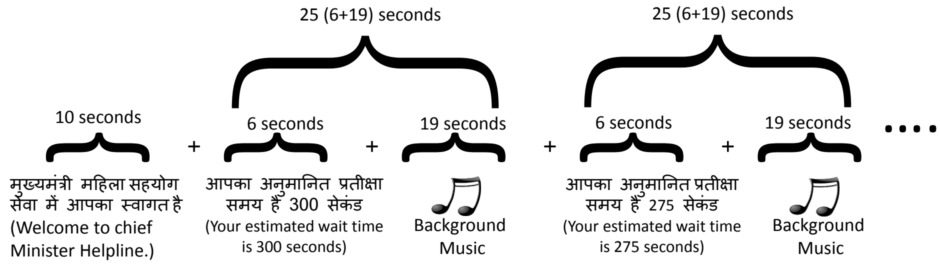


Figure 2.2: *Functioning of APB: First callers are greeted with a 10 seconds welcome message followed by first message about an estimated delay of 300 seconds. Each message about estimated delay takes 6 seconds. Thus, a caller with Queue-time greater than 16 seconds receives at least one delay announcement. Every estimated delay message is followed by background music of 19 seconds. The repeated announcement of delay messages by subsequently deleting 25 seconds from the previous estimate continues till the estimate reaches zero. After that caller only hears the background music.*

The value of each announcement against actual wait of caller is shown in Figure 2.2. It is a simple system which deducts 25 seconds from the last announced time and plays it in subsequent announcements as the time left for the caller. After the 12 ($300/25 = 12$) announcements, actual wait time becomes zero and hence system stops announcing any delay. The time for duration (300 seconds) and interval (25 seconds) of APB design is based on following considerations.

- *Duration:* Literature suggests to estimate the wait time for an announcement based on coverage probability (β) so that actual wait time never exceeds the announced time with probability β [51]. To ensure that the actual wait never crosses the announced wait, we would have to announce wait-time corresponding to $\beta = 1$. However, due to the exponential distribution of wait-time, the value corresponding to $\beta = 1$ will lie at infinity. We did analysis of 29,548 calls that are described in previous subsection (*Helpline inspection*) to estimate the wait time for announcement so that value of β is close to 1. The minimum value which covers sufficiently large number of callers in our system is 300 seconds ($\beta = 0.999$).

- *Interval:* Choice of interval is influenced by two aspects covered in [29] and [38]. Literature suggests constant attention towards the passage of time makes the wait time perceived as longer [29]. Hence, the value of intervals between the announcements should be high as every announcement draws the caller's attention towards the passage of time. On the other side, Fröhlitch et al. indicates that the wait time greater than 30 seconds risks a potential hang-up [38]. Nah et al. quoted another study that also indicates a 30 second threshold in another context [75]. Thus, we chose a high value of interval [29] that was less than 30 seconds [38, 75] and perfectly divided the duration (300s).

We designed to route every alternate call to the same system. For each new call, the system generates a numerical call ID in an incremental fashion. We took each new call as a different call even if it came from some previous number logged in our database and assigned it to one of the systems alternatively. To preserve the privacy of the caller, the system anonymized the caller-ID (telephone number) to a different numerical ID before storing it in the database. Thus, repeated calls from the same number could be identified but reverse mapping to the phone number was not possible.

2.3.3 Data Collection

For the analysis, we collected data for 1 month (6^{th} August - 5^{th} September 2014). The system logged the following call attributes:

- *Caller ID:* For each call connected to our system, we logged the anonymized caller id of the telephone number.
- *Time stamps:* For each call, we logged the time stamps of the call's start and end.
- *System allocation:* For each call, we logged the system name that was allocated, i.e., BL or APB.
- *Announcement Details:* We logged the number of announcement messages listened to during each call and the corresponding value announced in each announcement along with the time stamp of each announcement.
- *Queue Time:* We explicitly logged the time spent by each caller on the system waiting for an agent to answer her calls.
- *Call Status:* For each call, we logged whether it was answered by an agent or abandoned by the caller.

In one month, we received a total of 8,748 calls. There was a large number of calls that got answered or disconnected with waiting time (or Queue Time) $<=$ 16 seconds and hence did not receive any delay announcement (calls with Queue Time $>=$ 16 seconds received at least one delay announcement, see Figure 2.2). As we want to compare the performance of the system when callers got a delay announcement to its performance when callers did not get any announcement, we removed calls with Queue Time $<=$ 16 seconds. For consistency in analysis and comparability between the systems, we removed such calls from both the systems and were left with 1,353 calls.

2.3.4 Results and Analysis

Here, we present the analysis of 1,353 calls. Table 2.1 represents the distribution of calls between the two systems and the number of calls answered and abandoned on each system. These 1,353 calls represent the total calls from 1,082 callers, where some callers called more than once. The subsequent calls from the same caller may have a priming effect on their waiting behavior, based on their experience of the first call. To avoid any priming effect on the calls analysis, we have also presented the analysis for the data where only first call of each caller is considered. The distribution of the 1,082 remaining calls (i.e., considering only first call of each caller) is shown in Table 2.2. In the rest of the section, we will present the analysis for total calls and first callers separately.

Table 2.1: *Total number of abandon and answered call statistics for baseline and APB*

System	**Calls**		
	Answered	**Abandon**	**Total**
Baseline	255	471	726
APB	241	386	627
Total	496	857	1353

Table 2.2: *First Call: Total number of abandon and answered call statistics for baseline and APB while considering only the first call from each caller-ID.*

System	**Calls**		
	Answered	**Abandon**	**Total**
Baseline	208	383	591
APB	207	284	491
Total	415	667	1082

Abandon Rate

A caller who waits longer has more of a chance of her call getting answered than a caller who waits for a shorter duration of time. To observe which system (APB or Baseline) can hold a caller longer, Abandon Rate is defined as the percentage of total calls that were unanswered. Hence, a better system will observe an abandon rate lower than the other system. Based on the analysis of total calls (see Table 2.1), the abandon rate of APB (61.56% i.e., $386/627$) is less than the Baseline (64.87% i.e. $471/726$) by approximately 3%, but results were not statistically significant ($\chi^2(1, N = 1353) = 1.59$, $p = 0.2$). On analyzing the first call only, we found the abandon rate of APB (57.8% i.e., $284/491$) was much lower than the Baseline (64.8% i.e. $383/591$) ,and the results were statistically significant ($\chi^2(1, N = 1082) = 5.5$, $p = 0.019$). This indicates that APB performed better than Baseline in holding the callers. However, a caller waiting time is just one of the factors, and there are other factors, such as congestion in the line and number of agents which also affect the abandon rate of the system. Thus, we will further analyze the system for more insights.

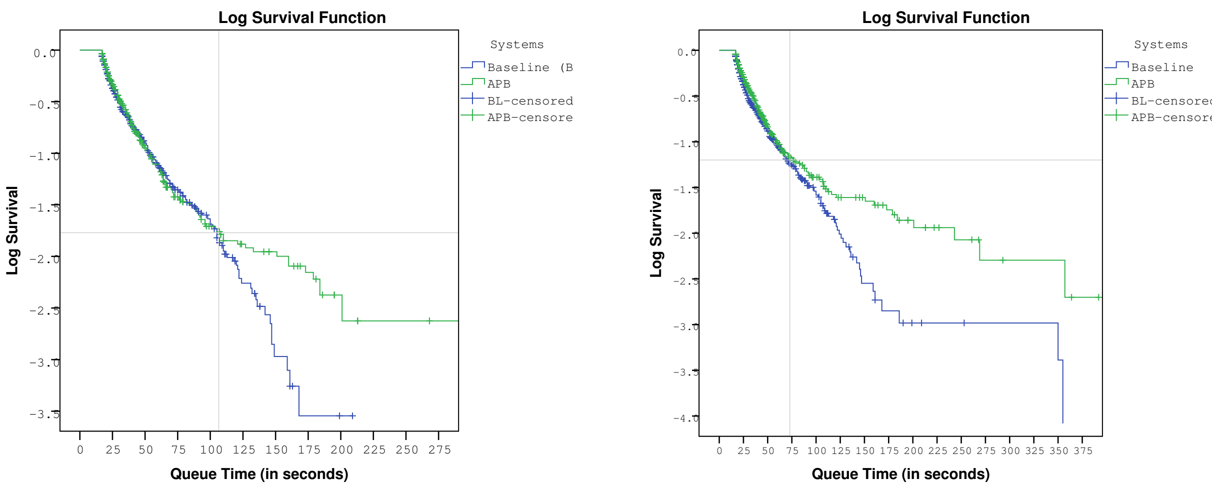
Survival Analysis

In this section, we demonstrate the use of survival analysis techniques as applied to wait time (Queue-Time) data collected in the experiment.

Probability Estimation

We did a survival analysis (Kaplan-Meier) to analyze the probability of callers waiting X seconds or more in the queue across different systems. In this analysis, the *queue time* is the *survival time* of the caller, *call abandonment* is the *observed event*, and *answered calls* are treated as *right censored data*[3]. We did a survival analysis of total calls and calls that were left after removing repeated calls from the same telephone number.

Figure 2.3a shows the estimated probability of callers waiting for a particular time (in seconds) on two different systems using total calls. The performance difference can be seen between the two systems for Queue Time > 100 seconds where the probability that callers will wait on APB is more than the Baseline. The same performance difference can be seen much earlier (around Queue Time = 60 seconds) when only first calls are considered (see Figure 2.3b).



(a) Total Calls (b) First Calls

Figure 2.3: *Survival Curve: A comparison of survivability of callers on Baseline, and APB. Markers on each curve show the corresponding censored (Answered calls) data.*

Size of Effect

We used the mean[4] and median[5] of probabilistic data generated through survival function to quantify the effect of different systems on caller waiting time. Table 2.3 shows the mean and median estimates of survival time for each system using total calls. The mean of APB (76.52 seconds) is approximately 13 seconds more than the Baseline (63.13 seconds). This shows that on an average, APB can hold each caller 13 seconds more than the Baseline. We observed that both the systems have the same median of 39 seconds that is evident as the left half of the

[3]In this experiment, the queue time of answered calls represents right censored data as caller could have waited more than the queue time but unknown by how much.

[4]Mean survival time is estimated as the area under the survival curve

[5]The median survival time is calculated as the smallest survival time for which the survivor function is less than or equal to 0.5

Table 2.3: *Means and Medians for Survival Time based on total calls*

	Mean[a]			
	Estimate	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Baseline	63.318	4.199	55.087	71.548
APB	76.522	5.910	64.938	88.106
Overall	69.737	3.623	62.637	76.837

	Median			
	Estimate	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Baseline	39.000	2.187	34.714	43.286
APB	39.000	1.595	35.874	42.126
Overall	39.000	1.379	36.296	41.704

Table 2.4: *Means and Medians for Survival Time based on first calls*

	Mean[a]			
	Estimate	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Baseline	74.100	7.364	59.667	88.533
APB	94.063	7.876	78.627	109.499
Overall	90.125	7.783	74.870	105.379

	Median[a]			
	Estimate	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Baseline	40.000	2.712	34.684	45.316
APB	43.000	2.582	37.938	48.062
Overall	42.000	1.661	38.744	45.256

a. Estimation is limited to the largest survival time if it is censored.

survivability curve is almost the same (see Figure 2.3). This can also be interpreted as if the helpline has sufficient agents where the waiting time of callers is below 39 seconds, then we will not observe any performance difference between the two systems.

We did the same analysis for *first calls*, and estimated mean and median are shown in Table 2.4. We can see a wider gap in the performance. The mean of APB (94.06 seconds) is approximately 20 seconds more than the Baseline (74.1 seconds). Similar to the analysis on total calls, median values are close to each other.

A performance difference of 13 seconds or 20 seconds (considering only first calls) may not impact full performance difference for a normal call center, but it is a considerable improvement for helpline callers where the average waiting time is less than 64 seconds (see mean of BL in Table 2.3) or 75 seconds (considering first call, see mean of BL in Table 2.4). It is also evident from the abandon rate, which was reduced to 3-7% (see previous subsection). Further, optimization on the APB design may bring better performance.

Statistical Significance

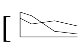
We did further analysis to check the statistical significance of our observed performance difference through Chi-Square statistics, i.e., Logrank (Mantel-Cox). While testing total calls for statistical significance, we did not observe any difference (χ^2 (1, 1353)=0.945, p=0.33). However, on analyzing the data of first calls, we find the results are statistically significant (χ^2 (1, 1082)=5.058, $\mathbf{p<0.05}$).

We further analyzed the reason for not achieving statistical significance with total calls. We believe one of the reasons could be the repeated calls in the data set. We hypothesize that a caller can estimate the waiting time in subsequent calls based on one's experience of the first call. Thus, irrespective of the system, callers have sufficient information on the waiting time. To check this, we analyzed 271 repeated calls (i.e. first calls subtracted from total calls). We did not find any statistically significant differences (χ^2 (1, 271)=0.003, p>0.05) in the two systems. The repeated calls on APB had a lower mean (83.38 seconds, 95% CI[52.25, 114.51]) and median (40 seconds, 95% CI[34.47, 45.53]) than the first calls.

2.3.5 Discussion

This section reflects on our results and discusses possible interpretations and limitations of it. We discuss how our proposed method can be generalized to other tasks, application types, and settings.

We showed that the user wait time data can be used objectively to evaluate the performance of temporal metaphors for their ability to hold callers. We showed survival analysis, a lifetime estimation technique, can be used for such purposes to generate probabilistic and statistical insights.

The first concept that we discussed is a survival curve. It is a continuous time-based evaluation of temporal metaphors. It denotes the probability a user will wait for a given time. Several insights can be drawn from this while comparing temporal metaphors. For instance, if curves are non-overlapping [] then the temporal metaphor with a curve on a higher side along ordinate (Y-axis) represents better performance. In the case of overlapping curves [], let's say intersecting at time **t** on the X-axis can be used to make a thoughtful decision on which temporal metaphor should be used based on the average wait time of task for which that temporal metaphor is employed. If average wait time of task is less than **t**, then the metaphor whose survival curve is at the higher side for wait time less than **t** should be used. Otherwise, if average wait time of task is greater than **t**, then the metaphor whose survival curve is at the higher side for wait time greater than **t** should be used. An important underlying assumption for statistical evaluation of the survival curve that overlapping curves violate is proportional hazard. If survival curves cross each other, then this is the evidence that hazards are not proportional. For such overlapping curves, the analysis should take into account the interaction effect and separate the analysis before and after the crossing. We refer the reader to [55] for hazard function and assumption of proportional hazards.

Second, we discussed quantifying the size of the effect of temporal metaphors on the wait time behavior of users. If a particular metaphor has performed well then quantifying it for the achieved difference in performance will further enhance the understanding of temporal metaphors. This may be valuable for practitioners and researchers. Such data needs to be reported with both the mean and median with their respective confidence intervals to have a better insight about the central tendency of the data. Medians may be good to report in non-Gaussian or positively skewed data (as in task time data in usability studies [88]), but sometimes due to resistance to extreme values, they do not capture the performance difference adequately. This is evident from our results where the medians are close to each other even when systems have a significant difference in performance statistically. Thus, the mean can be helpful in quantifying the effect size.

We also discussed assessing the statistical significance of the performance difference. We employed the Logrank test for this purpose, which is a form of the Chi-square test and widely used in survival analysis and life estimation techniques [19]. There are other similar tests like the Wilcoxon-test and Tarone-ware test which can be used for different applications based on the need. These tests differ in sensitivity to the survival curve and put different weights to initial events and events that are happening later in the time. For more discussion on the comparison of Logrank and other applicable tests in survival analysis and the effect of right-censoring, please refer [83].

This objective evaluation method is applicable to other application types and is well suited for large-scale, automatic usability evaluations. Along with the event of interest, user wait time behavior can be automatically logged in system and software products. Automatic system logging enables a large-scale data collection for usability assessments without relying explicitly on user response through a form or survey.

2.3.6 Next Steps

Having an objective assessment technique is the first step towards our goal of designing temporal metaphors for auditory systems like Helpline. Previous research on wait time has not explored objective assessment techniques. We identified the need for an objective assessment technique to minimize the influence of any contextual variables, generating objective outcomes like *How long a user will wait*. We developed a method for analyzing the wait time data using survival analysis, which can be the basis for an objective assessment based user's wait time rather than perceived time. Survival analysis or other lifetime estimation techniques (like Accelerated Failure Time Model [48] and Cox regression [55]) are popular among reliability engineers and medical scientists, and we have indicated its potential use in HCI for wait time experiment. Next, we use Survival analysis to investigate the performance of different design parameters of temporal metaphors. This could be helpful in designing metaphors that can hold callers longer those are waiting to get information from an assistant or an expert answering the calls received at helpline.

2.4 Theoretical Models for Comparing Auditory Progress-bar designs

In this section, we will describe properties of Telephonic queues and the possible variation in Auditory Progress Bar design that we will compare through a real world experiment.

2.4.1 Telephonic Queue

To formulate the research problem around waiting callers in a telephonic queue, it is required to understand different aspects of it:

- *Invisibility of the Queue:* The telephonic queues are invisible to the caller as the position in the queue and progress of the queue are not visible to the caller until and unless provided by the system. A customer in a telephonic queue may behave differently than from a physical queue. A lengthy physical queue may look irritating initially. However, once a decision to wait is made by a customer than advancement in the queue will bring more positive sense towards reaching the end goal. On the contrary, on the telephonic queue customers who are unaware of their position and length of the queue may take a decision to wait initially, but will feel more inclined to leave when put on hold for a longer time.
- *Patience of a caller:* Caller patience is another important factor in any queue (physical or telephonic). It can be measured as the time caller is willing to wait. There are several factors which affect the waiting time of a caller, e.g., cognitive state (stress, purpose etc.) of the caller at the time of the call, previous experience with the same system, etc. A caller under stress who is in some sort of emergency may not be that patient as a normal caller; in contrast, a caller who requires consultation or needs to report the grievance may wait longer than a normal caller. Similarly, previous experience with the system can act as assurance in the initial phase of waiting as caller may have an idea of approximate system response time. The uneasiness and impatience factor will start developing after the caller surpasses perceived average waiting time.
- *Time perception by a caller:* Human perception about time is different from the actual time. According to the maxim in the psychology literature, "uncertain wait is perceived longer than known finite wait" [67]. In the absence of known finite wait or estimate of the *required wait* on a telephonic queue, callers will generally perceive waiting time lengthier than the cases where they have an estimate of the required wait. Secondly, a constant attention towards the passage of time is also perceived as longer than otherwise.

2.4.2 Research Directions

Research on telephonic queues has shown that delay announcements can impact the waiting behavior of the caller. We believe that different messages have a different impact, and delay announcements and their reaction to the caller are not explored in depth. In this section, we formulate some research directions which are required to understand the underlying impact of the delay announcement on a caller:

- *Type of Information content:* Probably the most important question in designing a delay announcement is about the information content that is intended to be passed to the caller. One delay announcement can convey the position of the caller in the queue whereas another can convey expected time left, or it can simply be about requesting to wait more. Once we select an information type, we can further study the trade-off associated with the different value of information content. For example, if we select time as information type than a small value for announced waiting time will have more callers for a short time but with a large value for announced waiting time we will have few callers for a long time.
- *Frequency of Information delivery:* Delay announcements are given after a certain time gap. The impact of the different time gap between successive delay announcements is an important design parameter that we have studied in our experiment.
- Unverifiable information: An important observation in delay announcements is that the information provided to the customer is unverifiable. There is no way by which a caller can easily verify the information, e.g. if caller position is announced then the caller can not verify if the announced position is indeed her actual position. This leads us to ask if could falsify this information to lure callers to continue waiting? Would it work?
- *Expiry of expected waiting:* A caller may decide to leave if the time given by the delay announcement surpasses, because the caller may question the credibility of the system. However, an alternate hypothesis is that the caller may wait considering herself closer to end goal after having invested significant time in waiting for an agent.

2.4.3 Experiment Design

To study the impact of different delay announcements we designed different delay announcement and deployed it at 181 women helpline. Here, we explain our system and study design and the hypotheses.

System design

We designed 7 different systems (P1, T2, T3, T4, M5, L6, L7) with different types of delay announcements and compared them with a baseline system (B0). The system Position (P1) and Simple Time (T2) explores the role of announcing position and time-based information respectively. The system Dynamic Time (T3) is also a time-based system where estimated time can increase or decrease in the subsequent announcement. Stretched Time (T4) is designed to explore the non-verifiability of information where callers are deceived to wait more by artificially shrinking or expanding the time. Simple Message (M5) is similar to Baseline (B0) but every 20 seconds it explicitly announces to wait more. The system Long Time (L6) announces a high waiting time to explore the difference in system performance in comparison to a small value announced by T2. We have designed the another system similar to L6 termed as Less frequent Long Time (L7) to explore the effect on system performance by lowering the frequency of delay announcement. Detailed description of each system is given below:

1. *Baseline (B0):* This system is the existing system of 181 helpline which does not announce any delay announcement to the caller. It has a pre-recorded audio message which repeat-

edly plays welcome message to the caller in two languages i.e. Hindi and English.

"Welcome to Chief Minister's Women helpline!!"

2. *Position (P1):* This system is same as B0 but announces caller position in the queue. The caller who calls to the helpline first gets a welcome message similar to B0 followed by their position in the call. Then the system keeps updating position of the caller after every 20 seconds.

"Your waiting number in this queue is XXX!!"

3. *Simple Time (T2):* This system is same as B0 but announces an expected initial waiting time of 100 seconds to each caller. The system keeps updating waiting time after every 20 seconds. The value of each announcement against actual wait of the caller is shown in Table 2.5. This is a simple system which deducts 20 seconds from the last announced time and plays it in the subsequent announcement as the time left for the caller. After the 5 announcements, actual wait time becomes zero and hence system stops announcing any delay announcement for the rest of the waiting period. This system helps in understanding whether a caller leaves the system after initial waiting time is over, or she will continue to stay.

"Your waiting time is X seconds!!"

Table 2.5: *Announcement in different delay uses the following table to announce the expected time left. First column shows actual time a caller has waited at that announcement*

Actual wait (In seconds)	T2	T3	T4	L6	L7
16*	100	100	100	200	200
36	80	120	70	180	-
56	60	110	60	160	160
76	40	70	50	140	-
96	20	90	40	120	120
116	-	70	30	100	-
136	-	60	25	80	80
156	-	60	20	60	-
176	-	40	15	40	40
196	-	30	10	20	-
216	-	20	5	-	-
*at t = 0 a welcome message having audio duration of 16 seconds is played immediately followed by respective delay announcement					

4. *Dynamic Time (T3):* This system is same as T2 and announces the time left for the caller after every 20 seconds. However, in this system we simulated a situation where expected time of the caller may not necessarily decreases from the last announced time. The system

increases wait time twice i.e. one after actual wait of 20 seconds and the second time after actual wait of 80 seconds (see Table 2.5).

"Your waiting time is X seconds!!"

5. *Stretched Time (T4):* This system is same as T2 and announces the time left for the caller after every 20 seconds. However, in this system, though the announced time always decreases, but the decreased time can stretch or shrink and does not correspond to the actual decrease in waiting time. In our experiment, time has stretched to 30 seconds in first 20 seconds but has shrunk all the time in the later announcement (see Table 2.5). The user of such system always feels the expected time left for waiting has decreased from the previous announcement but we have adjusted the pace of time by changing the amount of time decreased (differs from the actual wait of 20 seconds) from the last announcement.

   "Your waiting time is X seconds!!"

6. *Simple Message (M5):* This system does not announce any time or position information in its delay announcement. It is a simple system which after every 20 second announces that

   "You are in the queue. Please wait!!"

7. *Long Time (L6):* This system is similar to the simple time (T2) but starts it by announcing an initial waiting time of 200 seconds instead of 100 seconds. In the subsequent announcement (after every 20 seconds) the system reduces waiting time by 20 seconds (see Table 2.5).
8. *Less frequent Long Time (L7):* This system is similar to Long time (L6) and starts by announcing an initial waiting time of 200 seconds. However, in the subsequent announcement happens at a time gap of 40 seconds and the system reduces waiting time by 40 seconds instead of the time gap and time reduction of 20 seconds (see Table 2.5).

Study design

Since, we have 8 systems in total, we designed to route every 8^{th} call to the same system. For each new call, system generates a numerical call ID in an incremental fashion. This call ID is operated with modulo 8 to decide the system to handle this call. Table 2.6 shows assignment based on output of the modulo 8 on call id:

We decided to take only first call of a user for our analysis to avoid any bias that may come because of user's experience with the system. We took each new call as a different call even if it came from some previous number logged in our database and assigned it to one of the systems as per above algorithm. Later, for the analysis we removed any subsequent calls from the same number and took only the first call of any caller for analysis.

2.4.4 Data Collection

For the analysis, we collected data for 12 days. The system logged following attributes associated with each call:

Table 2.6: *Assignment of system based modulo 8 operation on the system generated call id.*

Output(=Call ID % 8)	System
1	BO
2	P1
3	T2
4	T3
5	T4
6	M5
7	L6
0	L7

- *Caller ID:* For each call connected to our system we logged the caller id of the number and mapped to a different set of numeric keys to preserve the privacy of caller.
- *Time stamps:* For each call we logged time stamp of call start and call end
- *System allocated:* For each call we logged the system name that is allocated for announcing the delay according to algorithms discussed earlier.
- *Announcement details:* For each call we maintained a detailed entry into the database related to each announcement made by the system. We logged the number of announcements listened in each call and the corresponding value announced in each announcement along with the time stamp of each announcement.
- *Queue Time:* We explicitly log the time spent by each caller on the IVR waiting for an agent to answer her calls.
- *Call status:* We maintained the log for each call whether it was answered by an agent or abandoned by the caller.

In a span of 12 days, we received a total of 16,922 calls. We analyzed 16,922 calls out of which 9,612 were unique callers. We analyzed the first call of these callers on our system which received at least one delay announcement from the system. We found that out of 9,612 calls, only 926 calls received one or more delay announcement (calls with Queue Time >= 16 seconds received at least 1 delay announcement, see Table 2.5). Call distribution across the system differs in number because repeated calls from earlier callers were removed from the analysis. Further distribution of call across each system is summarized in Table 2.7.

2.4.5 Results and Analysis

The objective of each system is to hold the caller for a longer time so that the number of abandoned calls can be decreased. Thus, the percentage of abandoned calls (hereafter referred as abandon rate) will be minimum for the best system. Another comparison for measuring the system ability to hold a caller is to calculate the probability that the caller will wait for X seconds on each system. The best system will have the highest probability for any given value of X seconds (waiting time). We call this probability as caller survival probability. The survival analysis is a branch of statistics which is widely used in statistical analysis of call center and medical sci-

Table 2.7: *Distribution of First Call with at least one delay announcement of Unique Callers (N=926)*

System	Answered Calls	Abandoned Calls	Total Calls	Abandon (in %)
B0	68	57	125	45.60%
P1	62	61	123	49.60%
T2	71	60	131	45.80%
T3	74	56	130	43.10%
T4	55	48	103	46.60%
M5	55	45	100	45.00%
L6	49	56	105	53.30%
L7	64	45	109	41.30%
Overall	498	428	926	46.20%

ences to calculate such probabilities [55, 63]. We have used Kaplan-Meier estimator to perform the survival analysis of the queue time (waiting time) of calls on each system [36].

In this section, we will compare the performances of different systems. We will analyze the 8 systems based on abandon rate and survival analysis to understand the system dynamics of each system. We will also establish that delay announcements in emergency helpline can influence the waiting behaviour of the callers.We have reported both the mean and median survival time. In quantitative analysis median is a good metric for performance comparison when dealing with non-Gaussian or positively skewed data (as in task time data in usability studies [88]), but sometimes due to resistance to extreme values, they do not capture the performance difference adequately.

Abandon Rate

Abandon rate is a crude measure to see the effectiveness of our system. Abandon rate is defined as the percentage of total calls which were unanswered. A caller waiting longer has more chance of her call getting answered than a caller who waits for a shorter time. Hence, a better system will observe an abandon rate lower than the other systems. However, a caller waiting for longer is one of the factors, and there are multiple other factors such as congestion in the line and number of agents which also affect the abandon rate of the system. Table 2.7 shows the abandon rate of each system.The system L7 experienced minimum abandon rate. Chi Square test did not show any statistically significant difference between the abandon rate of the systems (Chi Square ($X^2(7) = 4.38$, $P = 0.735$). To further analyze the waiting behavior on each system, we will focus on waiting time of the callers through a statistical method (Survival Analysis).

Caller Survivability

We did survival analysis (Kaplan-Meier), to analyze the probability of callers waiting for X seconds or more in the queue across different systems. In this analysis, *queue time* is the *survival*

Table 2.8: Estimation of survival mean and median

	Mean^a			
System	Estimate	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
B0	77.637	14.169	49.865	105.409
P1	74.957	8.558	58.183	91.731
T2	117.406	17.335	83.429	151.383
T3	176.441	20.68	135.908	216.973
T4	73.468	9.903	54.059	92.877
M5	84.482	10.152	64.585	104.38
L6	68.178	9.858	48.857	87.5
L7	86.817	10.02	67.178	106.455
Overall	109.346	8.656	92.38	126.313
a. Estimation is limited to the largest survival time if it is censored.				
	Median			
System	Estimate	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
B0	43	5.744	31.741	54.259
P1	42	4.744	32.702	51.298
T2	56	13.091	30.342	81.658
T3	51	4.31	42.553	59.447
T4	49	9.665	30.056	67.944
M5	71	15.632	40.361	101.639
L6	43	6.374	30.508	55.492
L7	54	15.411	23.795	84.205
Overall	48	2.526	43.048	52.952

time of the caller, *call abandonment* is the *observed event*, and *answered calls* are treated as *right censored data*[6]. We did survival analysis of 926 calls which were left after removing repeated calls from the same telephone number and was not dropped while listening to the welcome message (queue-time<16 seconds). Table 2.8 shows mean[7] and median[8] estimation of survival time for different systems. Figure 2.4 shows survival probabilities of such callers for different systems on a logarithmic scale.

We did further analysis to interpret the survivability curve through different Chi-Square statistics, i.e., Logrank (Mantel-Cox), and Breslow (Generalized Wilcoxon).,as these tests give different weight to different parts of the survival curve [9].

[6]A right censored data is a data point that is known to be above a certain value but it is unknown by how much. In this experiment, the queue time of answered calls represent right censored data as caller could have waited more than the queue time but unknown by how much.

[7]Mean survival time is estimated as the area under the survival curve

[8]The median survival time is calculated as the smallest survival time for which the survivor function is less than or equal to 0.5

[9]To read more about rank invariant test please refer to [83]

2.4.5.1 Acceptability and Retainability

For a user population, survival curve indicates overall satisfaction at different waiting-time. But with increasing wait-time number of satisfied users decrease and it is important to find that whether the decrease in satisfaction is due to mismatch of *expectation* or different *perception*. Thus, two things we are particularly interested in finding are:

- *Acceptability:* Upon listening to the first delay announcement, the caller may drop the call immediately because announced value is not worth waiting. For instance, estimated waiting time for a caller is 120 seconds and his/her position in the queue is 2. In this case, it may happen that a caller may decide to wait upon hearing that his/her position in the queue is 2 and may not wait if the caller gets the same information as waiting time (instead of position) i.e. the caller has to wait for 120 seconds. The system with less convincing delay information will see a higher call drop at the starting of the survival curve i.e. the system has less acceptability.
- *Retainability:* Among the callers who decide to wait after hearing the initial announcement may differ in waiting time from one system to another. A system may be able to hold the callers significantly longer than the other systems and the difference can be seen through the tail of the survival curve.

Acceptability models the user expectation in the system. Announcing an initial wait time more than user expectation may result in the immediate abandon of call. The second attribute i.e., Retainability models the user perception in the system. If a waiting user perceives that (s)he has waited for more than sufficient than (s)he will abandon the call. The statistical test can be selectively applied to study the parameter of interest. Log-rank is sensitive towards the end (or Right Tail) of the survival curve which may help to identify if a system is able to hold the callers more than the other. Breslow test gives more weight to early failures of survival curve, thus helps to understand if the system is acceptable to the callers at first delay announcement.

Curve Fitting

In our experiment, survival curve of each system shows the characteristics of an exponential curve (see Figure 2.4). However, comparing the performance characteristics of different systems is difficult through survival curves as they intersect each other multiple times. To overcome the problem of multiple time intersecting survival curves, we model the system survival probabilities through exponential distribution as shown in Figure 2.5. Resulting exponential curves intersect each other at most once. Based on the exponential model, we estimate the value of acceptability and retainability of each system for comparison.

The exponential function is given below:

$$Y = \alpha e^{(-\lambda X)} + C$$

'Y' represents survival probabilities, and X represent corresponding queue time. The constant term α represents the acceptability of the system. The acceptability of the system is the survival probability of the user who are willing to wait after the first announcement (i.e. QueueTime = 16+ δ where $\delta \to 0^{+}$). As the system with less convincing delay information will have a higher

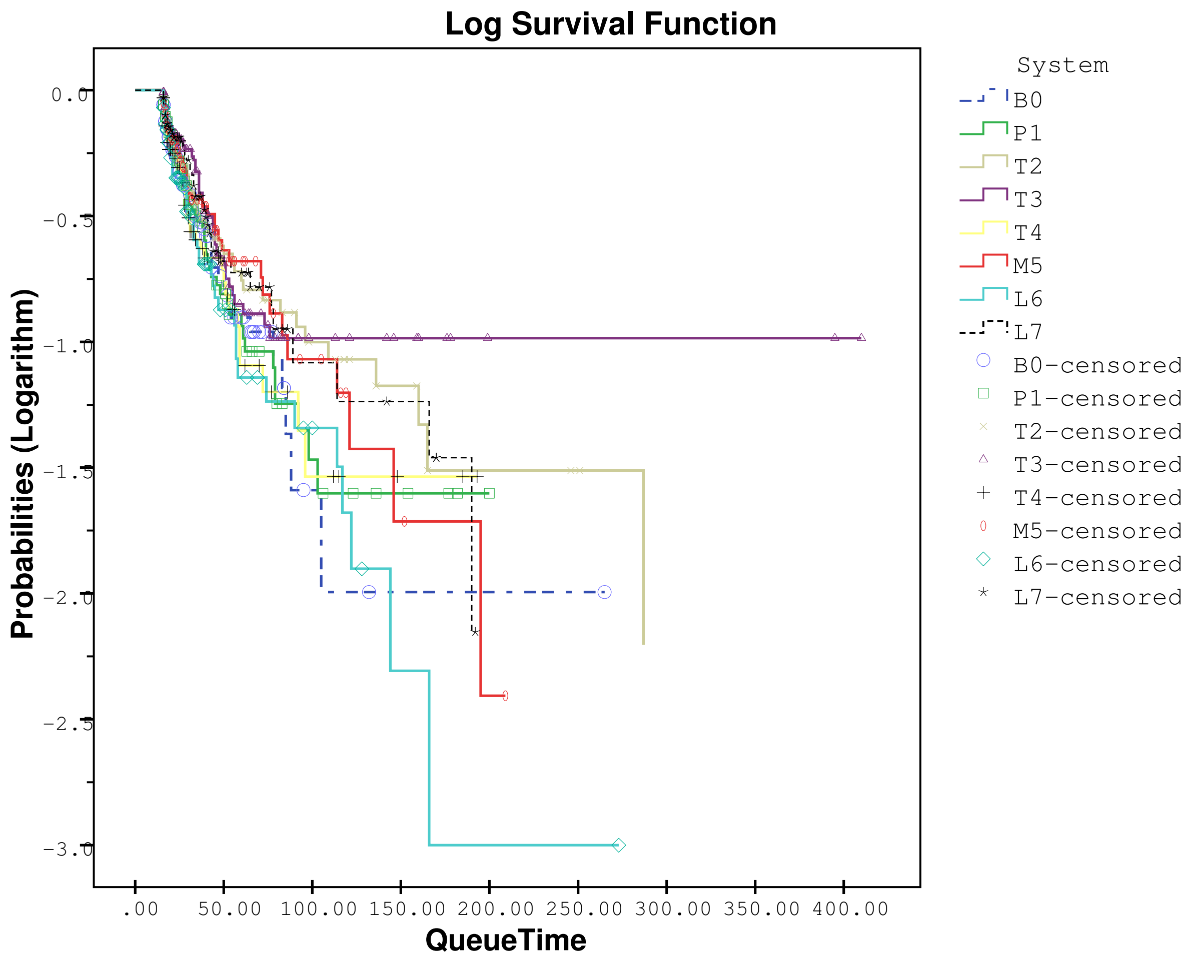


Figure 2.4: *Survival Curve: Abscissa shows the survival probability against the plotted queue-time on ordinate for each system. The markers are the censored data (refers to answered call in this case). A higher value (on the abscissa) of the curve means a better system in promoting caller to wait more.*

call drop at the starting of the survival curve and thus resulting in a smaller value of α. We can see that at X=0, the value of Y is obtained by extrapolating the curve for X < 16. Hence, the estimated value of α through extrapolation of survival probabilities at X=0 may be greater than 1.

Here, λ is the decay rate of the exponential function. This measures the rate at which caller leaves the system among those who chose to wait for more after the first announcement. A smaller value of λ means high retainability of the system. The constant *C* is smoothing parameter to minimize the standard error for the fitted distribution. Thus, the final curve for a system can shift up or down based on values of (α, λ). **The overall system performance is measured the area under the curve (i.e. $\int_{16}^{\infty}(\alpha e^{(-\lambda X)} + C)dx$).** In the case of a non-overlapping curve, we interpret a system has better overall performance if its curve is placed on a higher side of ordinate (Y-axis). The value of λ and α for each system are stated in Table 2.9.The system P1 has the

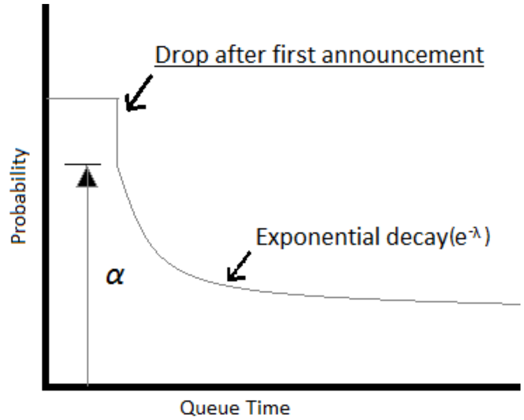


Figure 2.5: *Theoretical Model: A sharp decrease in survival probability at X=16 models the callers left the system after listening to first delay announcement. The system leaving behaviour of remaining callers are modeled with an exponential function*

highest acceptability (α=1.24) and system M5 has best retainability (λ=0.016).

Table 2.9: *Estimated decay rate for each system. A low value of exponential decay shows that system can retain caller for longer*

System	λ			α		
	Estimate	Lower Bound	Upper Bound	Estimate	Lower Bound	Upper Bound
M5	0.016	0.013	0.018	1.032	0.977	1.087
T2	0.020	-0.058	0.092	0.95	0.893	1.014
L7	0.022	-0.043	0.087	1.074	1.024	1.123
B0	0.025	-0.032	0.081	1.153	1.113	1.194
L6	0.027	-0.067	0.122	1.188	1.148	1.228
P1	0.032	-0.013	0.078	1.248	1.235	1.261
T4	0.034	-0.075	0.143	1.186	1.154	1.217
T3	0.04	-0.083	0.163	1.197	1.177	1.217

Through curve fitting via optimization, we produced the final fitted model for each system as shown in Figure 2.6. We can see time-based systems T2 and T3 have overall performed better than the other systems. From Figure 2.6, we conclude that the overall system performance have following relative order (i.e. A curve on higher side of ordinate is a better system as it will have larger area under the curve):

$$\mathbf{L6} < \mathbf{B0} < \mathbf{P1} < \mathbf{T4} < \mathbf{L7} < \mathbf{M5} < \mathbf{T2} < \mathbf{T3} \tag{2.4}$$

The order of retainability and acceptability (as given below) of the system is assessed through the decay rate (i.e. λ) and α of the estimated exponential is given in Eq 2 and 3, respectively.

$$T3 < T4 < P1 < L6 < B0 < L7 < T2 < M5 \tag{2.5}$$

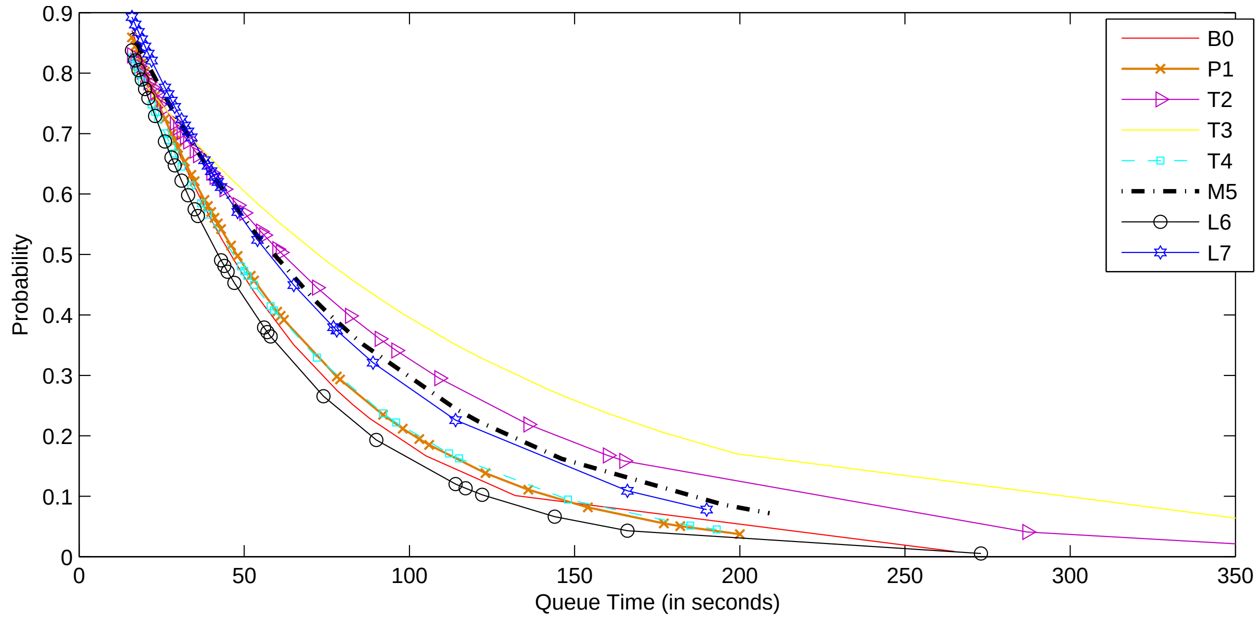


Figure 2.6: *Curve Fitting: System performance after curve fitting via optimization. We can see system differs in starting point of the curve at X=16 and has a different decay rate.*

$$T2 < M5 < L7 < B0 < T4 < L6 < T3 < P1 \tag{2.6}$$

The order of retainability and acceptability is different from the order of overall system performance (comparing E2 and 3 with Eq 1). This shows that both the factors have an implication on the overall system performance. For example, System P1 has best acceptability (see Eq3) but still appear very low in the overall system performance ranking (see Eq1), due to low retainability as shown in Eq 2.

Inferences

In this section, we interpret about different systems and their characteristics (Type of information, frequency, etc.) based on acceptability, retainability and overall system performance calculated through survival mean, median, curve, decay rate, etc. To establish any argument for statistical significance ($p<0.05$), we will use Breslow and Logrank for acceptability and retainability, respectively (see Table 2.10).

1. *Baseline (B0))*: We now establish whether delay announcement system is effective or not. The ranking based on overall system performance shows that all the systems except L6 have performed better than Baseline system (see Eq 1). This shows that informed delay announcements are effective in holding the callers for long. The reason for the bad performance of L6 is discussed later with other time-based systems.
2. *Simple Message (M5)*: The system M5 does not provide any delay information apart from consistently telling the caller that "You are in a Queue". We found that the system M5 performed much better than the Baseline B0 (see Eq 1 and Figure 2.6). M5 shows that explicitly informing the caller to wait is much more effective than B0 style of repeatedly

Table 2.10: *Statistics: Bold faced values shows that pair (T3, T4) and (T3, L6) are statistically different (p<0.05)*

System		Breslow statistics are shown in left diagonal	B0	P1	T2	T3	T4	M5	L6	L7	Log-Rank statistics are shown in right diagonal
B0	X^2		–	0.02	1.47	2.6	0.02	1.2	0.29	1.46	
	Sig.		–	0.89	0.23	0.11	0.88	0.27	0.59	0.23	
P1	X^2		0.05	–	1.08	2.74	0.12	0.73	0.76	1.19	
	Sig.		0.83	–	0.3	0.1	0.73	0.39	0.38	0.28	
T2	X^2		0.59	0.33	–	0.83	1.62	0.08	3.76	0.01	
	Sig.		0.44	0.56	–	0.36	0.2	0.78	0.05	0.91	
T3	X^2		2.67	2.36	0.69	–	3.3	0.53	6.57	0.26	
	Sig.		0.1	0.12	0.41	–	0.07	0.47	**0.01**	0.61	
T4	X^2		0.12	0.38	1.48	4.06	–	1.2	0.23	1.72	
	Sig.		0.73	0.54	0.22	**0.04**	–	0.27	0.63	0.19	
M5	X^2		0.72	0.43	0	0.4	1.23	–	2.5	0.02	
	Sig.		0.4	0.51	0.95	0.53	0.27	–	0.11	0.89	
L6	X^2		0.4	0.75	2.08	5.28	0.03	1.86	–	3.77	
	Sig.		0.53	0.39	0.15	**0.02**	0.87	0.17	–	0.05	
L7	X^2		1.33	1.06	0.17	0.15	2.27	0.1	3.16	–	
	Sig.		0.25	0.3	0.68	0.7	0.13	0.76	0.08	–	

playing background music and a welcome message without an explicit message for the caller to wait more. We will discuss later about the good retainability and poor acceptability of M5 by comparing it with Time-based and position based information.

3. *Time*: We compare five time-based systems (T2, T3, T4, L6, L7) based on overall system performance (see Eq 1); all the systems except Long Time (L6) perform better than the Baseline system (B0). To study the negative performance of L6, we compare it with Simple Time (T2) as the two systems only differ in the initial value of the time announced by them i.e., T2 starts with the first announcement as 100 seconds and L6 with 200 seconds. The Log-rank statistic shows (see Table 2.10) that the statistical significance of the performance difference in T2 and L6 is close to 0.05 which indicates that systems have significantly different retainability. We empirically find that the announcing large waiting time has a deterrent effect on the caller.

   Further, we find that time-based systems have lesser retainability when compared with the Simple Message M5 (see Eq 2). We explain this performance difference based on the working of cognitive timers[10]. A cognitive timer provides the information regarding subjective time and has a direct positive correlation with the amount of attention focused on the passage of time. Thus, the time is perceived as passing slowly when attention is focused primarily on time [22]. Thus, we conclude that time-based system should have an effective mechanism to lower the attention towards the passage of time. One such mechanism is to lower the frequency of delay announcement, which we will discuss next.

[10] Cognitive Timers are mental vehicles for processing temporal information

4. *Frequency*: The negative effect of high frequency can be seen by comparing the performance of Long Time systems L6 with L7. The system L6 and L7 differ only in frequency of delivering the delay information. The overall performance of the system L6 is worse among the 8 systems (see Eq 1 and Figure 2.6). While the system L7 has attained a better overall system performance by slowing down the frequency of delivering the delay message (see Eq 1 and Figure 2.6). At high frequency, the time-based system puts more attention on the passage of time. Thus making caller perceive passing time longer than the same time-based information delivered at low frequency. This results in better reatinability of L7 (see Eq 2). The Log-rank statistics for significance difference between L6 and L7 is slightly greater than 0.05 (when checked up to third decimal place).

5. *Position*: The system P1 announces position in the telephonic queue as delay information. The system P1 is designed to provide a sense of progress without putting any attention towards the passage of time thus avoiding the negative effect of cognitive timer on system retainability. We hypothesize that P1 should be best performing among the 8 systems. We find that P1 has best acceptability (see Eq 3). However, Eq 1 shows that the overall system performance of P1 is only better than Baseline (B0) and Long Time (L6). Figure 2.7 shows survivability curve of P1, Baseline (B0), and Simple Time (T2). Survivability curve of P1 and B0 are almost similar, although Chi square statistics (see Table 2.10 and 2.10) does not show any significance for statistical similarity (p<0.95). In fact, the survival curve shows the performance of P1 drops below B0 between Queue time of 50 and 70 seconds.

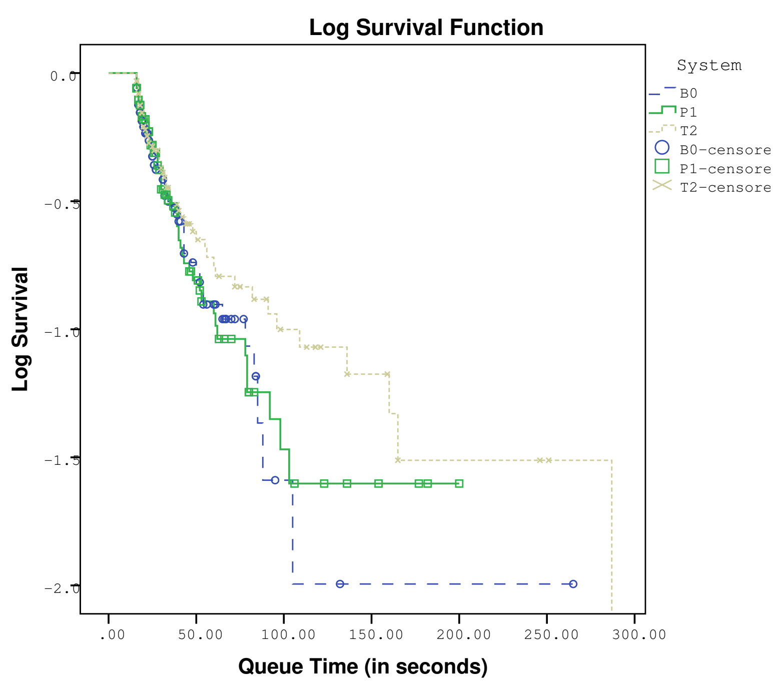


Figure 2.7: *Survival Curve: A comparison of survivability of callers on Position (P1), Baseline (B0), and Simple Time (T2). Markers on each curve show the corresponding censored (Answered calls) data.*

We investigate for a possible explanation of these counter-intuitive results. We find that position value announced to the caller in P1 range from 1 to 8 (M=1.609, SD =1.19). This shows that majority of the callers got their position in the queue as either 1 or 2. We believe that such a small value announced to the callers as their position in the queue created a high expectation among the callers for their call getting answered early. This may be the reason that some people who could have waited without any information became impatient later (poor retaianability) due to uncertain wait time[11] and high expectation. Such impatience of caller leads to increase in the abandon rate. We found that the abandon rate of P1 (49.60%) is higher than Baseline B0 (45.60%).

We conclude that position based system which has the best acceptability (see Eq 3) has very poor retainability. Thus, such system should not be used for a scenario where the system has high waiting time.

6. *Unverifiable information*: System Dynamic Time (T3) and Stretched Time (T4) manipulate the time information announced to the callers differently. We hypothesized that always decreasing estimated time in T4 is better than the Dynamic system (T3), where estimated time can increase in a subsequent announcement. This attribute of T3 could have a negative effect on the callers who were promised to wait for a lesser time in an earlier announcement. Secondly, we also hypothesize that artificial shrinking or expanding time can help us to slow down the perceived waiting time as the caller may or may not be watching their wristwatch or mobile phone to confirm the actual time passed with time announced by the system T4.

   Contrary to our hypothesis, T3 has the best overall performance (see Eq 1). T3 is not just better than stretched Time (T4), in fact, T4 has the minimum estimated mean and T3 has the maximum estimated mean among the 8 systems, which is more than the two-fold of the Baseline (B0). The initial part of Survivability curve (Figure 2.4) shows a significant behaviour change in T3 and T4 after X=50 seconds. The Breslow-Wilcoxon statistics also shows that the difference between T3 and T4 is also statistically significant ($p< 0.05$). Therefore, we reject the first hypothesis.

   The relatively poor performance of T4 in comparison to T3, we may infer that for a short time window of 20 seconds between each delay announcement, artificially reducing the time window from 20 seconds to 10 (by a factor of 2) or 5 (by a factor of 4) seconds may be easily noticeable by the caller, which may lead to distrust in the system. However, multiple increase and decrease in the estimated time left by Dynamic Time (T3) may have created some uncertainty about the system but caller probably may not loose the trust in the system.

7. *Expiry of expected waiting time*: We observe the caller behaviour after the expiry of expected waiting time for Time based system (T2, T3, T4, L6, L7). Expiry time for T2, T3 and T4 is 100 seconds and for L6 and L7 it is 200 seconds. Thus, we observe survival curve of T2, T3, and T4 at 100 seconds and L6 and L7 at 200 seconds and find that the survival probability is non-zero at the expiry time for all the system. This suggests caller

[11] Uncertain Waits Are Longer than Known, Finite Waits [67]

do wait after the expiry time. We can further confirm this by comparing estimated mean of simple time (T2) with its expiry time. The simple time system (T2) announced an initial waiting time of 100 seconds. After the expiry of 100 seconds, we observe that callers may continue to wait on T2. The estimated survival mean of T2 is 117.4 seconds (95% CI [83.4, 151.8]) which is higher than announced 100 seconds. There can be many reasons for such an observation, like caller feels that they are very near to end goal or they are not aware of the waiting time expiry. Since we could not interview the callers, the exact reason for such an observation can not be determined with surety. However, this interesting finding indicates that the system may have some time to respond to its caller even after the expiry of expected waiting time.

2.4.6 Experiment Summary

We have shown that the delay announcement works for the caller in distress. Towards this, we studied different design parameters for a delay announcement system and evaluated them in terms of acceptability, retainability and overall system performance. This work has the following contribution for the research community:

1. Design considerations for delay announcement for distress caller

- We showed that delay announcements in emergency helpline can influence the waiting behaviour of the callers. All the systems except L6 have performed better than Baseline system and are effective in holding the callers longer.
- A simple message explicitly informing the caller to wait is much more effective than repeatedly playing background music and a welcome message without an explicit message for the caller to wait more.
- We find that the announcing large waiting time has a deterrent effect on the caller.
- A better overall system performance can be achieved by slowing down the frequency of delivering the time-based delay message.
- Position based delay information system created a high expectation among the callers for their call getting answered early may result in poor retainability.
- Artificially reducing the time window from 20 seconds to 10 (by a factor of 2) or 5 (by a factor of 4) seconds may be easily noticeable by the caller, which may lead to distrust in the system.
- Multiple increase and decrease in the estimated time left may have created some uncertainty about the system but caller probably may not lose the trust in the system.
- We found that the survival probability is non-zero at the expiry time for all the system. This indicates that the system may have some time to respond to its caller even after the expiry of expected waiting time.

2. Quantitative evaluation of the system In this study, we developed a theoretical model based on survival probabilities calculated through the standard statistical tool (Kaplan-Meier). We introduced two novel concepts of acceptability (α) and retainability (λ) based on caller reactions to first and subsequent delay announcements. Our theoretical model provides a good understanding and combines them to evaluate overall system performance. We believe that it is a sound technique for quantitative analysis for such system that will have applicability in scenarios other than emergency helpline. Such quantitative analysis is also required when we conduct study on a large scale and collecting user response from all of them becomes a difficult task or users are not available to give feedback similar to the situation of emergency helpline.

2.5 Conclusion

The use of Auditory Progress bar as Temporal metaphors for Helpline system showed promising result towards inducing a longer wait time in caller behavior. This provides agent more time to service a call and reducing the chance of caller getting unanswered. This solution is our effort to address problems of helpline that cannot scale due to the limited workforce. The problem of limited workforce demands more automation so that dependency on human workforce can be reduced. Towards this, the use of voice application has been introduced to automate the information exchange wherever possible. In voice application system, a user can store information in the form of audio recording that can be accessed later by another user through voice-based menu navigation. In the next chapter, we focus on such menu based voice application that is increasingly gaining significance interest among the research community.

Chapter 3

Navigation Time: Information exchange through menu based voice response system

Interactive Voice Response (IVR)[1] systems have helped in automating information exchange through the telephonic medium. Automating the information exchange process reduces the cost of employing workforce and easily scales up with the growing call traffic. In several use cases, voice application also serves as a front end to Helpline system where the user first interact with Automated Voice application to resolve their query. If voice application fails to resolve users' query, then it provides an opportunity to connect with a human operator to address the query. These use cases have resulted in the massive commercial deployments of the voice applications. The commercial market for voice applications is growing at a rapid CAGR of over 10% and is expected to reach a global market of USD 2.78 billion by 2017 [33].

In the IVR systems, the information is made available to the end-users, over an interface of simple key presses or voice commands, using a pre-defined menu structure to answer user queries. Based on the input from the user, for each menu item, the response is extracted from a database connected with the voice application. The menu structure and the information that it retrieves from the database usually remain fixed throughout the lifetime of an IVR System, and any change requires manual intervention. Every information, provided by an IVR System, has a unique contextual value. The IVR System tries to capture the context at a very coarse-grained level by offering a fixed menu for traversal. The fixed menu structure by its very nature ignores the dynamic nature of context and provides information as envisioned by the developer of the system at the beginning. For example, in the case of Indian Railway IVR System, the menu structure to access information remains fixed throughout the year and offers the same interface for all the users, thus not taking into account contextual factors like nature of queries, caller abilities to interact with the system, or interaction pattern.

In this work, we focus on 'time' as one of the critical contextual factors that affect the information sought by the user. Let us take an example from real-life where the nature of the information sought from a system by its users changes with time: FAQs database on the sites of colleges and universities websites provide answers to frequent questions that a prospective student may have in mind. The FAQs grows with time and usually ends up with a long list of

[1] "Interactive Voice Response " and "Automated Voice Application" are used interchangeably.

questions and associated answers. However the nature of questions, that are accessed at a given time, also depend on external factors; for example, before the entrance exam for admissions, queries are mostly related to entrance exam dates, syllabus, eligibility criteria, etc. while after the examination, questions change for results, fees structure, hostel accommodation, etc. Here, it is quite clear that an external event, i.e., the examination has a direct impact on the questions that get the most attention. The menu structure of most of the present-day IVR system is also designed to answer commonly occurring questions same as FAQs. Thus, IVR Systems with static content end up with either providing non-relevant information or overloading user with the task of navigating to the desired information in a highly complex structure with lots of menu items requiring key presses and repetition of instructions which are not always very clear. This results in wasting the time of the user in finding the information they need and adding to the frustration of using the IVR system. In current IVR System, such shortcoming can only be overcome by manual intervention, i.e., by either changing the menu structures or editing the information base from time to time by constantly monitoring the system and then changing it accordingly, however this solution is neither scalable nor dynamic enough to be employed in real-life systems which have high availability requirements. This has motivated us to study, in detail, the design issues of IVR systems and to build an IVR system that can adapt itself to the changing information need of the users over time.

3.1 Overview of Voice based Telephonic System

In this section, first, we describe the different ways an automated voice application can be classified and designed using the Interactive Voice Response. Then, we explore research that has shown the impact of voice-based system in developing regions. Further, we discuss the literature that identifies and addresses usability problems of the voice-based system so that its performance can be improved.

3.1.1 Terms and Definition

Here we describe terms and definition related to user interaction on IVR system. To understand the user interaction better, let us walk through a typical usage scenario of IVR interaction. A user calls into an IVR system having some specific information need, e.g., knowing the current status of plane tickets or credit card balance, etc. The user then encounters the menu option prompting her for a key-press to select one of the options. Once a user selects an option, she receives further information on how to access next information; after a sequence of key-presses, the user receives the required information, to which she listens. If the user wants some other option, then again she has to find the correct menu option through various key-presses and then after reaching the correct information option, the user listens to the requested information. The whole interaction can be divided into three constituents:

- *Navigation time:* We define navigation time as the time required reaching the correct menu option, i.e., the menu item on which user receives the required information and not just another set of instructions.

- *Key-presses:* Different key-presses that are required in order to access the correct menu item.
- *Access time:* The access time is the time taken in listening to the required information.

Thus based on above definition, the term call duration is defined as a summation of *Navigation time* and *Access Time*:

$$CallDuration = Navigation\,Time + Access\,Time$$

Our focus is to reduce navigation time so that user reaches the desired information as quickly as possible.

3.1.2 Elements of Interactive Voice Response System

There are several methods to design and develop an Interactive Voice Response system e.g., speech and touch-tone (key-press). An IVR application can be as simple as a menu based application where the menu selection is done through pressing keys on any touch-tone[2] based phone. Today, almost all the phones are touch-tone and hence such a touch-tone application can be widely accessed via any phone. Touch-tone IVR can be considered as first generation IVR. They are widely used for languages where speech recognition is not very well supported. However, improvements in automated speech recognition for languages like English along with other improvements in the spoken system has led to advanced IVR applications where input modality is speech rather than touch-tone. Advanced speech-based IVR applications are trying to replace human in handling the calls. Figure 3.1 shows broad classification of IVR applications based on their ability to understand human speech.[3]

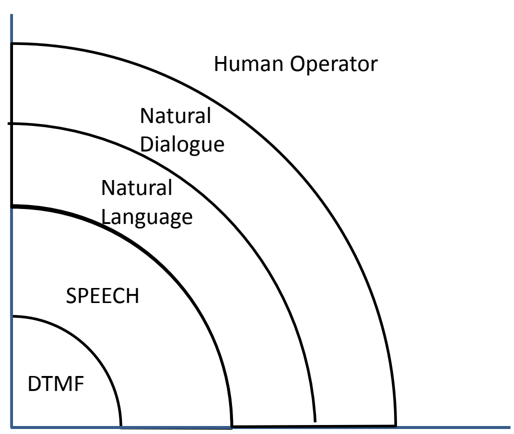


Figure 3.1: *Advancement in IVR systems in reaching capabilities of human operator to understand the speech.*

[2]Touch-tone is also referred as Dual Tone Multi-Frequency
[3]Source: `http://www.ibinfotech.com/services.html`

IVR classification

IVR system can be classified in several ways such as based on Interface design, System design, and their ability to replace human attendant in performing certain tasks. However, we focus on classification based on replacing human attendant or mimicking human in attending the call. Figure 3.1 shows different types of IVR with varying complexity in mimicking human behavior.

- **DTMF**
  Independent of speech recognition these systems provide a voice menu to be selected by pressing keys. Voice menu announces different tasks that IVR system can perform and the guidance to select a particular option by pressing a key. These are highly accurate as the key pressed by the user to perform an action can be correctly detected by IVR system.

- **Speech**
  The basic speech-based IVR is a menu based IVR system where the user can say some keywords to select a menu option rather than pressing keys. This speech based menu design ease outs the constraints of key-press based design where designers have to map every element of interaction within the set of 12 keys. Thus a speech based design gives more flexibility to develop interactive menu based applications.

- **Natural Language**
  The more sophisticated system is a natural language system where user can respond to a prompt through natural language with the only restriction that the user must give a relevant response to the system query e.g.
  System: *What is your age?*
  The valid responses to this query can be any natural variant of *I am 18 years old* , *My age is 18 years, 18 etc*. However, response to such query that *I want to go Los Angeles* will be an invalid response.

- **Natural Dialogue**
  Natural dialogue system has a subtle difference from natural language system. In natural language system, the user can respond to a system in natural language which is a free form of the speech but spoken response must have all the information which is required for current dialogue to proceed. However, in the natural dialogue system, a user can respond with partial information and the system can detect and ask for additional information. Thus, natural dialogue systems allow users to make queries in natural language and the restriction of context to what user can say is more flexible. The natural dialogue system tries to guess the context and intent of the user query and responds according to it.

We have categorized the IVR in 4 broad categories depending upon their capabilities. But choosing the type of IVR system to deploy is application dependent. With the increasing features of IVR the chances of making errors also increase. On extending the use of speech from the limited vocabulary to large vocabulary and from directed-dialogue to a natural dialogue making system more prone to recognition and semantic errors.

Infrastructure and Hardware

IVR system is composed of several hardware and software components. Figure 3.2 shows different software and hardware component of a standalone IVR system. Usually, the IVR system is a part of the bigger infrastructure in which network and communication elements participate in receiving and routing the calls across different networks. Some of the components like IP-PBX can be realized either in the form of hardware and software. In general hardware requirements across different IVR are same and the variability in IVR capability is achieved through software component. A software component can be as simple as decoding DTMF digits and playing appropriate audio prompt through some predefined mapping or it can be a complex dialogue system. An IVR system can be broadly viewed in terms of following three components.

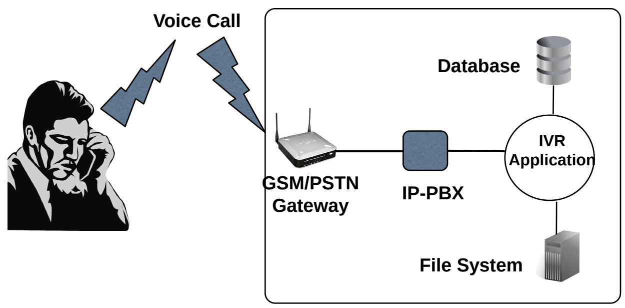


Figure 3.2: *IVR infrastructure: Voice call through GSM or PSTN network can connect to IVR application hosted on IP-PBX through a gateway. IVR application additionally utilizes database and file system functionality to perform desired task.*

- **Gateway**
  A gateway is a device which converts the signals of the telephone line to the signals of Session Initiation Protocol (SIP) and vice-versa. Users calls to the IVR system and connects through a gateway (PSTN/GSM) as shown in Figure 3.2. Different gateways may have different capabilities to handle the number of calls.
- **IP-PBX**
  IP-PBX stands for Internet Protocol-Private branch exchange used for delivering voice and video over a data network and can be interfaced with the Public Switched Telephone Network (PSTN). IP-PBX can be implemented as hardware or software. Mostly used IP-PBXs in the industry are Asterisk and FreeSWITCH.
- **IVR application**
  An IVR application is a software component that interacts with other software component to process speech or DTMF (touch-tone) inputs received from the user. An IP-PBX can host multiple IVR applications where a set of rules commonly known as dialplan defines which application is to be activated for a particular call. These IVR applications make the use of database and file system to store and fetch the desired information. The detail discussion on it is in next section.

Software Components

IVR application is a software component and can be implemented in several ways. Based on the complexity of IVR application it can have several components. Figure 3.3 shows the different components of IVR application. As shown in Figure 3.3, IVR can take input in the form of

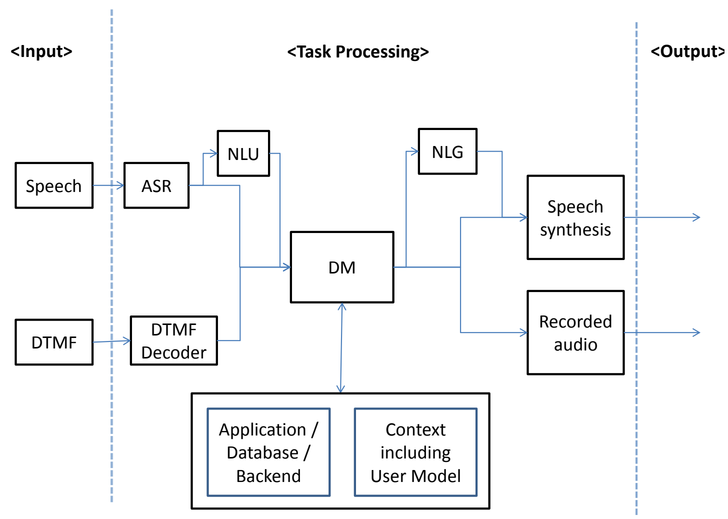


Figure 3.3: *Architecture of IVR system depicting input , output and task processing element*

speech or DTMF (touch-tone). If the input mode is DTMF, the generated DTMF tone is supplied to DTMF decoder which returns the digit representation of it. If the input modality is speech then supplied speech is passed to Automated Speech Recognition (ASR) engine. The ASR returns a textual representation of speech which can be either directly fed to Dialouge Manager (DM) or through a Natural Language Understanding (NLU) module which converts natural language to an internal representation of structured information to be used by DM. On receiving the information from the user, DM processes it with the help of application or database running at backend which may additionally use any contextual information or user modeling to improve the response of the system. Based on IVR design DM may output information in various ways. In the simplest way, it can play a recorded audio file or it can generate speech by passing textual input to a speech synthesizer. In a bit sophisticated manner it can produce an information as some internal representation which is fed to Natural Language Generator (NLG) to constitute the information in natural language. The output of NLG is also fed to the speech synthesizer which generates an output in the form of speech.

3.2 Related Work

Research on Voice based Information system focuses on two broad aspect i.e. 1) their social and cultural impact and 2) usability issues with voice interfaces. These are discussed below:

3.2.1 Voice Based Information System: Impact and Potential

Voice based Information System also has great social and cultural impact. Following studies explore the social and cultural impact of voice-based information system.

- Social Media: Avaaj Otalo, an interactive voice application for small-scale farmers in Gujarat, India developed into lively social media which enables open discussion with peers (farmers) [80]. The findings of Avaaj Otalo deployment show the impact of voice-based information system in rural areas. Several other voice-based social media using phone exists like VoiKiosk which is a configurable voice-based portal for information dissemination in developing region. [4]. The use of such voice forums shows the potential to increase the reach and demand of information.
- Citizen Journalism: CGNET Swara, a citizen journalism project lets rural India use the power of user-generated content to amplify their voice [73]. It is a voice-based portal, freely accessible via mobile phone, that allows anyone to report and listen to stories of local interest. Reported stories are moderated by journalists and become available for playback online as well as over the phone.
- Penetrating hidden populations: Sambasivan et al. designed and developed a phone broadcasting system for reaching out to at-risk populations in urban India [87]. The system helped improving outreach of a Non-governmental organization dedicated to assisting Urban Sex Workers (USWs) in Bangalore, India. This work shows interesting use case of voice-based information system in reaching to the hidden population by responding to the unique design constraints of the USW community such as privacy, a desire to remain invisible, and timing constraints. In a similar effort for healthcare, Joshi et al. designed voice based system to reach HIV patient [**?**]
- Good Governance: Voice-based systems also find their use for various governance purposes like resolving local issues and extending the reach of government services. Jharkhand Mobile Radio (JMR) is a platform accessible through any simple phone [40]. It helps to take up the issues with the relevant development authorities to resolve them quickly. Barnard et al. have shown that voice-based system can overcome the barriers to access electronic government services in South Africa [18].

3.2.2 Usability and system design for voice system

Studies in the area of IVR and automated telephone services have proposed various approaches to improve the usability of the system. Perugini et al. [82] discussed three design dimensions for automated telephone services that could be used to study design issues in IVR. These three design dimensions in their conceptual design space were, nature of the user in terms of addressable[4] input (in-turn vs. out-of-turn), input modality (touch vs. text vs. voice), and interaction style (menu-based vs. natural language). In their work, they studied *out-of-turn interaction:* a different nature of user addressable input. Their work reflected user's model of the task while navigating

[4]By addressable information authors mean the information which the system can accept from the user or in other words, the information that the user can supply.

They do not mean information that the system indexes (addresses).

through the menus. They showed that this type of approach is only possible for IVR with speech recognition. However, its use may be limited in developing regions wherein speech technology is not advanced for native languages.

IVR system's input modalities for VUI (Voice User Interfaces) have also been extensively explored. Lee and Lai [64] compared dial interfaces with speech and showed that dial interfaces are preferred for linear task and speech is preferred for the non-linear task. Patel et al. [79] showed that dial interface outperformed speech in terms of task completion rate and learnability. They also showed that dial interface was relatively easier to use than speech. Shrivastava et al. investigated the effects of augmenting IVRs with visuals [94].

Navigation is an essential component of any menu based IVR. Multiple studies have proposed their own methods to improve navigation in menu based IVR. Skip and Scan is one such navigation approach for menu based IVR, wherein caller could easily navigate back and forth through menus without first listening to all of the prompts for a particular menu [85]. Zap and Zoom [46] is another proposed approach for navigation in menu based IVR, that improves over Skip and Scan approach by allowing users to jump directly to a location in IVR using shortcuts. However, Zap and Zoom requires that the caller should be aware of options in IVR menu beforehand otherwise, they will not be able to take the decision of the location they want to jump upon in the IVR menu. HyperSpeech provides an interface for directly addressing from one location to another location in the system [11]. However, their approach can not be extended to key-press based IVR systems. SpeechSkimmer [12], present a system that reduces the time to listen speech recording by time-compressed speech and pause removal. Certain adaptive approaches are tried in audio systems to improve the system performance.

Apart from technological aspects, human factors also impact system design significantly. Grover et al. [92] proposed a dialog design model for low-literate users. They discussed that socio-cultural and domestic environment of users may affect the usability of IVR systems. Some users preferred DTMF for privacy reasons. They also emphasized that techniques like speech recognition and profiling cannot work in cases where mobile sets have multiple ownerships . In a country like India, people generally share mobile phones among family members therefore leading to the common existence of multiple ownership. BlindSight [66], a prototype voice application for mobile phones allows users to achieve eyes free (using the phone without looking at it) error rate below 5%. The experiment revealed that overhead for the eyes-free use of mobile phone keypad is only 200ms per keystroke compared to sighted use. Bernhard Suhm [96] studied cognitive limitations relevant to voice user interface design.

3.2.3 Problems identified

So far, most of the studies towards improving the performance of voice system are confined to proposing different ways of skipping the voice content while navigating in a sequential manner (as in Zap & Zoom [85]). One way to improve over existing navigation techniques is to directly provide the desired information to the user by predicting its information need. Towards this, we propose the use of adaptive interfaces that utilizes data collected from usage of voice system to reduce navigation difficulties.

3.3 Key press based Voice Application for University Admission Process: A motivating scenario

We study use of voice applications for university admission process taking IIIT-Delhi as a use case. IIIT-Delhi (or IIIT-D) is a state university in New Delhi, India. Every year thousands of applicants apply for admission in undergraduate and postgraduate courses. Applicants appear for a written exam based on which a merit list is created. All the relevant information regarding admissions is provided on institute's official website. To resolve any query, applicants are also provided with a telephone number on institute's official website. A call executive from IIIT-D's staff is appointed to answer these queries. Some of the calls go unattended as the call executive is busy in some other work during the office hours of 9 AM to 5 PM. There are also a significant number of unattended calls from people calling in non-office hours. This has provided us an opportunity to deploy a key-pressed IVR that can reduce load on call executive in working hour and also ensure availability of information to callers in non-office hour. We did series of key press based IVR deployment at IIIT-D to understand IVR usage and test our hypothesis.

Key pressed based IVR systems are essentially menu based IVR system where menu options are accessed by pressing keys on the telephonic keypad. Although menu can be accessed through voice command but this requires support from speech recognition technologies that are complex to develop in a setting where multiple users may be using diverse languages and dialects. On the other hand, work on improving navigation in key-press based systems is still in its nascent stage while their use is steadily increasing. Use of key-pressed based IVRs is not limited to researchers or NGOs working in developing countries, but, even the MNCs (Multi-National Companies) operating in developing countries like India use key-press based IVRs only, though they can easily afford a costly voice-recognition software, if one is available. In India, we checked several IVR systems, deployed by big companies e.g. HP, Vodafone, Tata, etc. yet we could not find a single system that used voice recognition to provide interaction. We believe that a similar situation exists for other developing countries where English (or other western languages) is not a native language. Key-press based IVR systems have wider applicability and despite advances in speech recognition, they will continue to dominate IVR deployment (especially in non-English speaking developing regions) for some time to come. Though speech recognition will help in making better IVR systems, given the current progress in this area (especially with respect to non-western languages), it will take a long time to come up with voice recognition softwares that can be deployed in the field.

3.4 Field Deployment

For the academic session 2011-12, the annual intake in an undergraduate course (for which the IVR was developed and deployed) at IIIT-D was of 120 students. This study was conducted for the session 2011, which had 2,211 applicants for the admission. In this section, we present an analysis of usage of our IVR system that was used by applicants (referred as users in the rest of the paper) over 13 days of deployment. Since multitudes of information were accessible through multiple mediums vis-a-vis IVR, web interface and talking to a call executive over the phone, we also present the comparative analysis of information dissemination through multiple mediums.

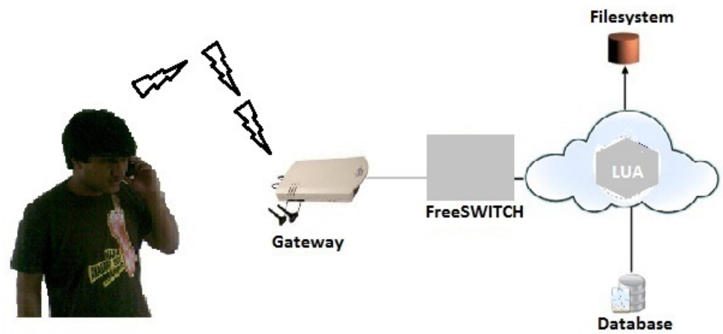


Figure 3.4: *A user accessing information over IVR: user dials the number of IVR on his phone. The call is first received by the gateway and is then forwarded to FreeSWITCH (Telephony platform). An application written in Lua answers the call by playing appropriate voice prompts. Lua application accesses database and file system for storing and retrieving information.*

3.4.1 Design process

The information content for the IIIT-D IVR was prepared with the help of the experienced and proficient staff of the admission department of IIIT-D. To ensure the validity of information content and usability of the system, IIIT-D IVR went through a series of demonstration to the individuals involved in the admission and academic processes. Voice quality and information content were the primary issues as reported during the demonstrations. The initial system was tested with various Text to Speech (TTS) engines for delivering the information content of the system. We tried our system with the voices available in Cepstral, TextAloud, AT&T Natural Voices and Microsoft Voice. Due to unsatisfactory performance of voices available in TTS (both foreign as well as Indian accent voices), we ultimately opted for human recorded voice. We tested our system with 3 humans (2 male and 1 female) voices. After taking inputs from multiple people on different aspects of voice, including clarity, we decided to use a female voice for our deployment. The information content was made short and concise, which also helped in optimizing the time duration for each call. All information presented through IVR was supported with appropriate URLs (announced at the end of voice prompt), to facilitate accessing the specific content in detail on the IIIT-D website.

3.4.2 Implementation

IIIT-D IVR was implemented as a Voice application written in Lua[5] and hosted on FreeSWITCH (an open source telephony platform). Applicants calling to IIIT-D were connected through a gateway as shown in Figure 3.4. Linksys SPA 3102, single line device with the capability to handle one call at a time, was used as the gateway. The gateway converted the analog signal of the telephone line to the Session Initiation Protocol (SIP) and forwards the request to the FreeSWITCH. User input in the form of Dual-tone multi-frequency (DTMF) key press was forwarded by the FreeSWITCH to the application written in Lua.

[5] http://www.lua.org/

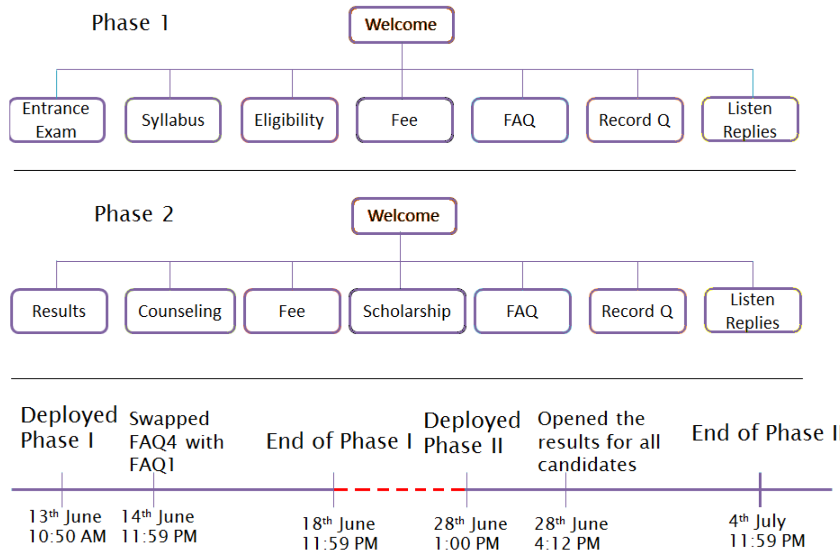


Figure 3.5: *Menu options available in each phase of IVR (1st phase corresponds to Pre-examination and 2nd phase corresponds to Post-examination). The time-line shows the modification done in the system during the experiment.*

3.4.3 Deployment

After deploying IVR in IIIT-D, we conducted the study in two phases for a total of 13 days (6 days for the first phase and 7 days for the second phase). The first phase was Pre-examination phase. In this phase, IVR presented information related to examination and online application process. The second phase was Post-examination phase and it was used to deliver results of the entrance examination along with other information like counseling schedule and fee structure.

The Pre-examination phase was set up to handle calls in the absence of a human attendee. Institute published a telephone number on its official website to resolve the queries of applicants related to admission procedure. For this purpose, a call executive was also appointed to respond to those queries. We integrated our IVR system with the telephone line of the call executive. Callers were not aware of the existence of IVR a priori. The call was transferred to IVR only if it was not attended by the call executive within the first 16 seconds.

Figure 3.5 shows the menu structure of the IVR system deployed at IIIT-D together with the changes we performed in the IVR with time. Pre-examination phase started on 13^{th} June and ended on 18^{th} June.[6] We continuously monitored the call recording as well as interviewed the call executive on a daily basis. Based on the feedback from call executive, who specified that majority of the received queries were related to admit card, we made the first change in the running experiment on 14^{th} night. The 4^{th} item in the FAQ menu that was related to admit card queries was swapped with the 1^{st} item in the FAQ menu. Written exam for the admission process at IIIT-D was conducted on 19^{th} June, that marked the end of the first phase.

The second phase started with results declaration on 28^{th} June. Results were declared, simultaneously on web and IVR, only for the selected candidates. Selected candidates comprised

[6]All times mentioned in this paper are in IST (Indian Standard Time).

of top 247 ranks amongst the total of 2,211 candidates who appeared in the exam. In this phase, we used separate telephone lines for IVR and call executive. There was no option to call the executive and get the results checked by her. Unlike the first phase wherein no specific phone number for IVR was mentioned (and the executive line defaulted to IVR when not picked in time), this time, the telephone number to access information through IVR was explicitly published on the official website of IIIT-D. In the first few hours of the second phase, call executive received a huge number of telephonic queries related to the result of unselected applicants. Due to this, we made a change, both on web and IVR on 28^{th} June at 4:12 PM wherein we declared the results of all the candidates, providing ranks for all the students who appeared in the exam. A separate sheet was uploaded on the official website that contained the top few selected ranks (corresponding to candidates who will potentially be admitted).

3.4.4 Data Collection

We collected data from five sources:

- *Log of system navigation* - We recorded every interaction between the IVR system and the caller. It included DTMF key press along with the time stamp and the voice prompts played by the system.
- *Audio recording of calls* - We performed audio recording for all the calls to capture the callers' interaction with the call executive. Callers were informed at the beginning regarding audio recording of the calls. These recordings were later analyzed to understand the context of the call to understand if any systemic changes are required in the IVR system.
- *Log of Linksys SPA3102* - We collected log of PSTN gateway (Linksys) that includes information on call duration, call forwarding and callerID.
- *Web log* - We logged every web request for results received on our website. It includes the 9-digit application number and the web server time at which request was made.

We also logged various parameter associated with the call. These include a timestamp of DTMF keypresses, hangup cause of calls, duration of the ring before the call was picked up by IVR or call executive, among others. The telephony platform (FreeSWITCH) used by us had 7 different level of logging. Log levels are from most critical to least critical. We enabled the most detailed level of logging in FreeSWITCH that helped us to log all the events. We also collected the detailed log of every SIP packet exchange between PSTN gateway and FreeSWITCH. Detailed log from multiple sources enabled cross-validation of logged parameters.

3.4.5 Study Findings

Pre-examination phase involved comparison of information access on IVR and call executive. Post-examination phase involved comparison of information access over IVR and web interface. In this section, we present important findings from our deployment that helped us to form a new hypothesis. We validate this hypothesis in subsequent user studies conducted by us.

Traffic Overview

In Pre-examination phase, 421 calls were made to the IIIT-D telephone line in a span of 6 days. Thirty-three calls went unattended because neither the call executive nor IVR picked the call as the call executive was busy and the caller disconnected the call within 16 seconds. For the remaining 388 calls, 180 calls were attended by call executive and 206 calls were handled by IVR. Two call recordings were corrupt due to malfunctioning of the system.

Post-examination phase was focused towards delivering the results of applicants and provided other useful information like counseling schedule, fee structure etc. Over the seven days of deployment, 405 calls were made to IVR and 42,420 requests were received over the web.

It was interesting that 42,420 requests were received on the web interface for getting the results for 2,211 applicants. This indicates that on an average people have accessed the result for more than 20 times.

IVR Content and Usage

As shown in Figure 3.5, IVR had seven options in each phase. The first five options contained information specific to the corresponding topic as shown in Figure 3.5. Sixth option allowed the user to record his / her question if the user could not find the sought after information on the IVR. The seventh option, *Listen Replies*, allowed the IVR caller to listen to any response that the call executive may have recorded for the previously asked query in the system. Order of the first five options presented in the IVR menu was based on the discussion with the administrative staff experienced with handling the admission procedure for the past three years. The objective was to present the options in order of their relevance as suggested by the experienced staff. This ordering made us believe that the number of hits (from callers calling into IVR) on each option will be in the order of their position in the menu structure. Table 3.1 shows the number of hits on each menu item in each of the two phases. Contrary to our earlier belief, fifth menu item in Pre-examination phase of IVR had more hits than the second, the third and the fourth option. Similarly fifth option in Post-examination phase has more hits than the third and the fourth item. This shows that in spite of taking inputs from experienced staff, the order of relevance based upon human belief (from prior experience) was different from what was experienced by accurately logging the system usage. It also suggests that it is not easy to predict the correct order of options and it may make sense for the system to learn it based on past usage in an automated manner. In the discussion section, we highlight the importance of having correct order of option.

Option	**1**	**2**	**3**	**4**	**5**	**6**
Phase1	62	17	10	8	32	18
Phase2	284	16	5	4	7	10

Table 3.1: *Number of hits on each option in first and second phase of IVR. Fifth menu item in Pre-examination phase of IVR had more hits than the second, the third and the fourth option.*

In order to get information access pattern, we logged every interaction of the caller with the IVR. We also analyzed the call recordings to know about information received by the caller from the call executive. For the repeated callers, we observed that once the caller gets a specific

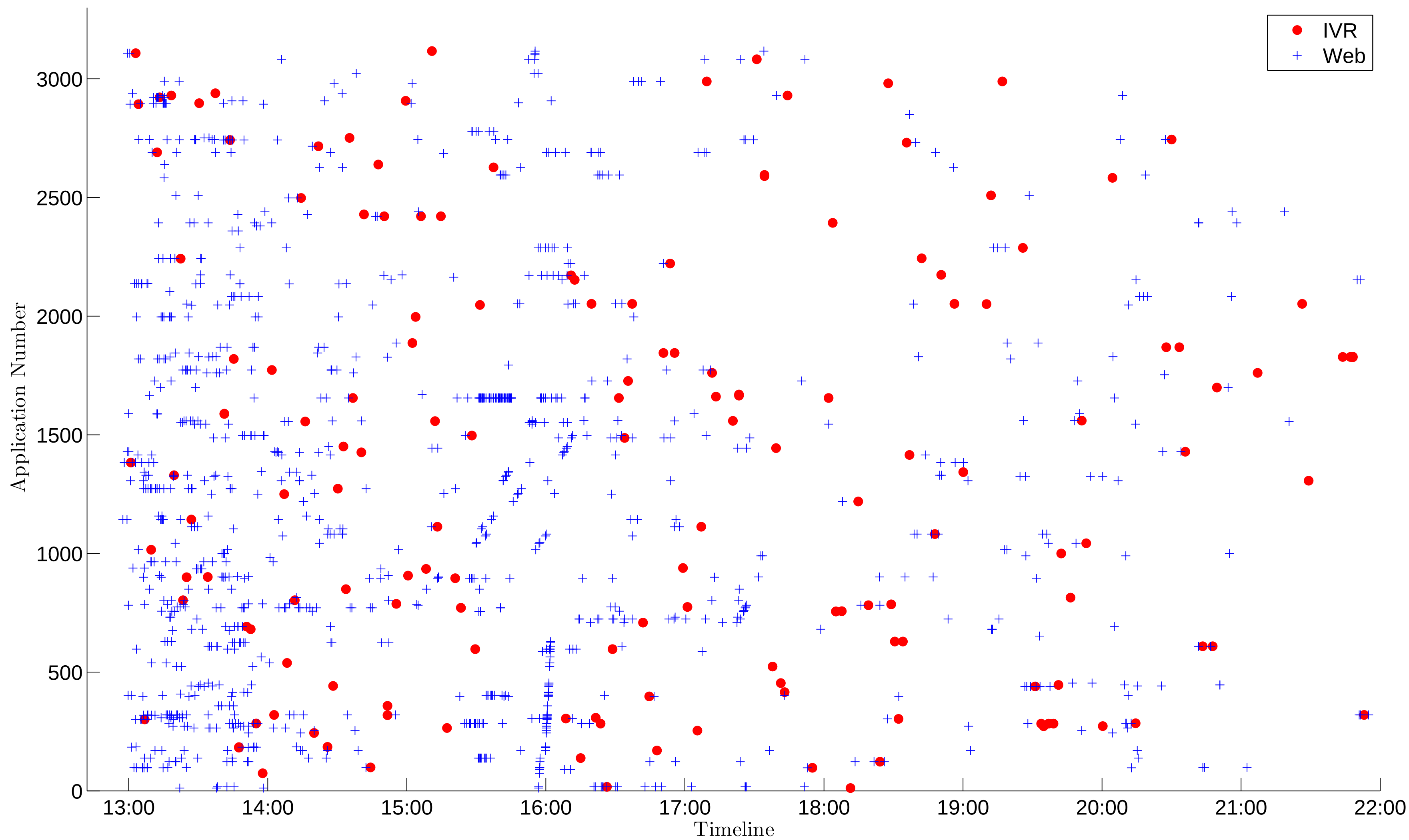


Figure 3.6: *Pattern for access of information on the first day of Post-examination Phase for both web and IVR interface. Y-axis represent anonymized application number. It is evident that there were multiple access by same application number. Multiple access of information for anonymized application number 850 is shown in shaded region.*

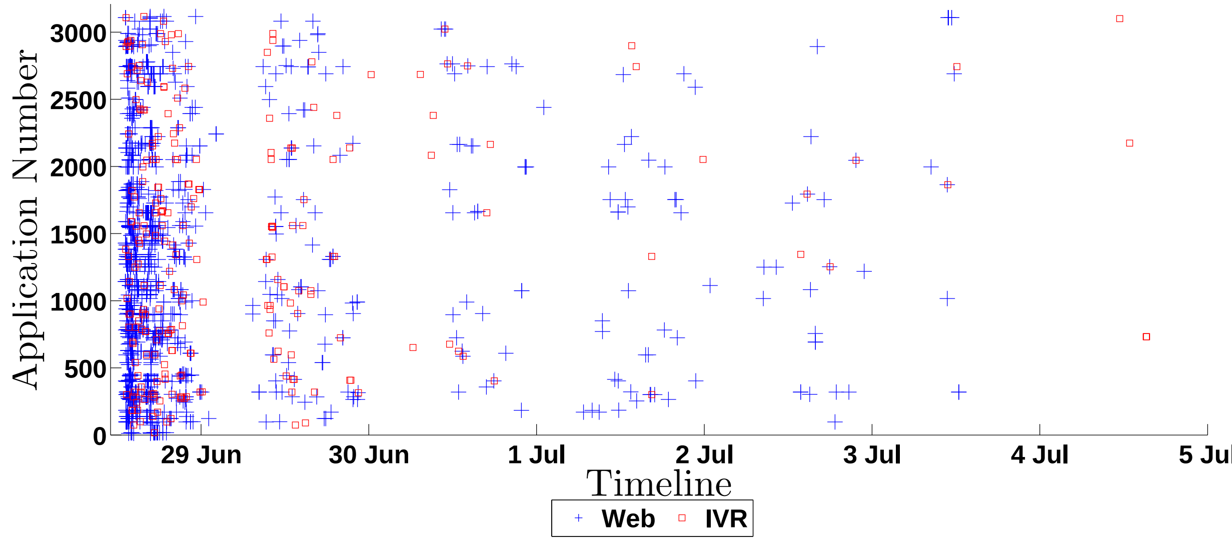


Figure 3.7: *Common result seekers: Applicants who accessed the information both on IVR and web interface.*

piece of information from call executive they did not try to access the same over the IVR in their successive calls. We also observed that callers did not ask the call executive for the information they had already received over the IVR in their previous calls. This leads to an assumption that once the caller was informed about any piece of information through any medium (call executive or IVR), they will not try to access it again or over another medium. Implications of this assumption may impact the relevance of an information for a repeated caller - it may be assumed that once an option has been accessed, it will not be accessed again. However, the analysis of Post-examination phase provides us contrasting outcomes (discussed next).

Post-examination phase was focused towards delivering the results of applicant that can be considered as the most critical piece of information in the admission process. Applicants had the option to access their result both on web and IVR. We logged every request for results. From the logged data, we observed that applicants checked their results multiple times and through multiple interfaces vis-a-vis on web and on IVR. Figure 3.6 shows requests for results made by those applicants who accessed this information on both web and IVR interface on the first day of Post-examination Phase with a time line. Y-axis represent anonymized application number. The data points encircled in the Figure 3.6 shows that an applicant with application number “850” has made 4 web request followed by 1 IVR call and again 3 web requests. In our dataset we found that 189 application numbers were common among queries made by users on IVR and web interface. It was also interesting to observe that there were 8 applicants who checked their result on IVR only.

Figure 3.7 shows the trend among 189 applicant who access their result both on web and IVR. It also shows high amount of traffic on the first day itself. Figure 3.6 displays the detailed information for these 189 applicants for the first day. This data helped us to capture the users behavior towards critical information like results. It shows that people have tendency to cross check critical piece of information like results through multiple source. Based on this contrary behavior from the Pre-examination phase, it may be hypothesized that a critical piece of information (such as examination result) may be accessed multiple times (as well as on multiple mediums, if available) to cross check the validity of information. On the other hand users may just believe the first time, information content that is not very time critical (such as queries related to admission process).

As shown in Figure 3.5, we made a small change in the information content in Post-examination phase on 28^{th} June at 4:12 PM. Initially the results were confined to successful candidates (247 out of the total of 2,211 who appeared) only. Unsuccessful candidates were told as "ineligible for counseling". We observed a different pattern of access among successful and unsuccessful candidates. The successful applicants accessed their result on an average 6.2 times compared to 1.7 times of unsuccessful applicants. This may be because successful applicants took pride in the result or felt happy in accessing the information or shared it with others also. Repeated access of same information does indicate that once such information is identified, it should be handled understanding that it will be accessed repeatedly.

3.4.6 Discussion: Towards Adaptive IVR System

This study was our first step towards understanding the usage of IVR system. We showed that relying on human feedback for predicting the relevance order of information, to decide on the menu structure of IVR is not straight forward. Our results demonstrate that an automated analysis of traffic pattern and adaptation will probably give a better relevance order than relying on human expertise. IVRs have been typically found to be frustrating as they may provide information that is not relevant in the current context of the call, thus requiring higher amount of time to deliver the relevant information. This problem can be addressed by an adaptive IVR that adapts to every caller as well as global pattern of usage, e.g. by rearranging the information based on the order of relevance as learned by the system.

Let us now discuss, in detail, how rearranging the menu items in the order of relevance in an IVR system can help reduce the amount of time spent by the user. Table 3.2 gives the exact number of hits on each of the menu item in our experimental IVR during the Pre-examination phase. If suppose for every menu item, IVR takes c seconds to announce about it (while delivering the complete menu structure) and d seconds to play specific information contained inside the item to the caller, then minimum amount of time required by ordering of option as mentioned in Table 3.2 will be

Option	**1**	**2**	**3**	**4**	**5**	**6**
Hits	62	17	10	8	32	18

Table 3.2: *Number of hits on each option in first phase.*

$$\begin{aligned}62(c+d)+17(2c+d)+10(3c+d)+8(4c+d)\\ +32(5c+d)+18(6c+d)=147d+426c\end{aligned}$$

However, if instead the IVR had realigned the menu items in the decreasing order of their hits, resulting minimum amount of time required will be:

$$
\begin{aligned}
62(c+d) + 32(2c+d) + 18(3c+d) + 17(4c+d) \\
+10(5c+d) + 8(6c+d) = 147d + 346c
\end{aligned}
$$

Therefore, a simple adaptation, of automated rearrangement of the menu items in the order of their relevance, by IVR could possibly lead to significant time saving. Similarly we have observed that very few people listened to the non-critical information option that they had already listened to in their previous encounters with the IVR system. An adaptable IVR can put such information at the end of the menu structure.

We would like to mention that we are not claiming that above presented solutions will result in a better IVR in every situation. Instead, our claim is that an adaptable IVR will prove to be better than present-day IVR systems and adaptation should be an integral part of design in next-generation IVR systems. What the actual adaptation would be will depend on the type of IVR and will vary from one IVR system to another IVR system.

3.5 Study 1: Comparing Design of Adaptive Interfaces for Voice menu

Motivated by observation and findings of previous deployment we designed a user study to test our hypothesis with adaptive IVR system. First hypothesis is adaptive interfaces may bias a user in selecting information other than desired (H1). This is necessary as our research goal is to assist user reach desired information quick rather than biasing it for some other information. Our second hypothesis is that information relevance varies with time (H2). Validating this hypothesis will make adaptive IVR a preferred choice over the static menu systems. Our third hypothesis is restructuring menu such that desired information is presented to user before other information stored in the system reduces selection or navigation time (H3).

In this study, we investigate the performances of adaptive interfaces for touch-tone IVR system to optimize for future callers based on past system usage by other callers. We show that a significant portion of the call duration goes in selecting the correct menu option in IVR. To reduce this, desired menu options by a prospective caller must appear early in the sequence. We show that adaptive approaches to decide the optimal menu structure for future caller outperform static menu based IVR system. We have designed, deployed and evaluated different adaptive schemes for IVR in a real world study.

Next, we describe the design, features, and implementation of IVR system. We also describes the data collection and outlines the system statistics for the usage of IVR systems from a real world deployment, followed by the results and analysis. Based on these findings, we discuss some design implications for an IVR system with improved user experience.

3.5.1 Design and Methodology

With intent to reduce navigation time in IVR, we designed and deployed five adaptive systems along with conventional static system at IIIT-Delhi to conduct a real world experiment. IVR sys-

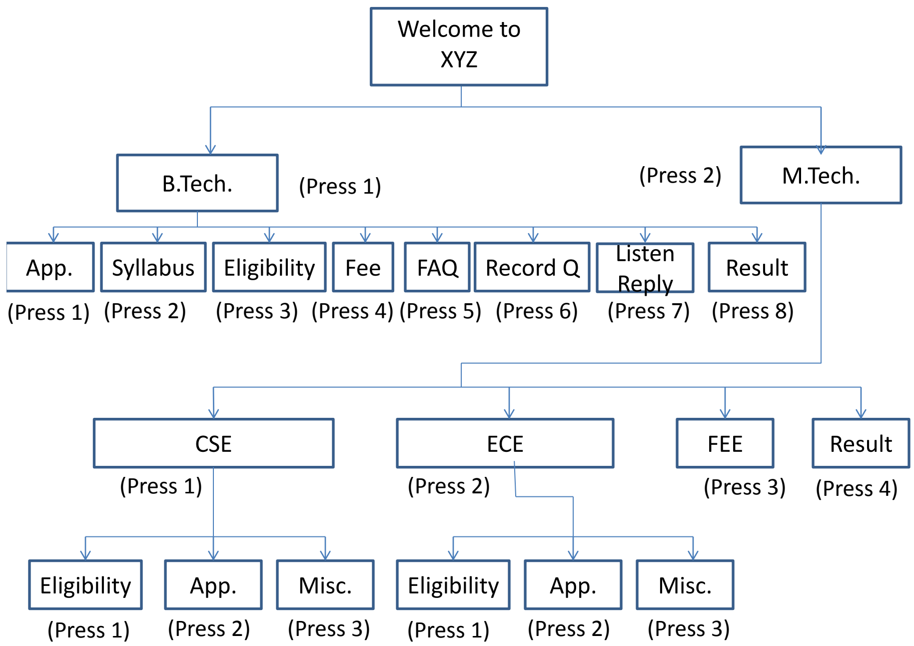


Figure 3.8: *Base-line System menu diagram - the menu options remain same.*

tems served information to the applicants for admissions in the undergraduate and post-graduate courses. For the academic year 2012-13 more than 2,678 applicants applied for the undergraduate and 963 applicants applied for the post-graduate course. Information content to be put on IVR was decided based on the input of academic staff who had experience in handling past admissions. Information related to admission was also available on the official website of the University along with telephone numbers to use IVR facility. Figure 3.8 shows the menu options in IVR system.

All six systems i.e., 5 adaptive and 1 static were deployed in parallel. Adaptive systems were named as Hierarchical-I, Hierarchical-II, Direct-I, Direct-II and Speedy-Voice while the static system was named as Default. Hierarchical-I and II used the same adaptive algorithm known as Hierarchical, however, operate on different historical data to restructure menu options. In Hierarchical algorithm, sibling nodes (menu options) of a parent are rearranged in descending order of the number of times a node has been accessed in the past. Note that, we maintained the relative position of menu items vertically in the hierarchy (see Figure 3.9).

Similarly, Direct-I and II implemented an adaptive algorithm known as Direct, however, operate on different data. In the Direct algorithm, the menu items at the leaf of IVR menu tree gets rearranged in descending order of the number of times a node has been accessed in the past by the users as shown in Figure 3.10. The original IVR structure is not preserved and levels, such as the sub-menus, are replaced by lengthier menu sequence as the system progresses. Although our direct scheme goes against the recommendation of keeping the number of nodes limited to 7. From the usage, we found that a substantial number of users (around 40%) went beyond the 7^{th} option.

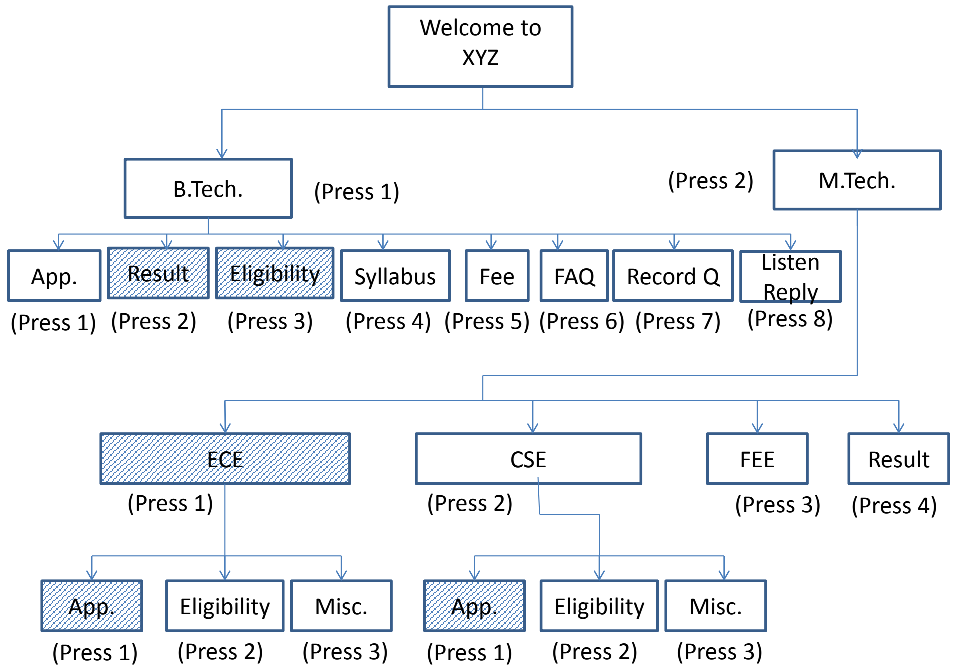


Figure 3.9: *Hierarchical System Menu - the menu options after some usage. The shaded boxes have changed their position.*

Type-I systems i.e. Hierarchical-I and Direct-I use historical data generated by calls made on the default (baseline) system whereas Type-II systems i.e. Hierarchical-II and Direct-II use historical data from two systems i.e. call made on the default system along with calls made on themselves. This helped us to investigate whether apart from adaptive algorithm does data used for adaptation also plays a significant role or not.

The fifth adaptive system Speedy-Voice adjusted the pace of audio content rather than restructuring the menu options. It increased the pace by 5% if user selects an option ahead of its announcement by IVR and decreased the pace by 5% if user does not select any option. We limited the pace adjustment to 0% (1x [7]) to maximum of 65% (1.65x).
To ensure that every system received equal number of calls, we routed each call in round robin fashion. Our system was up from 28th May to 7th July, 2012. In this duration we received 1,120 calls on our system. Enforced by experimental design, each system received equal number of calls as indicated in Table 3.3. Calls were audio recorded and users were informed about this at the beginning of each call.

We also logged the phone number, received as caller-ID information, with each call. Based on phone numbers, we found that system was accessed by 637 callers with 417 single time callers and 220 repeated callers[8]. For 39 calls, our system could not retrieved the caller-ID, hence we are not sure if these 39 calls came from repeated callers or they were also unique callers. For 315

[7] original pace of audio content
[8] who made more than one call

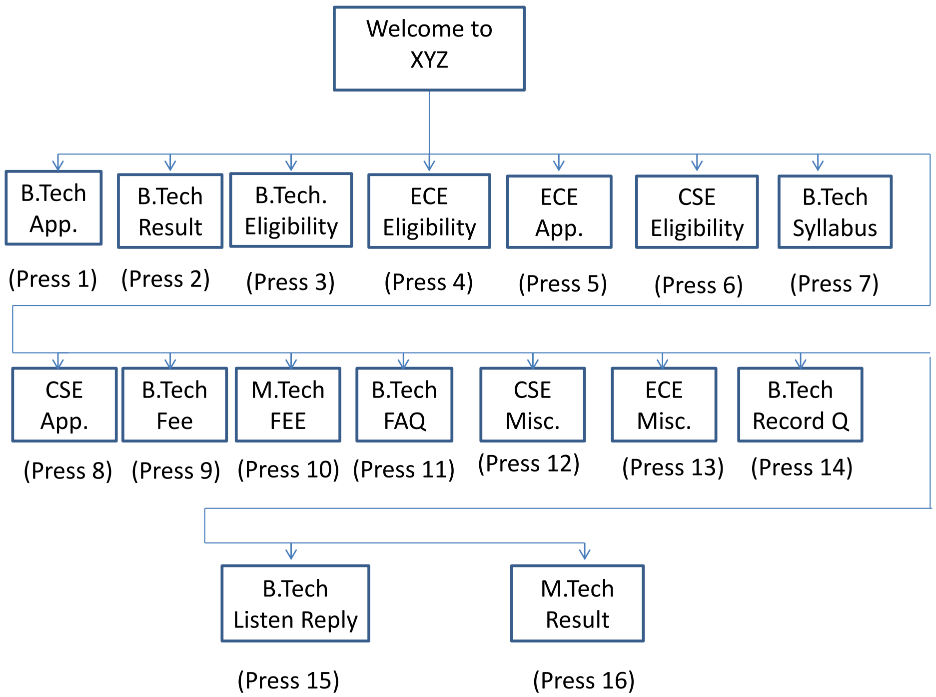


Figure 3.10: *Direct System Menu - the menu options after some usage.*

calls out of 1,120 calls, the caller did not access any information. The reasons for such behaviour could be that the users found IVR menu as confusing or they expected to interact to a human. At the end, we also conducted a post-study among the successful applicants [9], to collect responses for user experiences with our IVR system.

3.5.2 Results and Analysis

In this section, we will show that how restructuring the menu of IVR system will result in better performance of the system. We will measure the performance of the system based on navigation times of the call. To prove that adaptive systems have performed better than Default (base-line) system it is essential to ensure that other experimental parameters were not influenced by it. To ensure this we are also showing that restructuring the menu does not affect users' behavior or the information they seek from the system.

H1: Information access is menu structure dependent.

Prima facie, it seems that menu structure will dictate user's choice in accessing menu options. We were interested in knowing that does restructuring a menu affects the user choices on accessing a particular information. We did an analysis based on content accessed across 6

[9]only the people, who were invited for counseling. Counseling is a part of admission process where a successful applicant gets admission in a course based on merit list.

Table 3.3: Numbers of calls received on each system

System	Number of calls
Base-line	187
Hierarchical-1	187
Direct-1	187
Hierarchical-2	187
Direct-2	186
Speedy-Voice	186
Total	1120

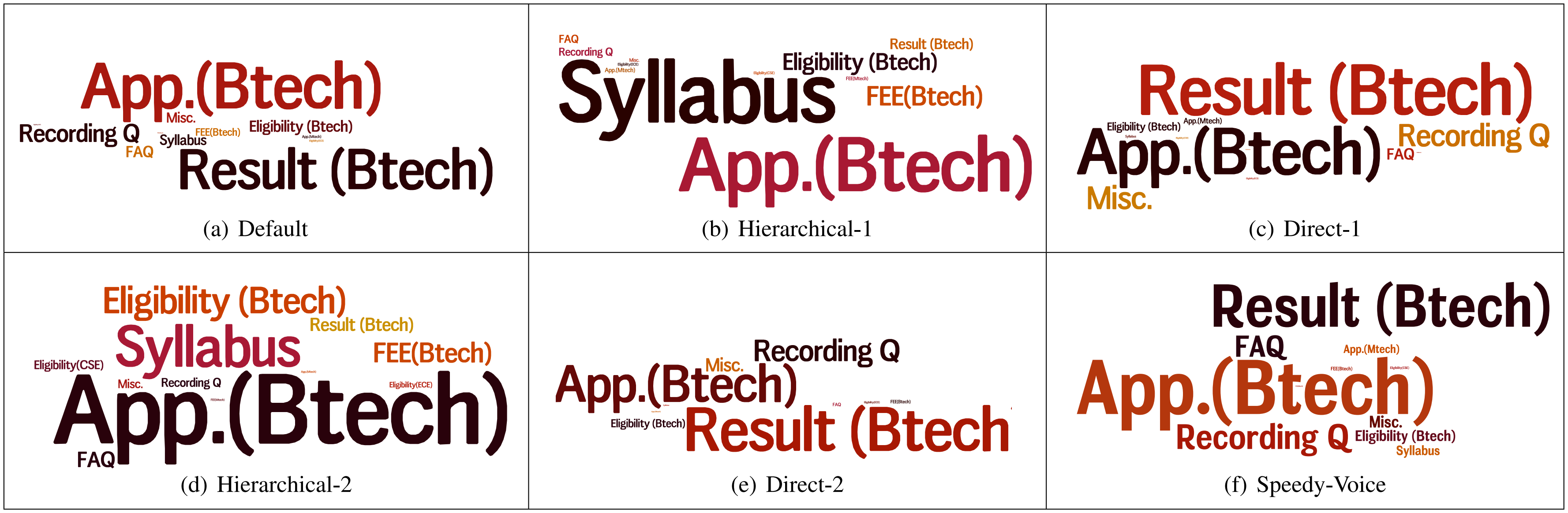
systems and found that relative popularity of menu items were same across all system as shown in Figure 3.11. We also concluded that information access was independent of their position in the menu. As after the result declaration, *result* was 8^{th} option in Default and 2^{nd} Option in adaptive systems (after system adapted in post result declaration phase). But we received high access count on *result* in both the systems. Similarly in the beginning of the deployment, the option *application process* was accessed more than any other option across all the systems. This shows that user behavior and information they seek *does not* depend on menu restructuring. So it is important to study how restructuring affects the navigation time. It is important to see if it leads to better or worse performance.

H2: Information relevance varies with time.

A particular information on the system, relevant at a given time instant, may or may not be so as other times. We have found that the an information, which is popular at a given time instant, is popular on all the system at that time instant. We have seen that external event, e.g. result declaration, drives the relevance of information in the system. B.Tech application process was accessed on all the system on all the time. The number of time *result* was accessed increased in all the system after result declaration. The important implication of this hypothesis is that information relevance varies with time and it also independent of the underlying system.

H3: Restructuring menu, in order of relevance, reduces navigation time

Each call has two-time components i.e. navigation time and access time. Adding navigation time and access time gives total call duration. Navigation time is time taken in listening and traversing through the menu while access time is the time taken in listening to the actual audio content that satisfies user's information need. Access time is information content dependent whereas navigation time is dependent on menu structure. Table 3.4, shows that navigation time is highly dependent on position of option in the menu (see Figure 3.8 for menu). In the baseline system, we can see that as we go farther in the menu sequence, average navigation time increases. Our adaptive schemes reduce navigation time by providing relevant information early in the menu to the caller. Table 3.5, shows statistics for the 805 $(1120 - 315)$ calls where access time is non-zero, i.e., the callers interacted with the IVR system.



(a) Default (b) Hierarchical-1 (c) Direct-1

(d) Hierarchical-2 (e) Direct-2 (f) Speedy-Voice

Figure 3.11: *Information access trend across six system is shown through the tag cloud. Tags represents the menu options in the system. The size of a tag is determined through the relative frequency of corresponding menu option in a particular system.*

Table 3.4: Navigation time statistics for Default system

Option	Option Count	Avg. Nav. Time (in sec)
Btech App. process	49	35.4
Btech Syllabus	12	36.25
Btech eligibility	12	49.58
FAQ	12	50.1
Question	19	69.73
Btech Result	43	70.88

All the adaptive system that work on the menu sequence performed better than the default system. It is evident that even a basic, such as ours, adaptation scheme that works on rearranging the menu sequence out performs the static sequence IVR systems. However, similar basic scheme for Speedy-Voice adaptation did not work in real scenario. To measure the statistical significance of our result we used ANOVA to test the performance difference among the Navigation time of the six IVR systems. Our result shows significant difference among the call Navigation time (ANOVA, $F(5,799) = 5.794$, p-value = 2.89E-05). Table 3.5, shows average Navigation time for the 6 IVR systems.

We also made some interesting observations about callers (repeated caller) who made call more than once. Repeated callers who behaved like first time caller were largely benefited from the adaptation (see Table 3.6), i.e., these callers interacted with IVR system without using any prior knowledge and did not barge-in. Barge In refers to a caller pressing a key (i.e. responding) prior to even listening the menu-option.

Barge-in is enabled in many IVR system to makes them more usable for repeated callers. Next, we observed callers who used their previous experience with the system in their subsequent calls to make menu option selection decision. We wanted to see how adaption affects their interaction. Our system did not restrict any of its users from barge-in. We analyzed repeated callers whose first system was default to clearly differentiate between static and adaptive schemes. Overall, many of the repeated callers, who tried barge-in in their repeated encounter, hang-up the call when they did not find the same option as expected in their last encounter. From such data, we can conclude that, users who tried to use previous knowledge, were affected badly by our adaptive schemes.

3.6 Study 2: Comparing algorithm for Adaptive Interfaces

In this user study, we investigate for algorithmic insights so that performance of underlying algorithm used to build adaptive interfaces, we studied in the previous deployment, can be further improved. This user study was conducted at IIIT-D for the academic year 2013-14, for which more than 2,678 applicants applied for the undergraduate courses that had a total intake of 160 students. For post-graduate courses, 963 applicants applied for the total of 85 number of seats offered by the University. The focus of this study to understand data driven algorithm that will result in fast and correct adaptation of IVR rather than interfaces. Similar to our previous de-

Table 3.5: *Average Navigation time across 6 systems where calls have non zero value for access time.*

Groups	***Count***	***Avg. in sec***
Speedy Voice	126	86.6
Default	131	77.4
Hierarchical1	129	76.9
Direct1	143	62.6
Hierarchical2	140	74.9
Direct2	136	63.2

Table 3.6: *Anonymized data showing repeated caller who benefited from adaptive systems*

Phone Number	**1st call id**	**2nd call id**	**Benefit**	**Quantification of benefit**
9310xxxxxx	811	813	Faster result	59 seconds
9451xxxxxx	745	747	Faster result	26 seconds
9752xxxxxx	649	663	Faster result	5 seconds

ployment these systems provided the information to the user about the admission process. Next, we discuss the basis of choosing the data-driven mechanism for building adaptive interfaces.

3.6.1 Data Driven mechanism

The sequence of menu options in IVR system is defined at the time of deployment and remains fixed throughout its lifetime. A menu option in an IVR system is accessed serially. Thus accessing a menu option far ahead in the menu, takes more time than the options that come early in the sequence. One method to decide the order in which menu should appear in the sequence is based upon the order in which they are most likely to be accessed, i.e., the most likely option will be the first option, second most likely will be the second option and so on [86]. However, the relevance of a particular menu option may vary with time and cannot be fixed prior at the time of deployment. For instance, for an academic institute's IVR system the *"Result"* option will be the most likely option to be chosen by the caller after result declaration but not before that or in the foggy winter the most likely option in Railway's IVR system could be *"Delayed trains"* whereas in peak summer for ticket booking *"Air-Condition train"* option may be accessed more than *"Non-AC Train"*. In several scenarios, the external factor (winter, summer, results) affecting the choice of menu option selection may not be predicted manually. However, the data-driven approach can detect this trend from IVR system past usage and predict the most likely menu option for upcoming callers of the system. With the right prediction callers can be served with desired menu option much early in the sequence than fixed menu. In this experiment, we will study different design parameters of data-driven approaches affecting the prediction of menu options.

The first variation that we study is the volume of past system usage data on which algorithm should operate. Using very old data may be irrelevant in predicting new trends whereas discarding too much old data may result in inaccurate prediction. Thus based on the volume of data being used by the algorithm, we categorize our algorithms as full data or partial data algorithm. Next variation in each type of algorithm that we study is about assigning appropriate weights to different data points in predicting the new trend. Along these variations, we propose four adaptive algorithms. As it has been mentioned for other adaptive system [56], we would also like to emphasize that 1) adaptation may (and usually does) employ a multitude of techniques, 2) we do not present a single correct approach, instead present a design space with various trade-offs.

3.6.2 System Design

This study explores the data-driven solutions targeted to the problem arising due to fixed structure of IVR menu. We believe that the study findings will also be applicable to menu based IVR systems accepting user input through voice command but not to the voice-based system using natural language processing or other similar techniques. We are also targeting systems that are deployed for mass usage and which can not have personal information of the user, e.g. railway inquiry, national government schemes, i.e. users are not identified to the system by their unique pins or login id, etc. The lack of any information about users makes our task even more challenging.

Next, we describe the proposed algorithms and the adaptive systems implementing them.

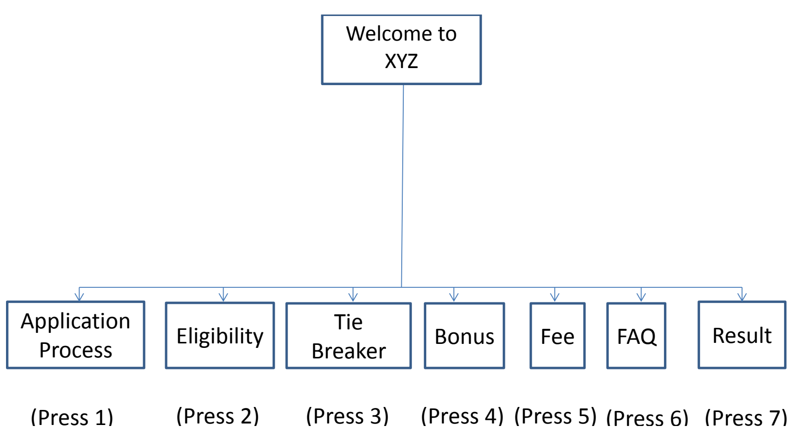


Figure 3.12: *Base-line System menu diagram - the menu options remain same.*

Adaptive Algorithms

We used the existing touchtone-based IVR system of IIITas baseline IVR system, against which we will compare our algorithms (see Figure 3.12). The baseline system is same as the currently

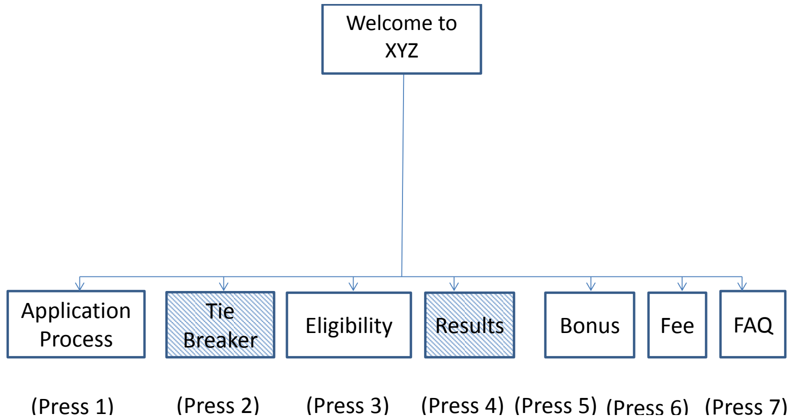


Figure 3.13: *Adaptive System Menu Diagram - the menu options after some usage. The shaded boxes have changed their position.*

available IVR systems in the industry, customer care services, etc. We believe that the history of calls reflects information needs of the users; hence, we use this to build a usable data driven adaptive IVR system. Each adaptive IVR system relies on the power of predicting the correct IVR menu configuration according to its algorithm. In correct menu configuration, the menu options are arranged in the order which is most likely to be accessed by future callers to effectively reduce the navigation time of the call. Our proposed adaptive systems for menu restructuring are as follows:

- **Full data algorithms:** These algorithms are based on the belief that the power of prediction in data-driven algorithm lies in the amount of data used for prediction. We have proposed two full data algorithms:
    1. *Simple Count (SC):* A naive algorithm, similar to Asthana et al. [15], which processes all the historical call data and counts the total number of each menu option accessed in the past. For the new caller, the options are arranged in decreasing order of their total count. For instance, Figure 3.12 shows the initial menu configuration which is changed to menu configuration shown in Figure 3.13 as the options *Tie-Breaker* and *Results* are accessed more than *Eligibility* and *Bonus* respectively. This Simple Count (SC) algorithm is used in 3 adaptive systems explained in Section 4.3.
    2. *Time weighted Count (TC):* An algorithm which processes all the historical call data but puts different weight to each call. The most recent call gets a numerical weight of 1, and the least recent call (first call in history) gets a weight of 1/N where N is the total number of calls in history. Weights of calls between most recent and least recent are equally spaced between the numerical value of 1/N and 1. The sum of the weighted count is arranged in decreasing order and it is the order of new menu options in future IVR menu configuration.
- **Partial data algorithms:** These algorithms are based on the belief that an accurate prediction of future relies more on recent history and not the full historical data. In fact, the full historical data can result in the wrong prediction. However, an important requirement is to know what amount of data qualifies as recent history while correctly predicting infor-

mation need of immediate future callers. We have proposed two partial data algorithm as follows:

1. *K-Frequency (KF):* An algorithm similar to the simple count (Frequency) but operates on recent K calls in the history. For each new call, K is known prior to the call which is used for menu rearrangement. The menu options are rearranged based on the simple count in recent K calls. After the end of each new call, K is updated. The new value of K is decided upon the menu options accessed in current calls. The algorithm tries to find the minimum value of K such that, in K recent calls, simple count algorithm gives the highest priority (maximum count) to the option accessed in the last call (most recent call), and all the menu options have count > 0 (i.e. each menu option is accessed at least once in K recent calls). Suppose a caller chooses Menu option 2 (out of 7 options in the menu) in the current call. Then K is the smallest integer such that, in last K call, the frequency(or count) of Menu option 2 is highest among the other option. The value of K is initially set to the total number of calls in history. On each new call, value of K is updated as described above.
2. *Recent item first (RF):* This algorithm calculates new menu configuration based on recency of the menu options accessed in the past calls. To calculate the new priority, algorithm chooses the menu option accessed in the last call. The option accessed in the last call is promoted to the first slot in the menu sequence and all other options shifted later in the sequence as it is they were presented to the last caller.

   As recency of a menu option is not affected by all the past calls in the history, so it is inherently a partial data algorithm. The algorithm is also inherently biased to most recent call as the option accessed in the last call is the most recent option.

Adaptive Systems

The adaptive algorithm discussed in the previous sub-section are used by one or more adaptive systems in our study. We have designed 6 adaptive systems - 3 different variants of the system which use Simple count (SC) adaptive scheme and one system each for TC, KF and RF algorithm. The details of each system are as follows:

1. *Simple adaptive (SA):* It uses adaptive scheme Simple Count (SC) and rearranges menu according to it. This menu rearrangement changes the numeric key associated with the menu option i.e. earlier option A was accessed by pressing 1 on the keypad of mobile phone, but after rearrangement, it may change to access it by pressing 3 instead of 1.
2. *Simple adaptive static keys (SAS):* It uses adaptive scheme SC and rearranges menu according to it. This menu rearrangement changes the position of the menu options announced in the menu, but not the numerical key associated with the menu option. For example, if the key associated with the option A was 1 then in all the rearrangements irrespective of when the option A is announced, the key to access option A remains 1. User may feel that announcement of menu options does not follow conventional IVR announcement where options are announced in the increasing order of the value of the key associated with options.

3. *Simple adaptive static keys with the loud announcement (SASL):* The adaptive system is similar to SAS. The only change that a user will observe on this system in comparison to SAS is loudness of voice announcing the value of the key. For example, while announcing the following message the text given in bold is announced louder than the remaining text. *"To know more about application procedure **PRESS 1**"*. This loudness is to emphasize the caller that options are not organized as conventional IVR system i.e. in increasing order of keys associated with them.
4. *Time based adaptive system (TAS):* It uses Time-weighted Count (TC) as an adaptive scheme to rearrange the menu. The keys associated with each option changes after re-arrangement and the menu options are announced in increasing order of key associated with them.
5. *Frequency based adaptive system (FAS):* It uses KF as an adaptive scheme to rearrange the menu. The key associated with each option changes after rearrangement and the menu options are announced in increasing order of key associated with them.
6. *Recency based adaptive system (RAS) :* It uses RF as an adaptive scheme to rearrange the menu. The key associated with each option changes after rearrangement and the menu options are announced in increasing order of key associated with them.

Experiment Design

We conducted experiments using between subject study designs. Each caller is assumed to have a unique caller ID. For each incoming call to our system, we assign it to one of the 7 IVR systems (1 baseline and 6 adaptive systems) to handle the call. Since, we have 7 systems in total, we designed to route calls coming from the new number to these 7 systems in the round-robin fashion. For each call coming from the new number (i.e. not previously assigned any system), the system generates a numerical call ID in an incremental fashion. This call ID is operated under modulo 7 arithmetic to decide the system to handle this call. Table 3.7 shows system assignment based on the output of the modulo 7 on call id. Repeated calls from the same number get the system assigned to them in the first call from that number.

Table 3.7: *Assignment of system based modulo 7 operation in the system generated call id.*

Output(=Call ID % 7)	System
1	Baseline (BL)
2	SA
3	SAS
4	SASL
5	TAS
6	FAS
0	RAS

We followed a rigorous approach to ensure the correctness and completeness of data for post-experimental analysis. We collected data from three sources discussed below.

- *Log of system navigation* - We recorded every interaction between the IVR system and the caller. It included DTMF key presses along with the time stamp and the voice prompts played by the system.
- *Log of Linksys SPA3102* - We collected log of PSTN[10] gateway (Linksys) that included information on call duration, call forwarding and caller ID.

We also logged various parameters associated with a call. These include time-stamp of DTMF keypresses, hangup cause of calls, duration of the ring before the call was picked up by IVR, etc. The telephony platform (FreeSWITCH) used in this study had 7 different levels of logging, from most detailed to least detailed. We announced our callers that calls are recorded and data will be used for research purposes. We were not required to go through an Institutional Review Board (IRB) approval process before conducting the study. However, the authors of this paper have previously been involved in studies with the IRB approvals and have applied similar practices in this study.

3.6.3 System Usage

We received a total of 692 calls on our system in a duration of 15 days. Based on experiment design each system received number of calls as indicated in Table 3.8.

Table 3.8: *Numbers of calls and unique callers received on each system*

System	**Number of calls**	**Number of unique Callers**
Base-line (BL)	69	46
SA	99	57
SAS	123	51
SASL	103	53
TAS	74	44
FAS	111	55
RAS	113	57
Total	**692**	**363**

The number of calls received by each system is different because of the different number of repeated callers on each system. We also logged the phone number, received as caller-ID information, with each call. For 23 calls, our system could not retrieve the caller id (i.e. the phone number), hence, we are not sure if these 23 calls came from repeated callers or they were also unique callers. After removing the 23 unidentified calls and repeated calls from our dataset, we have 363 unique callers as shown in Table 3.8.

[10]Public Switched Telephone Network

3.6.4 Results and Analysis

In this section, we will study the performance of different adaptive systems and the underlying adaptive algorithm. Each and every adaptive system was designed to test various aspects of designing adaptive system and algorithms. Prediction and presentation are two important directions that need to be answered in building a more usable system. An incorrect prediction may even result in a worse experience than with a static system. Prediction requires a clear understanding of data and patterns that are occurring in accessing data. The important questions are to decide the amount of data (full or partial) and data attributes (Frequency or recency) which can be used for accurate prediction. Another important factor in any usable system is the presentation or interface of the system itself. We will explore performance difference due to the presentation (static keys or loud announcement of associated keys) by comparing system performance with each other when they have same prediction capabilities but a different presentation (Simple Adaptive systems, i.e., SA, SAS and SASL).

Information Access

Our adaptive systems rely on the fact that users of IVR systems have a common information need which changes from time to time and this trend can be predicted from the data.
The information need of the caller is governed by factors external to the IVR system. Hence, information access across different systems should reflect this information need and should have the same trend. In other words, the adaptive systems should not affect the choice of the caller in accessing the information. Figure 3.14 shows information access pattern across different systems. We show the name of menu options and the relative size of the word to show the relative frequency of accessing different information.

In all the systems, the menu option "Results" can be seen as most popular and menu option "Tiebreaker" is the least popular. Almost the similar pattern can be seen with other 6 menu options. This shows that different systems do not affect the type of information chosen by callers and that their information need is governed by factors external to the system and hence can be seen consistent across all the system.

Utilization factor

The goal of our IVR system is to reduce the menu navigation time so that the caller can spend more time listening to meaningful information required from the system rather than listening to system prompts for menu options announcement. A caller can listen to multiple information in a single call. Therefore, judging system purely on low navigation time or high access time may not be an accurate metric as these values (navigation and access time) will be high in the calls with multiple information accesses. If the user listens to multiple information then the corresponding call duration will also be lengthier than normal call in a proportion approximately equal to the number of information items caller accessed in the call. Hence, a high access time but normalized by corresponding call duration represents the proportion of the call duration went for listening to the meaningful information. A comparison of such proportion across the different system is a good metric to assess the system performance. We define Utilization factor as a metric to

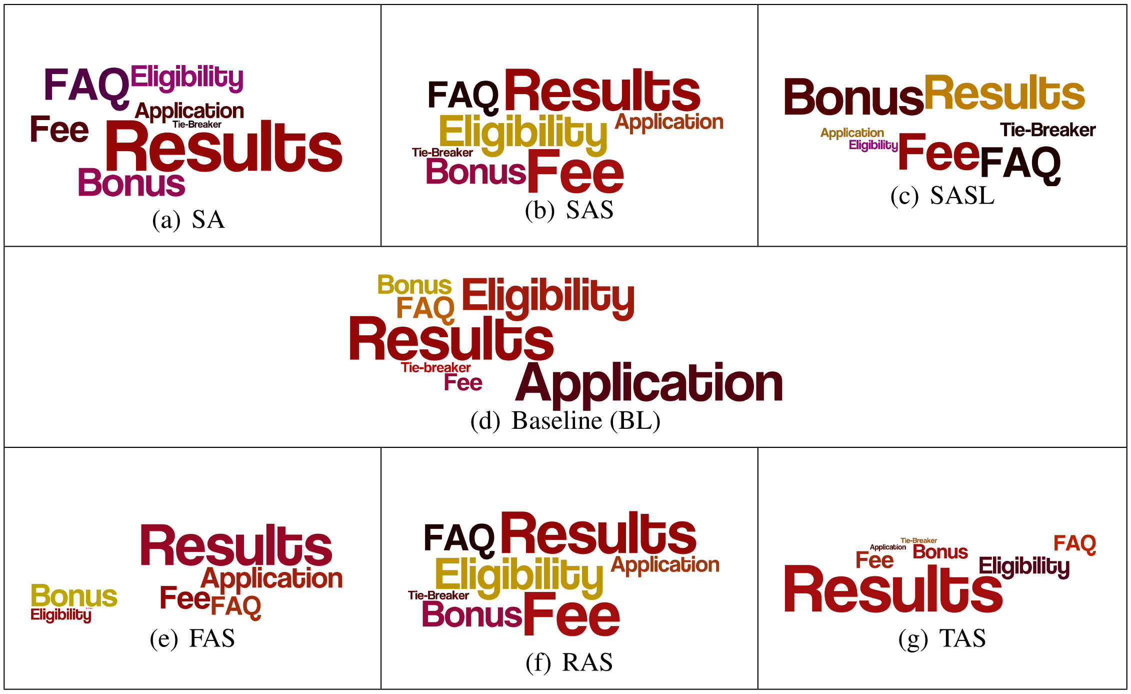


Figure 3.14: *Information access trend across six systems is shown through the tag cloud. A tag or a word represents the menu options in the system. The size of a tag is determined by the relative frequency of corresponding menu option in a particular system.*

measure the system performance in achieving this objective as follows:

$$Utilization factor(UF) = \frac{Access\ Time}{Call\ Duration}$$

UF varies between 0 and 1 with a higher value indicating a better system. In an ideal situation, the navigation time should be 0, thus making the access time equal to call duration i.e. making utilization factor equivalent to 1. We calculate the utilization factor for each call.

We separated the first call and repeated calls from the same number (Caller_ID) and compared the system based on utilization factor of the first call so that any bias due to system learnability should not affect the system performance. We applied the non-parametric (Kruskal-Wallis) test to study the statistical difference among the system performance. The mean ranks and test statistics for utilization factors are reported in Table 3.9

Arranging the system based on decreasing order of mean rank shows that the performance of $TAS > FAS > SAS > RAS > BL > SASL > SA$. However, the test statistics shows that utilization factor does not have statistical significance (p>0.05). In spite of KruskalWallis not reporting any significance difference, we performed posthoc test (U-test, Mann-Whitney) to check statistical significance of each system with Baseline (BL) system. The results are reported in Table 3.10.

The results show that the performance of TAS is significantly different ($p < 0.05$) from BL. TAS is based on full data algorithm and utilizes both the attributes of the date element i.e., frequency and recency. Based on these results we will try to infer the performance characteristics

Table 3.9: *Utilization Factor mean ranks and test statistics. N represents the number of unique call in each system.*

SystemType	N	Mean Rank
BL	46	174.858
SA	57	167.342
FAS	55	187.509
RAS	57	177.184
TAS	44	220.795
SASL	53	172.792
SAS	51	180.362
H=9.06, df=6, p-value=0.16		

Table 3.10: *Test statistics for Mann-Whitney for pair wise comparison with Base-Line (BL) System.*

System	Z-score	p-value
BL	NA	NA
SA (U=1275.5)	0.254	0.799
FAS (U=1177)	-0.632	0.526
RAS (U=1283.5)	-0.197	0.843
TAS* (U=843)	-1.963	**0.049**
SASL (U=1197)	0.171	0.863
SAS (U=1138.5)	-0.274	0.784

of the adaptive algorithm in the next section.

Impact of Data Ageing

Table 3.10 shows that only TAS has statistical significance for utilization factor when compared to BaseLine (BL). Table 3.9 shows that mean ranks of SAS, FAS and RAS have performed better than BL, although they do not have statistical significance as shown in Table 3.10. An interesting observation is that the system SA and SASL have performed even worse than BL i.e. mean ranks of SA (167.34) and SASL (172.79) are less than BL (174.85). The differences of BL with SA and SASL are not statistically significant ($p > 0.05$). In lack of statistical significance, we can not claim that SA and SASL will always perform worse than non-adaptive system. However, with this data we can easily assume that there exists a good possibility that the adaptive system may not improve the performance and can deteriorate the performance below the non-adaptive system. A careful inspection on the adaptive algorithm may reveal the reasons behind the performance difference among the systems.

In the decreasing order of mean rank, the performance of IVR systems is arranged as $TAS > FAS > SAS > RAS > BL > SASL > SA$. This performance sequence reveals several things. One of them is the incorrect prediction of the menu sequence due to old data. The algorithm TAS, FAS, and RAS give more weight to recent data. In the lack of time-based weighing, the present information need of the caller may be suppressed by past information need of the caller depending on call volume. We call it a *data ageing* problem where the system is not able to detect current trend because the current trend is dominated by high data volume of past trend.

The Simple Count algorithm which is used in SA, SAS, SASL is susceptible to the ageing problem of data because they give equal weight to recent and old data. The current trend detected by these systems may actually be an old trend. By the time system detects the current trend, the current trend may have changed. In such situation where series of prediction goes wrong by the system, it may decrease the performance as well as shown by SA and SASL. In the case of SAS (also based on SC algorithm), we have observed better performance but results lack any statistical significance.

The only exception in above-observed performance sequence is the performance of SAS$>$ RAS. The SC algorithm used in the SAS is most susceptible to data ageing among the 4 studied adaptive algorithm and RF algorithm used in RAS is least susceptible to data ageing. We found that the number of calls used to predict menu configuration ranges from 5 to 83 (Mean = 31.4, SD = 17.62). These statistics show that RAS had lost valuable information by rejecting a huge number of calls that should have been considered for prediction. In spite of RAS not having any data-ageing problem, it shows that loss of information for trend detection has resulted in several incorrect predictions.

Impact of Data elements attributes

We considered two attributes of data elements, i.e., frequency and recency. Adaptive systems were based on either one of the two attributes or a mix of both. System SA is pure frequency based system and operates as a full data algorithm, whereas FAS is a frequency based system operating on partial data algorithm. FAS uses some recent K calls and rejects the old call data information. Thus, FAS also gets some properties of recency based system. RAS is a pure recency based system and its predictions simply ignore all the frequency based information in the recent calls. The TAS is a mix of both where the frequency of data elements is adjusted according to their recency. Moreover, TAS is a full data algorithm.

We already know that TAS has outperformed all other system and its performance as compared to BL was statistically different (p-value$<$0.05, see Table 3.10 for comparison with BL). Thus, it indicates that a full data algorithm which considers both the attributes (frequency and recency) is better than any other algorithm which is either partial data or considers only one data attribute. In the same direction, if we further analyze the system, the performance of FAS is measured as second best after TAS ($TAS > FAS > SAS > RAS > BL > SASL > SA$, see Table 3.9 for mean ranks). Although performance difference is not statistically significant (p$>$0.05, see Table 3.10 for comparison with BL), however, it suggests that a partial data algorithm system like FAS, which uses a frequency-based algorithm, has inherently attained some benefit of recency as the partial data considered for processing the recent data. The performance of Frequency and recency based system (FAS) on partial data has outperformed any other frequency based full data algorithm (SA, SAS, and SASL) or purely recency based system (RAS). Thus, the performance of TAS and FAS based on mean rank indicates that **a combination of frequency and recency is important in building better adaptive algorithm**.

Frequency and recency based systems have their own pros and cons. Both types of systems are trying to guess the current trend of information access so that popular information can be positioned on top. A Frequency based system is robust to small fluctuations in the trend which can be noise. However, this robustness can become a reason for late detection of the trend leading to incorrect prediction. Thus, a frequency based system may be suitable in the context where trending information (or popular information) is stable over a period of time, and when the trend changes frequency based adaptive system must get enough time (or data volume) that it can adapt before the trend changes again. In such context, the system will not be prone to the problem of the data ageing. In contrast to frequency based, the recency based algorithms are highly sensitive to fluctuation in trend. This solves the problem of the data ageing very well but can be misleading when the fluctuation in trend is noise and not the actual change of trend. We have shown earlier that RAS suffered from the loss of information due to recency and detected trends were an inaccurate prediction. Thus based on the performance of TAS we conclude that a full data algorithm considering both frequency and recency is better than other algorithms and robust to many contexts. Other adaptive algorithms may perform better than the non-adaptive system but require favourable conditions.

Information Presentation

We designed three systems (SA, SAS, SASL) all with same data algorithm, but differing in information presentation. In our experiment, there were two ways to present a rearranged menu. One way was to reassign the keys to each menu option after the rearrangement and another way was to keep fixed keys associated with a menu option. The second approach can sound awkward to the first time caller but can be good to the habitual callers who frequently access the system and have a tendency to barge-in (i.e. pressing the key - based on past usage of the system - before even listening to the menu option). The system SA presented the information like a conventional system where menu options are announced in order of the key associated with the option. However, the keys association with a menu option in SAS and SASL were fixed, so even after rearranging the menu, the order of menu announcement changes but the key press to select the option remains same. This may lead to an awkward situation where, for example, the first option is announced to be selected by pressing 7 followed by announcement of the second option to be selected by pressing 1 i.e., "For Results press 7..For Application press 1".

We tried to analyze the effect of the change in information presentation on the system performance. We performed a non-parametric statistical test (Mann-Whiteny) for pair-wise comparison of SAS, SASL and SA. The results are reported in Table 3.11, and as we can see the p-values

Table 3.11: *U-test (Mann-Whiteny) statistics for pair-wise comparison of SA, SAS and SASL*

System-I	System-II	U	Z-Score	P-value
SA	SASL	1498	0.082	0.934
SA	SAS	1384.5	0.461	0.644
SASL	SAS	1306	0.331	0.740

do not show any statistical difference between the system. We further tried to investigate caller who faced difficulty in accessing information through our IVR system due to presentation issues.

In Table 3.12, we show the number of callers on each system who were not able to access the information in their first attempt and in their second attempt respectively.

Table 3.12: *Repeated callers who were not able to access information (access time = 0) from IVR in their first and second attempt*

System	First Attempt	Second attempt
SA	15	7
SASL	17	9
SAS	18	12

These number shows the callers who faced difficulty in each system. The callers who were confused in their first attempt is very small to perform any statistical test. However, a rough estimate shows that the performance of SA and SASL are almost similar in the confused caller.

We initially hypothesized that system with static key (SAS, SASL) will create confusion among the caller as menu options are not presented in increasing order key value. A high number of confused callers on the static key system in comparison to SA (see Table 3.12) also indicates this. We kept the key announcement loud in SASL so that the correct key can be emphasized in the menu announcement. We feel that the effect of this loudness has contributed to the better performance of SASL in comparison to SAS in the second attempt (number of confused callers are less in SASL, see Table 3.12). However, in the lack of substantial data, we can not make any statistical claim and need to be further investigated in future studies.

3.6.5 Discussion

Adaptive interfaces based on data driven mechanism showed promising results towards reducing navigation time in menu based IVR system. Our findings applies to system where information relevance varies with time. We presented adaptive rearrangement of menu based IVR systems that are key-press based systems. However, the findings are equally applicable to menu based systems accepting user input through voice command but not to the system using natural language.

The advantage of our approach is that none of our technique require user profiling and thus works on a new caller. However, its continuously changing menu can be confusing for repeated caller. A stable mental model about the system that repeated caller has in subsequent interaction may create an adverse situation [16]. To overcome such problem, we feel that system like static keys (SAS or SASL) can be a good solution but before that, they also need to be tested for the negative impact on new callers. Alternatively for repeated callers, one can extend the adaptive system with user profiling technique where the system freezes the menu configuration for each caller and does not change in subsequent interactions. Thus, a hybrid approach may suit better.

By and large, we propose that with the current findings, the adaptive system can be used to replace static menu based IVR system where an organization expects a large number of new callers and few repeated callers, otherwise a hybrid approach needs to be taken. We have tested our system in urban settings, and we hope that system will work in rural settings because menu rearrangement is completely invisible to its first-time user, and the user can not differentiate between whether it is a static system or adaptive system. Thus, we believe as the static system works with the rural population, the adaptive system should also work. We believe that adaptive algorithms have great potential and a right combination will help improve the usability. Towards this, our work provides good insights in selecting relevant parameters for adaptation.

Chapter 4

Maareech: Usability Testing Tool for Voice Response System using XML based User Models

Due to the widespread deployment of IVR systems, it has become imperative to test the IVR systems before their deployment. A software developer can test the system functionality, but not the interaction issues faced by its users. The solution to this problem could be to conduct a usability test with few participants through the lab studies, but these lab studies are time-consuming, costly and difficult to perform on a large scale. In this chapter, we present Maareech[1],

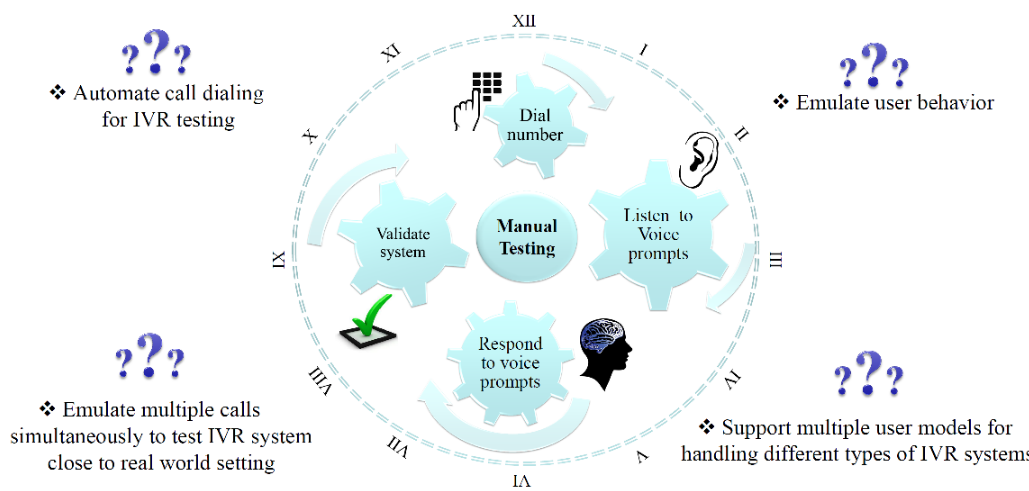


Figure 4.1: *Steps of IVR testing that are automated by Maareech in call emulation*

a tool that can mimic user behavior to simulate large-scale user testing of a system. Maareech is a complete and robust testing tool to test IVR systems. Maareech has the ability to dial a phone

[1]Maareech is a daemon in Hindu Epic Ramayana who could assume any form

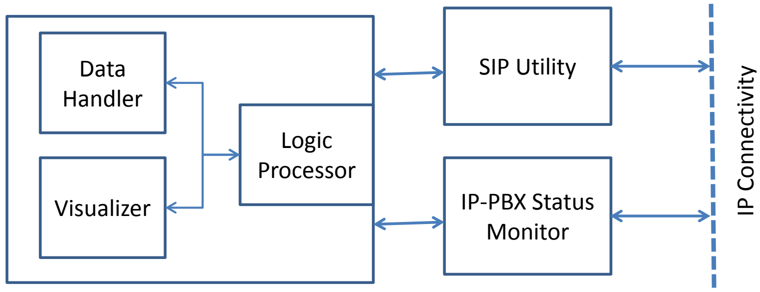


Figure 4.2: *Maareech's Architecture: It has modular architecture with five components*

number, listen to a voice prompt, enter the required DTMF[2] or recorded speech input for testing different types of IVR applications (see Figure 4.1). Maareech can also use data generated from real world calls for call emulation. Mimicking user behavior provides the ability to optimize and evaluate the performance of IVR applications. Maareech is built to help the HCI researchers conducting usability tests. It has the capability to incorporate new user models based on which an HCI researcher can test different system user interactions.

4.1 Architecture

Maareech has a modular architecture with five basic modules as shown in Figure 4.2. It is written in JAVA and designed using the MVC (i.e. Model, View, Controller) pattern. This makes Maareech highly customizable and extendable to different scenarios. In this section, we describe the architectural components of Maareech.

Logic Processor: This module is responsible for making all the decisions for the emulation process, e.g., call initiation, event regeneration, etc. It coordinates with other modules to perform call emulations. The Logic processor interprets the underlying user modules. New user models can be created under the Logic Processor to emulate user behavior for different scenarios. It can schedule events and maintain call queues[3].

Visualizer: This module provides the user interface of Maareech. It is responsible for taking inputs from the user for different emulations and showing the output to the user (see Figure 4.2). The Visualizer component is responsible for showing calls loaded in the call list, events in each call, and the current operation performed by Maareech in the test.

SIP utility: This module is responsible for performing all SIP (Session Initiation Protocol) based communication with the IP-PBX software hosting the IVR application. It provides an API for initiating and releasing calls, generates a DTMF keypress, and sends the audio file as a speech utterance. In the current implementation, the SIP Utility of Maareech uses the API of PJsua[4] for all SIP-based communications. PJsua can be replaced with another Java based SIP stacks such as MjSip[5] for developers requiring more control.

[2]Dual-tone multi-frequency signaling (DTMF) is used for telecommunication signaling over analog telephone lines

[3]By call queue, we refer to multiple calls scheduled to be initiated by Maareech one after the other.

[4]pjsua is an open source command line SIP user agent (softphone) system `http://www.pjsip.org/pjsua.htm`

[5]MjSip is a complete Java-based implementation of an SIP stack. `http://www.mjsip.org/`

Status Monitor: This module gathers information about the present state of IVR by directly communicating with the IP-PBX software. Any IP-PBX with a *command line interface (CLI)* can interact with Maareech by changing the connection configuration in the Status Monitor. This communication enables Maareech to know about the current configuration of the voice menu played by the IVR application hosted on IP-PBX software such as FreeSWITCH or Asterisk. In the current implementation, we have used fs_cli that is a CLI utility for connecting to FreeSWITCH (IP-PBX software).

Data Handler: This module is responsible for reading input data files (in XML format) created from logs of the IVR application and IP-PBX (FreeSWITCH), and converting it into a Java based object and vice-versa. All XML-based communication is done through JAXB[6]. The schema of XML documents defines the underlying user models used in Maareech. Maareech has been designed with rich data models to capture complex call scenarios. The data file contains logs about user responses and their corresponding contextual information that are generated by logging facilities of IP-PBX software. The read data is supplied to the Logic Processor module to emulate desired user behavior. The Data Handler module is also responsible for creating new objects for telephonic queues and call objects in each queue.

4.2 User Models in Maareech

We categorize currently available IVR applications, e.g. [14], [17], [13], into two categories: Static menu based IVR and Dynamic menu based IVR. Considering that, we have designed two user models to mimic user behavior.

Simple user model: Figure 4.3(a) represents a simple user model that has two composite attributes: meta-data and events. The meta-data attribute has four sub-attributes used for storing the meta-data of the call.

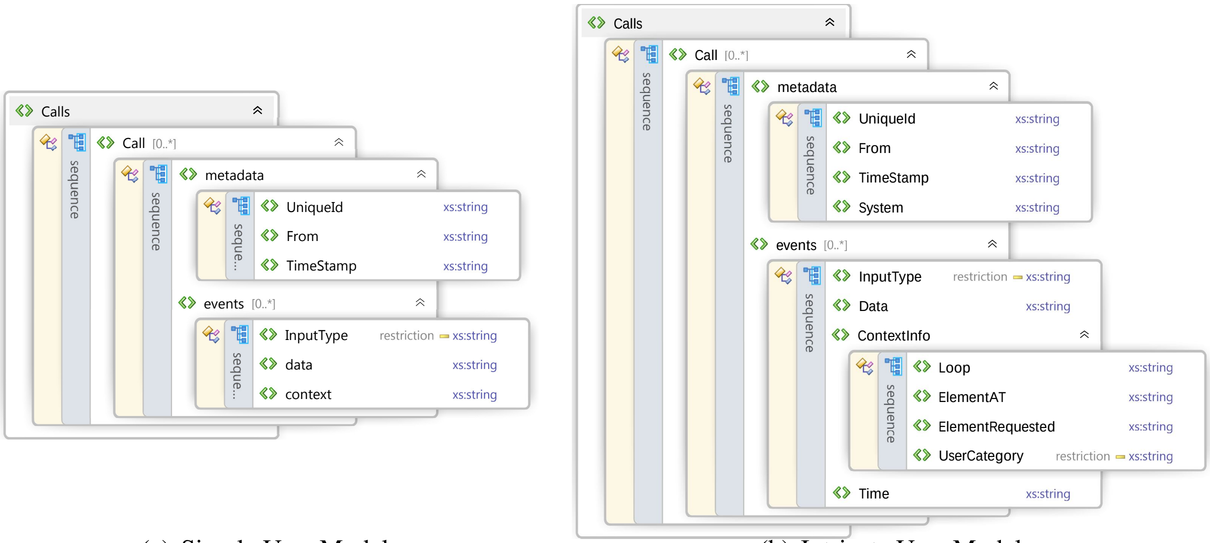
- *UniqueId*: A unique identifier for the call.
- *From*: Telephone number of the caller.
- *TimeStamp*: The time at which this call was made.
- *System*: The system specific identifier where the IVR system has multiple IVR applications running on it.

User responses are captured in an event attribute that has three sub-attributes: InputType, Data, and Time.

- *InputType:* It categorizes user responses into key-presses or audio.
- *Data:* It describes the series of DTMF keypresses or names of audio files, depending upon the corresponding InputType.
- *Time:* It captures the time in seconds at which the input was generated.

This simple user model is sophisticated enough to replay any call emulating user behavior for testing IVR applications that have static menu configuration. However, to test IVR applications where menu configuration may change dynamically, we need a richer user model.

[6]Java Architecture for XML Binding `http://www.oracle.com/technetwork/articles/javase/index-140168.html`



(a) Simple User Model (b) Intricate User Model

Figure 4.3: *Maarech Load: The figure on left side (a) shows CPU usage of Maareech in terms of static, running and total load. The figure on right side (b) shows memory usage of Maareech in terms of static, running and total load.*

Intricate user model: Some IVR applications change their menu configuration based on the time of the day, user preferences etc. Thus, the IVR menu configuration may be based upon information provided by the users, their history, and other factors like time, etc. Faithful emulation of the user behavior of earlier calls in the changed menu configuration requires prediction about user behavior in a new configuration. Correspondingly, it requires that the data used for emulation should have all the intricacies of user behavior required for a new configuration well represented in it. To handle such IVRs, a more intricate XML schema is required as shown in Figure 4.3(b).

The Intricate user model contains a tuple of 4 elements (i.e., InputType, Data, ContextInfo, Time) to describe the user response and the context in which the response was created. The InputType and Data attribute in the tuple are same as in the simple user model. This model has an additional attribute, ContextInfo. It is a composite attribute made up of 4 sub-attributes (i.e., Loop, ElementAt, ElementRequested, and UserCategory).

- *Loop:* It defines the number of times the menu was listened before selecting an option.
- *ElementAt:* It captures the announcement at the time the users made their selection.
- *ElementRequested:* This captures the option selected by a user.
- *UserCategory:* Each user is categorized in one of the two categories: expert or naive. An expert user is the one who selects an option ahead of the full completion of the announcement or just after the announcement was made. The user who waits for the announcement of several menu options before selecting a particular option is categorized as naive.

The intricate user model effectively captures the exact menu options (in case of advanced adaptive IVR systems) at the time of selection. This data is helpful in predicting user behavior in the new context where menu options are reconfigured. The events are regenerated when the contextual information stored in the XML document matches the context in the call.

4.3 Features

In Maareech, calls are emulated using two modes: user emulation using the previous call records, and testing using random DTMF generation.

4.3.1 User Emulation using previous call records:

In this mode, Maareech reads call logs stored in XML files that are obtained from a real world deployment. Maareech supports two formats that capture user models:

Simple user emulation: This user emulation works for currently available static IVR systems. In this format, Maareech replicates call events like key-press and speech recording, as they happened in an actual call. In this mode, Maareech does not assume any assistance from IVR applications hosted on the telephony server (FreeSWITCH or Asterisk). Real data stored for this emulation contains a time-stamp for each event relative to the start of the call. The events are generated based on the time-stamp captured in this model.

Intricate user emulation: The intricate user emulation model has been designed for upcoming personalized IVR systems, where menu options sequence may change in order to give better services. Maareech keeps track of menu options as they were accessed in the original call and responds to a correct menu option even if the menu sequence has changed. In this mode, the announcement of each menu option is assumed to be a state. Maareech responds to this state based on the response to the states stored in the data used for emulation. This mode assumes that the IVR application announces its state over the IP-PBX console as they occur. Maareech does not support sound processing or any other similar technology to capture and recognize the state. The IVR application announces its state on a command line interface. The status monitor in Maareech captures the state information through the command line interface utility of the IP-PBX software. In the current implementation, the application assumes this command line interface to be the FreeSWITCH console. Maareech connects to this console using the fs_cli utility that comes with preinstalled FreeSWITCH binaries.

We have incorporated some more features in Maareech that are helpful in analyzing IVRs for different usages. Maareech provides two basic features in this mode. The trial run of each feature was tested on a machine (HP Probook 4520s) running Ubuntu 12.04 with an Intel Core i5 processor and 2GB RAM.

Call reordering: This feature allows to change the original sequence of call arrivals on the IVRS. It is suitable for analyzing upcoming IVR applications that have dynamic menu configuration. It helps to study the dynamic IVR system that will behave differently for the same calls presented in the different order. In a trial run of Maareech, 1120 calls collected from real world usage, were shuffled in less than 3 seconds.

Number of telephone lines: This feature allows the developer to check problems which may arise due to shared access to system resources (e.g, accessing the same audio file for reading or log file for writing) when multiple instances of the IVR application handles more than one

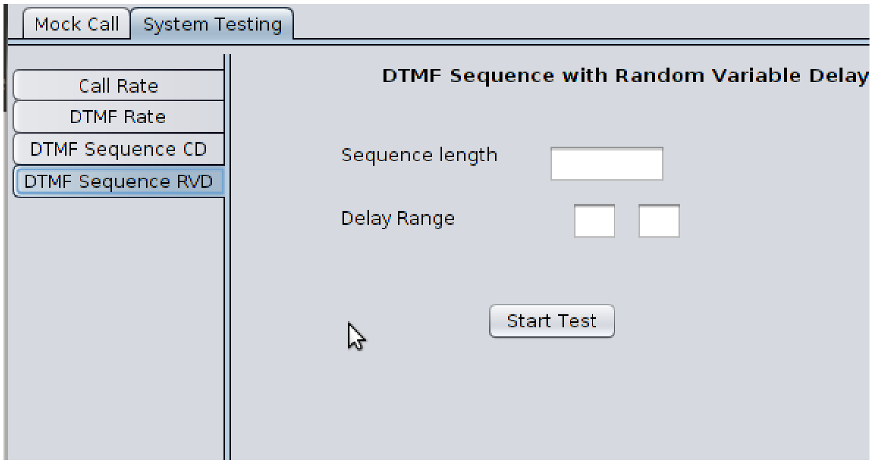


Figure 4.4: *The tabs on the left showing four type of IVRS tests.*

simultaneous call. By default, Maareech assumes a single line connection with the telephony server and only one call at a time is simulated as per the current call sequence. With this feature, the number of telephone line connections can be increased to view the behavior of the IVR application in handling multiple connections. In a trial run, Maareech was able to open eight lines with linphone[7] (a free SIP VoIP Client) as the SIP utility.

4.3.2 Testing using random DTMF generation:

In this mode, Maareech does not read any data file or log to perform tests. These tests are independent of user models and can be used to evaluate the IVR application's system parameters. It generates events like key presses or speech recording based on parameters specified by the user. It supports four types of IVR tests, as shown in Figure 4.4:

Call Load Test: This test helps in measuring the number of simultaneous calls an IVR application can process. Maareech can test an IVR application hosted at a remote location also, which enhances its utility. On starting this test, the system increases the number of calls made to IVR till it fails to handle any more calls. This test reports the integer value at which the IVR application crashes. In our test, we found that FreeSWITCH was able to process 123 simultaneous calls for our sample IVR application under test. The IVR application hosted on FreeSWITCH failed on 124^{th} call because of too many connections open to the MySQL database operating at the backend. We would like to mention that this is not a limitation of Maareech, but of the FreeSWITCH module that connects to database. Maareech helped in identifying it and did not report any failure as FreeSWITCH (call receiving module) was running properly even though IVR application was not accepting additional calls.

DTMF Rate: This test helps to detect the rate at which an IVR application can accept input. Maareech starts this test by sending 1 DTMF per second to IVR application under test and keeps on increasing number of DTMF per second. This test reports the integer value beyond which IVR do not accept more DTMF per second.

[7] www.linphone.org

Table 4.1: Hardware and Software configuration of machines used for experiment

	Machine-I	Machine-II
OS	Ubuntu 10.04	Ubuntu 12.04
Processor	Intel Core 2 Duo	Intel Core i5
Memory	3 GB DDR2	2 GB DDR3
Model	HP Compaq dx7400	HP Probook 4520s

Sequence Test: This test helps to detect the DTMF input sequence at which an IVR application may fail. The Maareech generates random DTMF key sequences, each with a constant time delay, in seconds, as defined by the IVR developers. An IVR developer needs to specify the desired sequence length to be generated by Maareech. At the end of the test, Maareech reports the sequence test cases if any, at which the IVR application fails to respond. In our test, Maareech was able to check 100 sequences, of sequence length 5 and delay of 5 seconds within each option, in 2,504 seconds. In this, 2500 seconds were taken due to the conditions specified in the test and 4 seconds in setting up the call.

Sequence test with random delay: This test detects erroneous DTMF inputs in a more complex manner. The constant delay is not helpful when the length of voice prompts played by the IVR application varies for the announcement of menu options. If voice prompts of menu options are of different lengths, then to test such IVR applications, different delays must be put between each DTMF input generated by Maareech. Random delays help create more test cases than just random DTMF sequences. In this test, Maareech generates random DTMF key sequences each with random time delays ranging from t_1 to t_2 seconds, where t_1 and t_2 are defined by the user in seconds. Ideally, t_1 should correspond to the length of the minimum voice prompt and t_2 be the length of the maximum voice prompt in IVR. This test is more rigorous than the previous sequence test. At the end of the test, Maareech reports the sequence test case at which the IVR application ends the call. In our test, Maareech can check 100 sequences, of sequence length 5 and delay range 5 to 7 seconds, in 3094 seconds.

4.4 Performance Evaluation

To evaluate the performance of Maareech, we conducted an experiment using the *Call Load Test* feature in Maareech. For this experiment, we setup an IVR application written in JAVA and hosted on FreeSWITCH. FreeSWITCH and Maareech were running on two different machines referred as Machine-I and Machine-II respectively, with LAN, 100 Mbps, connectivity between them. We chose this configuration to reflect the real world deployment. Table 4.1, shows the hardware and software configuration of each machine.

We started the *Call load* test feature of Maareech. It initiated a call every two seconds while keeping previously initiated calls alive until the end of the experiment or when they were terminated by FreeSWITCH. In total, we initiated 1024 calls through Maareech. A call in this experiment can be in one of three states:

- Active State: A call that is connected to FreeSWITCH and is not being terminated from either side (Maareech or FreeSWITCH).

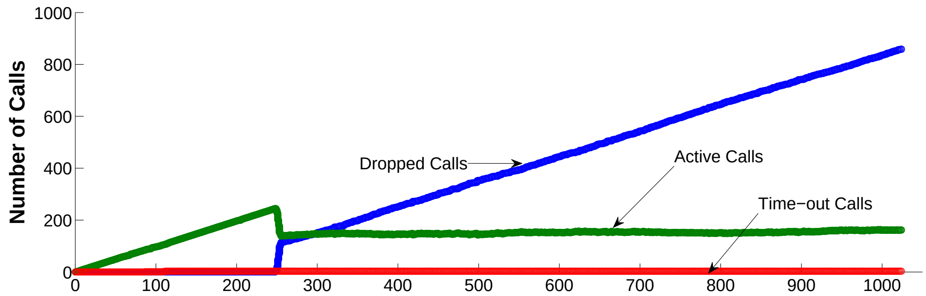


Figure 4.5: *X-axis represents the calls initiated by Maareech. The four lines representing calls in active, dropped, and timed-out states.*

- Dropped State: A call that was connected to but later terminated by FreeSWITCH.
- Time-out: A call whose resources were released by Maareech as it was not able to connect to FreeSWITCH.

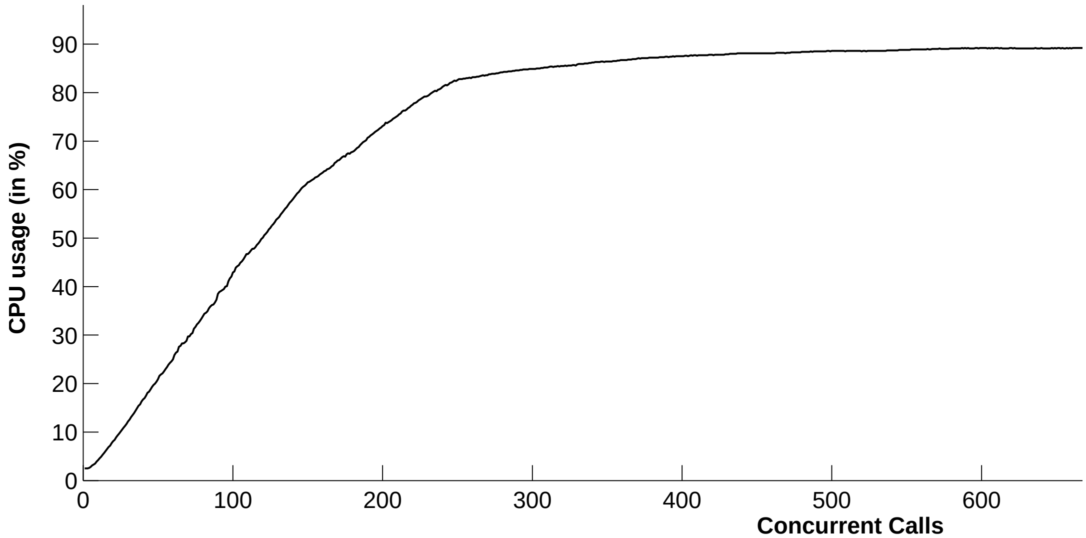
Figure 4.5 shows time-series of call states from our test. Initially, the FreeSWITCH was able to accept the calls as the load on Machine-I was low. Due to this, there were no dropped calls observed till the 233^{rd} call. After this, FreeSWITCH started dropping calls at a higher rate that reduced the count of active calls. The number of Active calls stabilizes around 145 calls, beyond this all new calls are dropped. Our results for the number of active calls (i.e. concurrent calls) for FreeSWITCH running on Machine-I are in-line with the results obtained by other developers as available on FreeSWITCH website[8]. The CPU and memory load were also measured on both the machines during the tests.

FreeSwitch Load: We measure the CPU usage and memory consumption through Linux utilities (e.g. ps) on Machine-I. Figure 4.6 (a) shows the CPU consumption of FreeSWITCH on Machine-I. CPU load saturated around 233^{rd} Call. After this, the rate at which calls were dropped become nearly equal to the rate at which new calls were accepted. We also found that three calls timed-out at 110^{th} call because of high network congestion between the two machines.

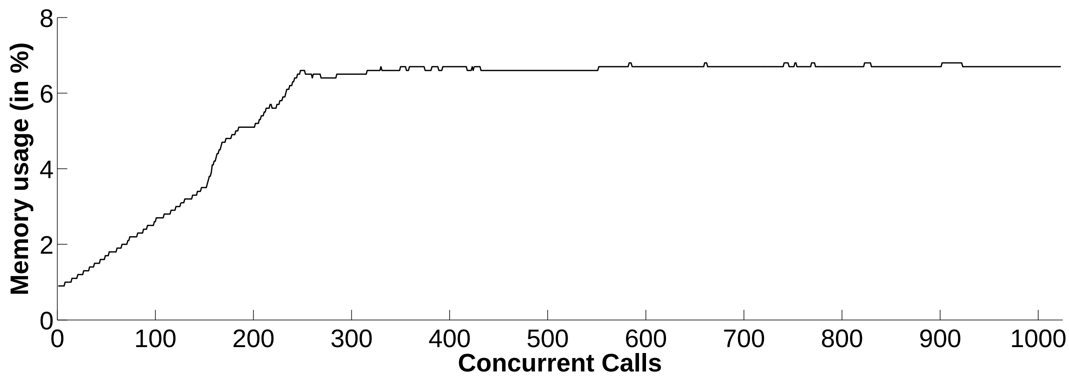
Figure 4.6 (b) shows memory usage of FreeSWITCH in the percentage of total memory on Machine-I. Similar to CPU usage, memory usage also saturated around 233^{rd} Call. This shows that memory requirement of FreeSWITCH did not increase after 233^{rd} call as the number of active calls were saturated due to the equal rate of calls dropped and calls accepted by FreeSWITCH.

Maareech Load: Maareech creates logs of various parameters (e.g. CPU usage, memory usage and call data rate) to collect data related to its performance. We categorize the memory load and CPU load into two categories: static load and running load. Static load refers to CPU usage in creating and holding the call objects. Running load refers to CPU load incurred due to the handling of underlying SIP communication. Total load is the sum of static and running load. We measured static and running load separately as the respected operation is handled by two different processes. Figure 4.7(a), shows CPU usage distribution of Maareech in terms of static

[8]Real-world performance data around the FreeSWITCH community. `http://wiki.freeswitch.org/wiki/Real-world_results`



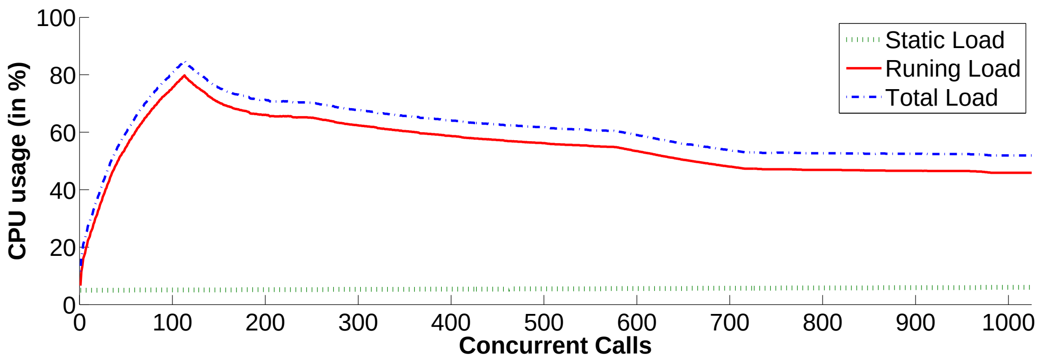
(a) CPU



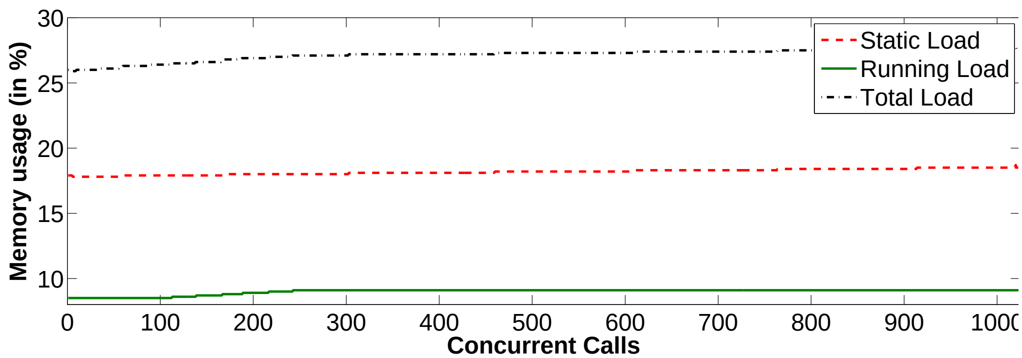
(b) Memory

Figure 4.6: *FreeSWITCH Load: Figure on the top shows CPU usage of FreeSWITCH. The Y-axis represents the CPU usage of one core and X-Axis represents the total number (alive + dropped) of calls connected to FreeSWITCH. Figure at the bottom shows memory usage of FreeSWITCH. The Y-axis shows the percentage of memory used by FreeSWITCH and X-axis represents the total number of calls connected to FreeSWITCH.*

load, running load and total load. We can observe that the CPU usage was maximum at 113^{th} call and decreases and stabilizes after 233^{rd} call. We did further investigation and believe that CPU usage increased because of network congestion as the three calls were also timed-out around maximum CPU usage because of network congestion. We also measured the static, running and total load on memory usage. Figure 4.3(b), shows memory load of Maareech. We find that static load gradually increases from 8.1% to 9.1% (i.e. 1% increase)for emulating 1024 calls. Similarly running load varied from 17.9% to 18.5%. Thus, it shows initiating each call in Maareech has memory load of 20 KB (calculated as 1% of 2 GB system memory divided by 1024 calls).



(a) CPU



(b) Memory

Figure 4.7: *Maarech Load: The figure on top (a) shows CPU usage of Maareech in terms of static, running and total load. The figure on bottom (b) shows memory usage of Maareech in terms of static, running and total load.*

4.5 Related Work

Simulators have been used in various computer science domain like network simulators [102], and analog and digital circuit simulators. Simulators in the IVR domain are primarily used to simulate call-centers[9]. Various industrial tools provide pre-deployment testing of IVR applications. It mainly includes testing of VoIP infrastructure [3]. *Empirix Hamper*[10] provides an extensive tool-set for IVR testing, monitoring and analyzing end-to-end IVR deployment. Nexus8610[11] is a traffic generator that simulates user behavior of various communication technologies including 3G / 2G Mobile, VoIP, and PSTN. Cyara Solutions[12] is one of the industrial players that provides IVR testing as a service. Tools like Call center simulators[13] help estimate the resource requirements for optimal performance of IVR.

Although, the industry has a variety of tools for testing IVR at the infrastructure level, developers are still doing manual testing or writing customized test scripts for each IVR application.

[9] `http://www.call-center-tech.com/`
[10] `http://www.empirix.com`
[11] `http://www.nexustelecom.com/products/nexus8610/`
[12] `http://www.cyarasolutions.com/`
[13] `http://www.xjtek.com/anylogic/demo_models/4/`

Hence, an emulation tool like Maareech, which is capable of mimicking user behavior is required to automate testing of IVR applications.

4.6 Discussion

Measuring the performance of any interactive system involving human is a challenging task. In this chapter, we have presented Maareech - a call emulator for user behavior. This enables a developer to test the IVR application from user experience perspective. We presented two user models for testing and measuring the performance of different IVR applications. User model based testing provides characteristic for modeling different users easily [35] and reduces human effort. We believe future work in this direction can further improve the system capability to mimic user behavior. With the increasing use of IVRS, it is imperative to have a robust testing tool for IVR applications. We have built Maareech for the same purpose. Over the time, Maareech has evolved to be very robust and has been used in testing of IVR applications that are deployed in the field. We have made the maareech publicly available for research community[14]

[14] `https://github.com/siddasthana/IIITD/tree/master/CallSimulator1.1`

Chapter 5

Intermittent Connectivity: Information Exchange through Smartphone Application

Telephonic solutions using smartphone application typically require cellular data connectivity for supporting information exchange. Especially for developing regions, researchers have reported the data connectivity issue while working on mobile based systems in rural areas [57, 91]. In the absence of reliable data connectivity, the smartphone applications may use some opportunistic methods to defer the data transmission till the data connectivity is restored. However, such an opportunistic method is not advisable for scenarios where the user may need minimal information and should not wait for the delay in the information exchange. Therefore, solving for the data connectivity issue can open up the possibilities of deploying smartphone solutions for a variety of use cases like m-banking, m-health [49], etc.In many use cases, the requirement for data transfer is minimal, e.g., sending SOS signal in an emergency situation or sending the summary of vital health parameters (Average heartbeat rate, blood pressure, etc.) in the case of a medical emergency, etc. In such use cases, information can be sent out or retrieved through SMS or voice channel instead of waiting for data connectivity to restore.

We envision two possible ways for supporting smartphone applications for regions with intermittent data connectivity. The first way is to design and develop a system with multimodal connectivity where information exchange can happen over multiple mediums that include voice, SMS, and data. This reduces the smartphone solutions dependency on cellular data and also ensure that some use cases of such solutions are always supported by voice and SMS. The second way is to build a mechanism that can enable data transfer through voice channel when data channels are not available with cellular connectivity. We discuss both the ways in details and present our findings from field deployments and real-world experiment showcasing our proposed solutions.

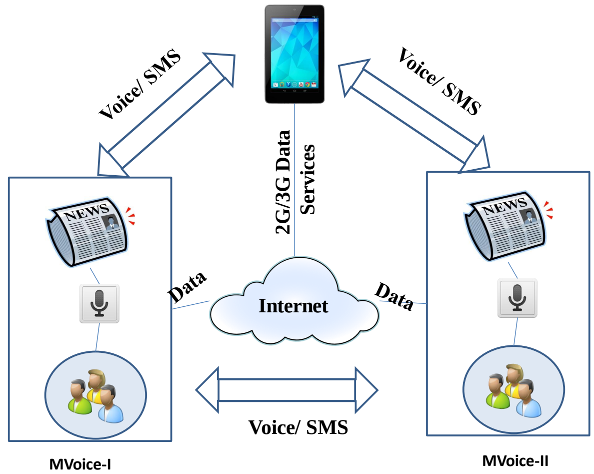


Figure 5.1: *Possible configuration in which two or more similar system like MVoice can connect with each other, the Internet resources and the telecommunication devices like tablet and phones.*

5.1 Mvoice: A multimodal system to integrate Smartphone solution for Information exchange

Inspired from multimodal connectivity of web-application like online banking, email services and social media where web-application is supported/accessed through alternate channels like SMS; we propose multimodal connectivity for mobile based information exchange to utilize smartphone based capabilities with voice and SMS based information exchange. For instance, in banking services, connectivity through SMS is used as security solution to provide OTP (one-time password) to the online banking user. Similarly, SMS can also be used to check emails or to post a message on social media like Twitter. We tried to extend the same capabilities for mobile based information exchange through multimodal connectivity as shown in Figure 5.1. With multimodal connectivity, the smartphone solutions can also be interfaced with other devices, voice application, web services, etc. making the reach of mobile based information exchange much wider. We show the usefulness of our multimodal communication capability of our system through a real world study. In such multimodal communication, smartphone applications can send an SMS to request for information when the Internet is not available to pull out the information from the server. Reply to such request from server side can be delivered in the form of audio through voice calls or text through SMS. The use of the Internet is to upload bulk data from the field whenever we get good Internet connectivity from the field.

We have named our system as MVoice. We propose MVoice as a scalable solution for mobile based information exchange with multimodal connectivity to the web and scalable to a bigger set of devices like tablet, smartphone, etc. Deployment of MVoice for illiterate and poor people in the rural area helped us to form several guidelines along the lines of improving the usability of the system.

5.2 System Design

We designed and developed *MVoice* in Django-Python which has an easy-to-use web interface for telephony system. The interface provides features to schedule calls and SMSs to be sent in future and configuring other aspect of telephony applications. Figure 5.2 shows a snapshot of the system interface showing callers' mobile numbers, the specific voice application (in this case, it is the Broadcast app.) that handled the call, scheduled time of calls, and a cut-off timing to stop making the call. The current status of a call is displayed in the form of a green or a red dot. Green means the call is successfully received by the caller and red means that either the call is yet to be made or call was not successful. Different filters on the right-hand side provide an ability to filter the calls based on voice application, or by date, or by successful or unsuccessful call. More filters can be added if required. Additionally, each column can also be sorted in ascending or descending order. The system has inbuilt application for broadcasting voice and text messages. Other applications can be developed as and when needed. To make the system

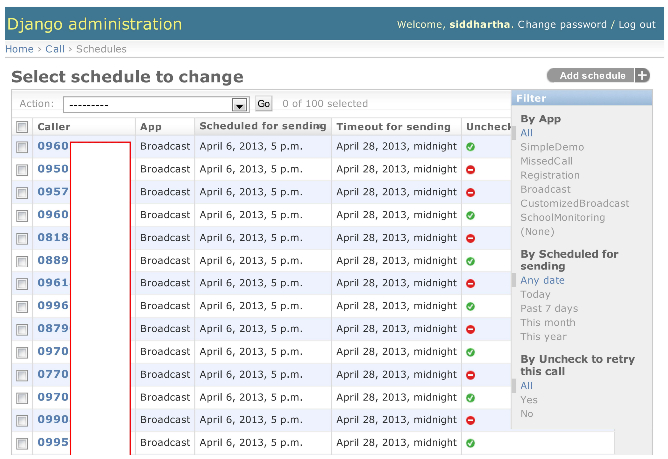


Figure 5.2: *Web Interface of the System*

a portable solution which can be installed on a Laptop, the telephony system is developed using open source telephony platform FreeSWITCH and interfaces with a GSM based USB dongle like HUAWEI E169; it can be extended to multiple dongles using external powered USB hub. The telephony application's code for call scheduling and voice applications are written in JAVA. Voice applications can have predefined messages to play while making a call or can take *mp3* files as input from the administrator through the web interface. We have also developed a TTS to read out the digits in a local language - *Telugu* - as a proof of concept. We used this TTS for our deployment in Andhra, India with our voice application. The TTS currently has a limited functionality to read out numeric text only. A voice application can be configured to respond to missed calls and can call back to the user. SMS listening application can be configured to listen to an incoming SMS and can initiate a call in response to an SMS. The telephony system has a modular architecture. The new telephony (voice, SMS, both) application can be introduced at any time, and the inbuilt scheduler can use this new voice or application to make calls. The system also provides reports for system usage through easy to use web interface which has sorting and searching features on multiple fields like the timestamp of a call, gateway used, callerid, telephony application, etc.

5.2.1 A rural deployment of MVoice

We started our work from a small area in Andhra Pradesh, India where over a period of nine months we studied the deployment of MVoice in rural areas. Andhra Pradesh is the first state to launch the MGNREGA scheme to alleviate poverty in rural areas through government employment schemes. Started in 2005, MGNREGA (Mahatma Gandhi National Rural Guarantee Act) is one of the biggest social experiments to strengthen the rural poor people with 100 days guaranteed wage employment. The scheme ensures work to be generated in the nearby villages of the person's residence. Every person is entitled to receive wages at the wage rate for each day of work. Failing to provide employment will result in paying an unemployment allowance to wage seeker. The disbursement of daily wages shall be made on a weekly basis and cannot be delayed beyond the fortnight. Every work allocated through MGNREGA has typically following phases:

- *Demand Registration:* People submit their application for demanding the work to the Gram Panchayat. For this, the person should have a valid Job card with photo identification.
- *Work Allocation:* Allocating work to the people who registered for work within 15 days of demand registration.
- *Field measurements:* Taking the measurement of work completed on a weekly basis by a field assistant.
- *Payment:* Disbursement of payment on weekly basis.

According to statistics available with the state civil supplies department, the population of Andhra Pradesh is 84 million out of which 95% are below the poverty line [84]. The State of Andhra is divided into 23 districts. We cover only Ghattu Mandal of Mehbubnagar district which has total 61 Mandals.

Our collaborators were actively involved with NGO's, local bodies and social audit teams working on MGNREGA in villages of Ghattu Mandal. Through them, we tried to study the working of NGO's and activists (hereafter refer as local bodies). We found that local bodies play a bigger role than just promoting MGNREGA scheme. But as a first step, we kept our focus on issues related MGNREGA scheme. We found out that they campaign from one village to another to educate people about the scheme, the rights government has provided through schemes and the potential benefit they can get through this scheme. For each campaigning, they use to identify local young literate people who can assist them in the campaign. In the campaign, they used to distribute their phone numbers so that in the case of query they can be reached on the phone. In this interaction, we also get to know many people has contacted to local activists and NGO over the phone for querying related to MGNREGA and other issues.

5.2.2 Experiences from field: Information dissemination and collection through MVoice

We felt the need of various information collection from the field. We need to collect mobile numbers of rural people and their corresponding job card number allotted them by Gram Panchayat. A job card number helps in identifying the status and work history of a household in the MGNREGA official website. The collected mobile number will be used to deliver personalized information to the rural people. Apart from that we have prepared an extensive survey to collect demographic details like village name, age, widow-pension receiver, mobile phone usage and sharing habits, the type of house, etc. Secondly, we were not sure about their proficiency in dealing with automated mobile based ICT tool. We plan to give them a demonstration so that they can easily use the system. Further, the system should be able to send them different information which includes creating awareness about MGNREGA, starting of different work cycle under MGNREGA like work allocation, payment release, etc.

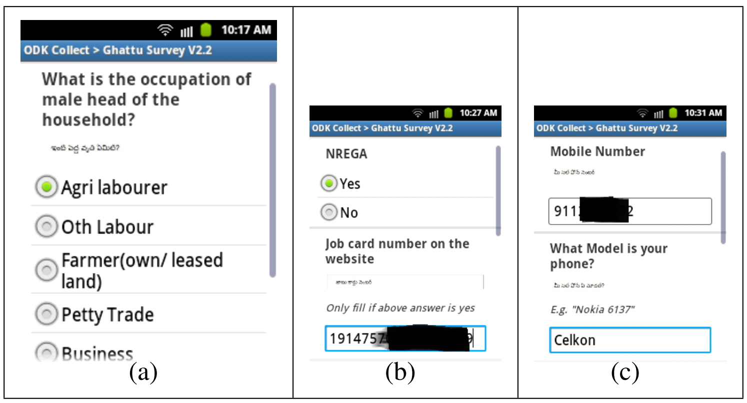


Figure 5.3: *ODK Form Screenshot*

Data collection was an extensive task. To be effective in the field we have used a mobile data collection methodology. We have used an android based tablet (dual SIM, GSM 3G enabled) for this purpose. The data collection application for survey is designed through open source set of tools Open Data Kit. Figure 5.3 shows the screenshots of our data collection application. We also installed application to measure the signal quality of mobile networks, GPS and other data understand the possible issues that may arise due to problems with infrastructure of mobile network ability.

To avoid an additional visit to workers for demonstrating the system, we clubbed the data collection and demonstration in the same visit. To achieve this, our ODK application generates SMS on completing the survey form in the form of tuple<Mobile No, Job card No>. This SMS is received by MVoice. The information contained in SMS is read by Mvoice and registers the person's mobile number into our system. This triggers a demonstration call to the number specified in the SMS. For each worker, we interviewed during the field visit we tried to demonstrate the system through the above mentioned mechanism. The call contains a pre-recorded voice message in Telugu to tell the people that you have been now registered with our system. The exact prompt we used in Telugu when translated to English is as following

"You've been registered into Ghattu MGNREGA helpline. Next, you will receive a text message with year to date wage earnings on your job card."

This way we were able to make an observation on an automated call received by the rural people registered with us through ODK data collection application. We observed their reaction on receiving the automated call and their understanding about the voice message played. This experiment helped us not only to study their reactions on receiving an automated call but also helped us in introducing them. We also tried to verify that whether people understand the basic instruction given to them through voice prompts. For this, we deployed another application on the MVoice and named it as Miscall application. The Miscall application initiates a callback to a mobile number on receiving a miscall from any mobile. The application contains an instruction to record any voice message after the beep which can be any complaint and grievance of the rural people. The experiment helped us to study the ability of the people to understand instructions on the voice message irrespective of their numerical literacy.

We have also tested the impact of using the voice of a familiar person in communicating the message or instruction given through the system. We made the voice instruction of Miscall application recorded in a female voice which is introduced to them as one of the Female activist popular in the field. In the voice message played to the people by the system, first, the person introduces her as Ms. X (A popular and well-known social activist in the area) and asked them to record their message after the beep. The recording has 30 seconds timeout after which a thank-you message is played in the same female voice by saying that we will get back to you. The exact call flow and voice messages, when translated into English, is as follows:

Prompt#1

Namaskar, I am Ms. X calling from our Ghattu MGNREGA helpline. Please tell me your name.
< Recording for 5 seconds >

Prompt#2

If you have complaints or suggestion you can tell us about it. One of our assistants will soon respond to your message. Please Record your message after the beep.
<Recording for 30 seconds>

Prompt#3

Thank you. Have a nice day

Finally, the data collected through tablet device is later synced to Google server whenever our volunteers get good the Internet connectivity. Our system also fetches the data from MGNREGA official website about payment, work allocation and different other information related to MGNREGA working. The data is used to convey timely and useful information to the worker. Currently, data fetching from MGNREGA website is manually done but the discussion with MGNREGA officials has assured an automated way of fetching the data from the MGNREGA server through an FTP channel.

Figure 5.4 shows the stepwise process. ODK application running on Android-based Tablet

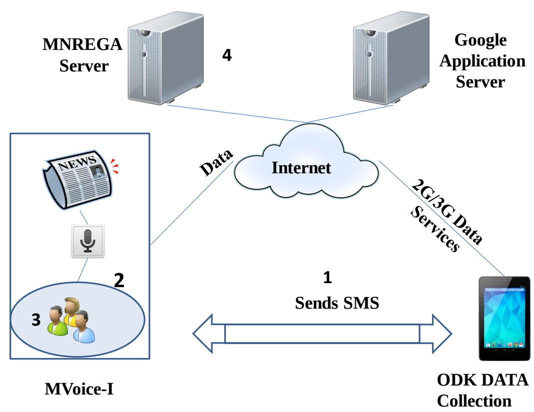


Figure 5.4: *MVoice application configured for MGNREGA.*

sends an SMS on form completion. The SMS is received by MVoice system which adds a new people in the MGNREGA community based on mobile number specified in the SMS. This addition of new people in the community triggers the Demonstration call to the mobile number of the recently added person. Based on the Internet availability other survey data is uploaded to Google Application server from where data can be exported in the form of *JSON* (JavaScript Object Notation) or *xlsx*. Additional information about the person and his/her information in MNREGA server is retrieved based on the JobCard number received in SMS. Later on, this information is used to provide different information to the user like payment released for a person etc.

5.2.3 Data Collection and Usage Statistics

Although first field visit to test the system functionality was only for four days across five villages but we are constantly registering more and more people into the system with the help of volunteers. The following demographic came through analyzing the data first 825 households registered with us through the volunteers. We have collected different demographic details. But here in this section we are only reporting to the details concerned about mobile phone and their usage. Table 5.1 shows households with different number of mobile phones with them. We found that out around 3.2% of the household did not have phones. Around 87% of household had one mobile phone with them.

We found that all the mobiles were prepaid. About 19% people did not know about their calling rates that their mobile operator charge them for making local calls and about 68% people do not know their calling rates for STD calls. When asked for sharing of mobile phone 83% said that they did not share mobile phone. Out of the people who shared their mobile 50% said that they will share it for both incoming and outgoing and only 1 person said that she will share it for outgoing only. 292 out of 571 who reported their monthly expense on the mobile said that they spent less than the Rs 100 in last 3 months.

Next, we tried to find how many people can open SMS and out of that how many can read SMS. We also figured out the number of mobile phone that supported receiving the text message

Table 5.1: *Mobile Phone Statistics*

# Mobile	# Households	Percentage
0	27	3.272727
1	722	87.51515
2	61	7.393939
3	9	1.090909
4	3	0.363636
5	1	0.121212
6	1	0.121212
7	1	0.121212

in Telugu. Out of 825 only 252 can open an SMS without help and 49 can open it with some help. The rest of 497 (i.e. people at-least 1 mobile phone) can not open an SMS. Out of 301 (252+49) only 209 could read SMS and 61 with some difficulty. We additionally found that 112 were regular user in using sending the text facility on mobile phone and 66 were able to send it with some difficulty. The support for Telugu font on mobile phone was found only with 128 people. Our findings are in line with other research studies conducted in developing region [49].

5.2.4 Discussion and next steps

MVoice deployment has provided us an opportunity to understand the challenges that can come to a mobile based ICT deployed in rural geographies of developing region. MVoice tried to overcome infrastructure challenges due to its multimodal capabilities to support information exchange. The information was delivered and requested through voice channels. Towards this use of miscal app initiated the request from user side that was served by the server in the form of audio content through the voice call. The use of SMS also enabled information exchange between smartphone application and the server. The bulk upload of data was deferred till the Internet service is restored. We had 3G enabled Android tablet, but the Internet was not available all the time which delayed the data syncing to the server. As described earlier, we also tried sending a small amount of survey data (i.e. sending tuple <Mobile No, Job card No> from the tablet on completing the survey) through SMS. We also observed some of these SMS were received at Mvoice end with a delay of 3-4 hours. This delay was beyond the desired limit for us as we can not wait 3-4 hours to demonstrate the system usage to every rural people. We found that availability of voice connectivity was much more significant than SMS and the Internet. Thus building an effective data transmission using voice channel of telephony may result in more reliable automation solutions for such scenario.

5.3 Acoustic modem: Enabling Data Transmission through voice channel

The cellular network is primarily designed to support voice connectivity hence ensures much more availability to voice channels than data channels. This has led researchers to devise a mechanism to use voice channel for data transmission when the data channel is not available. We discuss relevant research that helped us to identify the existing research gaps and the potential opportunity to improve state of the art.

5.3.1 Related work

Even though, the voice connectivity through cellular services has good coverage in the developing world; the data connectivity is largely limited to urban areas with poor connectivity in several rural areas [71]. A large-scale experiment conducted in developing world by Zahir et al. have tried to investigate the availability of cellular data networks across different rural and urban locations [57, 69, 91]. They have found several instances where the data transfer stalled for a large duration. Despite the technical feasibility to provide enhanced data services in such areas, economic reasons prevent the telecoms from it. These economic reasons include low purchasing power and low demand for active Internet subscription in these regions. Other economic reasons include high capital and operation cost, lack of reliable electricity supply, etc [34].

Several application domains that have an interest in rural areas like micro-financing, agriculture and healthcare can be enabled by having just a low-bandwidth point to point data connectivity. Though SMS text messaging service can serve as a simple data layer, but it cannot be used in the application requiring real-time data exchange. Seeing opportunity in such a scenario, previous research has tried to send data over GSM voice channel [5, 60] as voice connectivity is guaranteed by telecom even in the absence of data connectivity and has data rate much higher than SMS-based data transfer [34]. Such data transmission over the voice channel is known as acoustic modem where data is transmitted using modulation and demodulation of speech signals transmitted over the voice channel. The application of such an acoustic data modem has been shown in the area of encrypted end-to-end secure voice or data over the GSM voice channel in [27], for real-time and secure transmission of Point-of-Sale (POS) transaction information from the POS terminal to the financial host [59].

Till date, the research studies have focused on building acoustic modems that either improves data rate of transfer or reduce the error rate for robust data transfer. LaDue et al. [60] design a data modem for the GSM voice channel by training over the EFR codec having a throughput of 2 kbps with 10^{-6} Symbol Error Rate (SER). The symbols designed in their work were specific to EFR codec and may not work on other GSM codecs. Shahbazi et al. [90] designed a data modem having a data rate of 2 kbps with $1.5\text{x}10^{-5}$ SER and overcomes the disadvantage of huge symbol search space in [60]. Using QAM technique, Chmayssani et al. [31] proposed a flexible approach for EFR channel that allows data transmission up to 3 kbps with a binary error rate (BER) lower than $3\text{x}10^{-3}$. However, their results are purely simulation-based. Ali et al. [5] used M-FSK modulation to achieve low BER (i.e. $< 10^{-2}$) . They experimented on the real-channel test-bed and achieved data rates between 80 to 250 bits/s. Most of the previous studies assume the knowledge of underlying voice codec used in GSM [26, 31, 70, 90].

Such a modulation scheme that is designed for a specific codec have achieved good performance in terms of data rate or error in transmission. However, in GSM a codec or codec modes (as in Adaptive Multi-Rate Codec) can change on the fly depending upon channel condition. Dhananjay et al. have tried to overcome this problem by proposing a generic modulation scheme that is codec independent [34]. They achieved 1.2 kbps without assuming the knowledge of underlying voice codec [34]. Although, this generic scheme overcomes the problem of codec dependency, their data rate is much lesser than codec specific modulation scheme.

5.3.2 Research Gaps & Design opportunities

The narrowband voice channel in GSM (300Hz - 3400Hz) is primarily designed to support voice and is highly unpredictable for data transmission over the GSM voice channel. The cause of the error depends on a number of factors ranging from Voice Codec distortions to GSM Channel's instantaneous wireless link quality, etc. Despite the challenges, several attempts have been made to use voice channel for data transmission. Here, we identify factors that can optimize the data transmission over the voice channel.

- Adaptive System The performance of a data transmission technique is observed best in a particular codec/codec-mode where they were simulated during experimentation. However, codec and their modes keep changing based on channel conditions and thus requires data transmission technique to adapt based on changing codec and their modes. We propose an adaptive system that possesses the capability of switching to a more robust modulation scheme if the previously used scheme does not conform to the current channel conditions.
- Error Metric as Feedback The transmitter end should be aware of current transmission error rate to build an adaptive system for data transmission. In general, data transmission uses Forward Error Correction where only receiver end can estimate the transmission error. Thus, a feedback mechanism can be build to estimate transmission errors at sender's end.
- Using the Reverse Call Link Previous studies have utilized only the forward channel (i.e. sender to receiver) for data transmission. When a call is in session, bandwidth is allocated for both the forward and reverse call links, that means the cost of data transmission remains same whether we utilize the reverse call link or not. So, here we propose the use of the under-utilized reverse call link of the call session (originating from the transmitter unit) for carrying the symbols of a predefined Error Control Code (block codes such as CRC, Reed-Solomon, etc.) calculated over the last received frame of data symbols (at the receiver unit) which will serve as acknowledgment in the proposed protocol.

5.4 Adaptive framework for transmitting data over voice

We propose a solution for data connectivity issues based on the concept of acoustic modem. The proposed acoustic modem incorporate an adaptive framework that can change the modulation scheme based on the error rate of data transmission. Even though the voice call has two channels, the previous work was primarily focused on using the forward channel. This forward channel bears a significant load of carrying Forward error correcting (FEC) code which lowers the effective data rate. In FEC, data is coupled with error codes while transmitting it from the sender to receiver (hereafter referred as forward channel) whereas, the channel from receiver to the sender (hereafter referred as backward channel) is utilized for sending the acknowledgments. Proposed framework has decoupled the actual data from its error correction/detection codes, and the forward channel is used only for sending the data. The proposed framework utilizes the backward channel for error detection. The use of backward channel has effectively reduced the burden of transmitting error detection and correction code in the forward channel which carries the actual data and thus can further improve the effective data rate.

5.4.1 Proposed framework

In this section, we describe the system architecture of the proposed acoustic modem which mainly consists of transmitter and receiver units as depicted in Figure 5.5.

- Transmitter Unit The function of the transmitter unit is to send the acoustically modulated data and receive the acknowledgment by adaptively changing the modulation scheme for error control. This unit consists of following components:
  - Bit to Message Converter: It groups the input bits into messages D of K bits each.
  - Adaptive Selection Unit: It is further divided into following blocks:

    Modulator: It converts the messages into acoustic symbols according to the parameters (symbol length L, frequency dictionary) of the selected modulator model.

    DEMUX/MUX pair: Based on the current error rate of transmission this block selects an appropriate modulator model from n existing models.
  - Framer: Its function is to combine the M acoustically modulated symbols from the modulator and adds synchronization in the beginning and end using chirp signals to form a frame. The framing of data messages ensures signal alignment at the receiver.
  - GSM Voice Encoder: It transmits the speech-like signals from the modulator + framer block to the GSM network cloud. The function of the GSM voice encoder is to compress the digital data and transform it into analog signals to transmit it over the air.
  - Acknowledgment Decoder: The transmitter end continuously listens to the reverse link for acknowledgment frames. The size of acknowledgment frame in the proposed protocol is 40% of the data frame's (on the forward call link) size. These messages provide the error statistics of the modulator model currently being used.
  - Error Sensing & Decision Module (ESDM): This module handles the adaptive selection of modulator model. It is further divided into following blocks:

    Error Sensor: The decoded acknowledgment messages are the parity frame P_t of the data message sent through the reverse link. These parity frames are encoded using Error Control Codes. In our simulation, we have used Reed-Solomon error control codes. A bitwise comparison of parity bits received from reverse link and parity

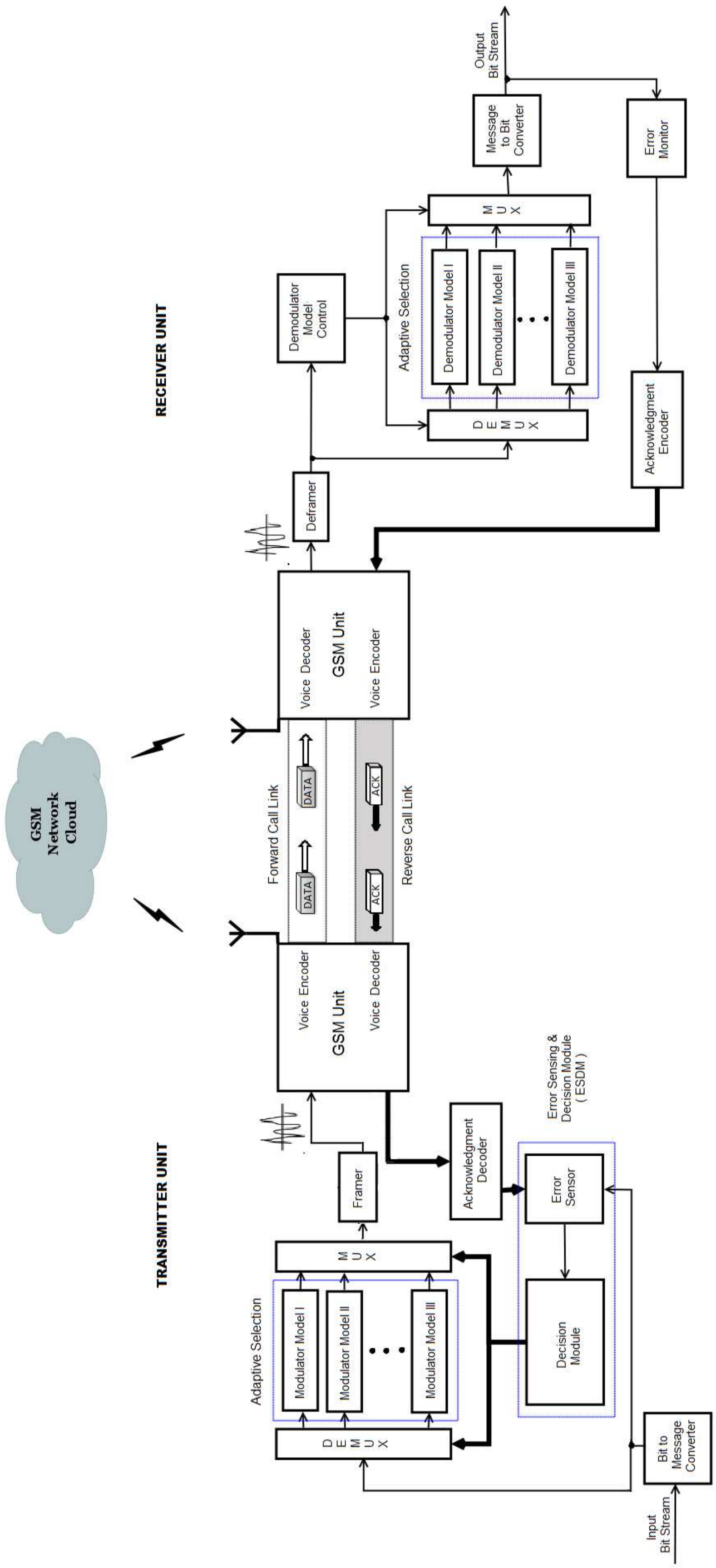


Figure 5.5: *Adaptive Framework: Showing various modules of transmitter and receiver unit in the framework*

calculated on actual data helps in estimating the error rate in the forward link.

Decision Module: The error statistics of data frame sent by the Error Sensor block is then fed into the Decision Module. Its function is to control the states of the DEMUX/MUX pair at the transmitter.

- Receiver Unit The function of the receiver unit is to convert the received acoustic signals into a message and send the acknowledgment packets over the reverse call link. This unit consists of the following components:
  - GSM voice decoder: It just performs the reverse operation of the GSM voice encoder. It converts the GSM signal back to acoustic symbols.
  - Deframer: The output of the GSM voice decoder is fed into the deframer block that extracts M data messages from the received data frame by locating the positions of these synchronization chirp signals using their unique auto-correlation property.
  - Demodulation Model Control: The function of this block is to control the state of DEMUX/MUX pair at the receiver end. It calculates the value of the parameter (such as symbol length L, frequency dictionary) to detect current modulation model. For example, the modulation/demodulation scheme presented in Hermes [34] employed the FSK demodulation technique and estimated the parameter by calculating the difference in frequency of the current and previous sinusoid of a data message.
  - DEMUX/MUX pair: The state of this pair triggers the selection of the demodulator according to the modulator at the transmitter end.
  - Demodulator: It converts the acoustic symbols back into data messages D' according to the modulation mechanism used at the transmitter.
  - Message to Bit Converter: It converts the group of demodulated data messages D' into K bits thus forming the output bit stream.
  - Error Monitor: The output of the demodulator block is also fed into the Error Monitor. It is the first step in making the acknowledgment packets to be sent on the reverse call link as feedback.
  - Acknowledgment Encoder: The function of this block is to modulate the parity message P from the Error Monitor into symbols interpretable by the Acknowledgment Decoder block at the transmitter.

5.4.2 Acknowledgment Protocol

We use standard Reed-Solomon error correcting codes [101]for designing Acknowledgment protocol. Firstly, we assume the number of errors n_{err} that may occur in M message symbols is less than the code minimum distance. It can be expressed as

$$M * e(orn_{err}) < d_{min} \tag{5.1}$$

where e is the fraction of errors that has occurred in M message symbols. Secondly, we assume that no burst errors occur in the frame of M symbols probably due to VAD triggering for the duration of 20ms. Lastly, we assume that the reverse call link (or backward channel) is noise-free. We propose an alternative to FEC where transmission of parity bits can be deferred till the transmission makes the decision whether to send parity bits or not. We name this mechanism as Minimalistic Forward Error correcting codes as parity bits are only sent in case of the erroneous frame. Algorithm 1 shows the parity bit that is sent to the receiver if n_{err} is greater than zero. On receiving the parity bits, receiver corrects the data frame through Reed-Solomon error correction. In case of error-free frame a T_{flag} is transmitted which is just a single symbol. On receiving T_{flag}, receiver finalizes the data frame (which is error free) without any error correction. Thus, if channel conditions favor a particular modulation scheme, then it's error-free frame will also increase. In that case, MFEC will significantly reduce the overhead of error correction.

Algorithm 1: Construct MFEC from error reported on backward channel

---

Given n_{err} already calculated on the previous data frame
if $n_{err} > 0$ **then**
  Transmit parity bit
else
  Transmit T_{flag}
end if

5.4.3 Evaluation

Our goal is to study the performance difference between conventional data transmission techniques (which use single modulation schemes) and proposed framework using an adaptive selection of different modulation schemes (in our case three schemes). We performed real-world experiments for data transmission over GSM voice channel and channel characteristics learned through it were used to evaluate the performance of our adaptive framework in simulation program designed and developed in Matlab. For the experiment, the hardware used for the system is described in Figure 5.5. It consists of Raspberry Pi[1]as transmitter, a desktop machine running on 2.53 GHz processor as a receiver and a Huawei E173 GSM modem (on Vodafone GSM network) as GSM voice codec unit. Both the transmitter and receiver run Asterisk Telephony Server[2] for handling calls.

Experiment Design

For the experiment, the adaptive framework is designed to have three modulator models. The modulator schemes were designed to have a data rate comparable to those mentioned in the literature. Dhananjay et al. [34] have quoted a raw data rate of 1.2 Kbps based on real-world

[1]http://www.raspberrypi.org/
[2]http://www.asterisk.org/

experiments (other studies are simulation based and are on specific codec). So we chose one with the raw data rate of 1.2 Kbps, other with data rate higher than this and lastly one with lower data rate.

Modulation: The proposed digital modulation technique is based on the PCCD-OFDM-ASK principle implemented in [34]. In Orthogonal Frequency Division Multiplexing (OFDM) technique, the data is encoded over sub-carriers. The frequency spacing between each sub-carrier is such that they are orthogonal to each other therefore it is the reciprocal of the useful symbol period T. For the GSM voice channel, the sampling rate is 8 KHz. The sub-carrier frequency spacing is defined as

$$\triangle f = 8000/L \tag{5.2}$$

where L defines the number of samples in each message symbol. As the GSM voice band is standardized as [300-3400] Hz, the sub-carriers must be well within this band to ensure that the modulated symbols are speech-like. For the OFDM modulation scheme defined in this paper the number of sub-carriers is 3. The number of bits K in each message symbol is 6. This gives the symbol population as

$$N = 2^K = 64 \tag{5.3}$$

The symbol design rule is as follows:

1. The frequency set F is mapped as,
   F = $\{flag_1, f_0, f_1, f_2.f_9, flag_2\} \in$ [300,3400] Hz
2. Break numbers into individual digits as D1 and D2.
3. Symbols are encoded as integers from 0 to 63. Their representation in time domain 0 < t < T is:
   if $D1 - D2 < 0$

   $$S_{D1D2} = sin(2\pi f_{D1}t) + sin(2\pi f_{D2}t) + sin(2\pi \\ flag_1 t)$$

   if $D1 - D2 = 0$

   $$S_{D1D2} = sin(2\pi f_{D1}t) + sin(2\pi flag_1 t) + sin(2\pi \\ flag_2 t)$$

   if $D1 - D2 > 0$

   $$S_{D1D2} = sin(2\pi f_{D1}t) + sin(2\pi f_{D2}t) + sin(2\pi \\ flag_1 t)$$

4. Finally, normalize the symbol power.

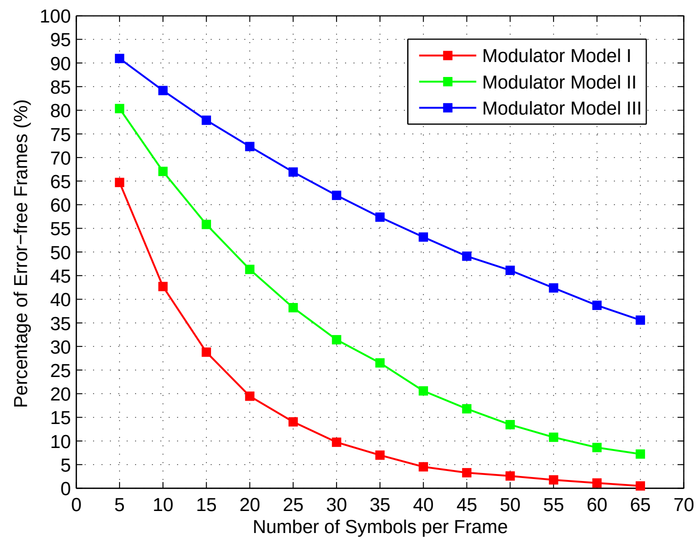


Figure 5.6: *Variation of error-free frames with frame size for all three modulator models*

Note that this symbol design rule gives N up to 100, but we select $N = 64$ so that K is an integer. The demodulation part is simple and uses FFT to find out which of the three sub-carriers are present out of the set F. It uses the same rule as described above for decoding the integer from 0-63.

The three modulators that we defined differ in terms of L, $\triangle$f, raw data rate (D_{raw}) and symbol error rate (SER) performance. We conducted a real-world experiment to make three simultaneous calls originating from Transmitter to Receiver each using one of the three modulator models summarized in Table II. Both mobile stations (MS) were operated over Vodafone (cellular service provider). We transmitted the same set of 2 x 10^5 symbols (196 KB text file of random characters mapped as integers from 0-63) in each call session over the forward call link. The duration of the call for modulator model I, II and III was 13 min, 17 min, and 26 min respectively. This experiment was successfully able to record the distortions present due to GSM voice channel previously discussed in section III. Further, we implemented our adaptive framework in a simulation program on Matlab using the resulting data from the real-world experiments conducted before.

Results

The demodulation was performed on the acoustic symbols recorded at the Receiver Unit. The received data files were compared against the original file to calculate the total number of errors that occurred during the transmission over GSM voice channel. We observed error fluctuations in all the three received data files. However, this fluctuation was seen to be directly proportional to the data rate of the modulation scheme employed for experimentation. Based on this we evaluated the percentage of error-free frames for various frame size$M$ (see Figure 5.6). It can be clearly observed from this that as the frame size M increases, the percentage of error-free frames decreases. However, this decrease is slowly varying for Modulator Model III as compared to

Modulator Model I as the former is more robust to error.

Table 5.2: *Modulator Model Parameters*

System	L (sample length)	$\triangle$f (Hz)	Data rate (bps)	Symbol Error Rate (SER) %
Modulator model I	31	260	1550	9.8
Modulator model II	40	200	1200	4.75
Modulator model III	62	130	774	2.9

Table 5.3: *Performance Of Modulator Models Compared With Proposed Framework*

System	Raw Data Rate (Draw) (in bps)	Effective Data Rate (Deff) (in bps)
Modulator Model I	1550	1096
Modulator Model II	1200	869
Modulator Model III	774	591
Modulator Model I + MFEC	1550	1124
Modulator Model II + MFEC	1200	915
Modulator Model III+ MFEC	774	666
Adaptive Modulator Selection + MFEC	1509	1110

Performance of Proposed Framework: Table IV summarizes the results of our simulation. If each of the three Modulators was employed individually for data transmission without any adaptive framework, the effective data rate would be lower than the case in which the MFEC algorithm was used along with each of them. Although the adaptive framework (which includes both the adaptive selection of modulators and MFEC algorithm) has a slightly lower raw data rate compared to the highest D_{raw} of 1500 bps, it results into an data transmission which has the capability of correcting all the errors at the receiver despite of the varying channel distortions. The effective data rate of 1110 bps was significant enough as compared to previously proposed works because of the intelligence added at the Transmitter & Receiver units.

5.5 An Android prototype for Data over Voice

In this section, we describe the implementation and functionality of an Android application capable of sending data over voice to a remote location.

5.5.1 Design Decision

In this section, we discuss and showcase our capability to deploy a mobile based solution that can send data without the Internet over voice channel. We thought to build an smart-phone

Figure 5.7: *Main screen of 181 SOS application*

application for women safety that can send SOS signal even when victim does not have access to the Internet on their mobile device. Victims of VAW (Violence against women) may not have Internet at crime scene as many people have habit to use free Internet through Wi-Fi at home or work place instead of using mobile data services. This has motivated us to build a smart-phone application that can send SOS signal without the Internet to relevant authorities along with location data from GPS of victims' mobile device. Next design decision , we took regarding platform for mobile application. Among the mobile operating system, Android has the largest market share which is approximately 60% of total market [76]. Thus the reason, we chose to develop an Android-based deployable solution for data over voice.

5.5.2 Implementation

In this section, we explain feature and functionalities of an example SOS application for women safety. The current implementation described here is for Android, though similar applications can be developed using the same technology for other mobile platforms including iOS.

Figure 5.7 shows the main screen of SOS application. It has following features and functionality:

- Registration: It is the first step where a user at the time of installing the application on the phone submits their detail to server which can be helpful to provide them assistance at the time of danger. On the left side of Figure 5.8 shows the screen when the user presses the register button on the main screen. A message appears which describes the importance of registration process and further instruction on how to proceed with registration. On the right side of Figure 5.8 shows the various details that users are asked to submit to 181 authorities by filling a form on their SOS application. Once the user fills up the details and presses the button at the bottom of the form, their details get uploaded on server. This process of uploading the details uses the Internet for uploading the data.
- Complaint Lodging: One of the most frequent complaints from women are about obscene

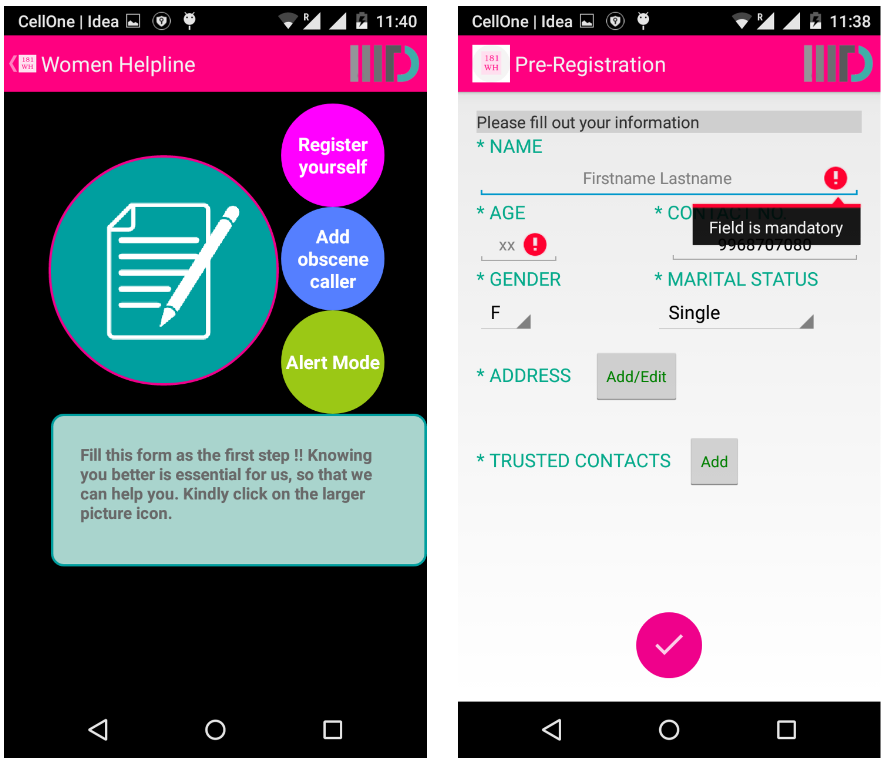


Figure 5.8: *Registration screen of SOS application*

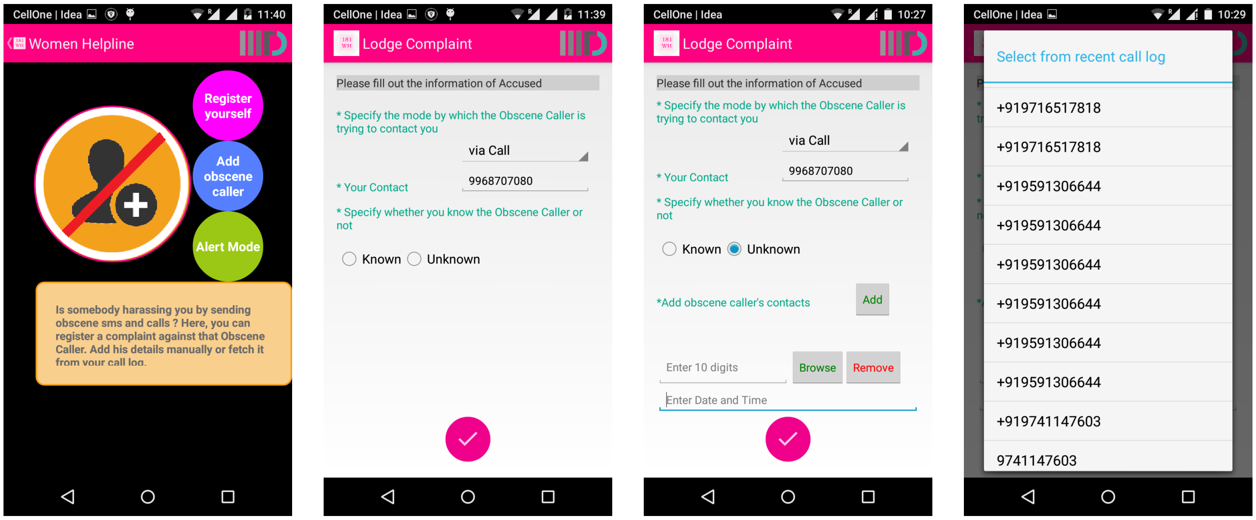


Figure 5.9: *Lodging of Complaint for an Obscene call by selecting it from the call log that is read by SOS application*

calls and messages. The second features lets its user to register a complaint about such incidents to relevant authority from the application itself. Application allows user to specify whether the complaint is about SMS or a call. User can specify the mobile number of accused by manually or selecting a number from its call log. Figure 5.9 shows a sequence of interaction between a user registering for an obscene call.

- SOS Alert (with/without the Internet): This feature is designed to send user location to relevant authorities while encountering safety issues. The location data transfer from mobile device to the server can happen in two ways i.e, either using Voice call or Internet Services. A user can save this setting and application will remember the mode to transfer the data. The designed application can be used in two ways to Signal an SOS message to relevant authorities. The user can either press Red button after opening the application as shown in Figure 5.10. However, in certain situation users(or victim) may not have time to unlock the phone to open the application and then press the SOS button. For such situation, the application provides an alternate way to signal SOS. A user can repeatedly (5-times) press the power button (i.e. Hard button to switch on-off the mobile) to signal the SOS to relevant authorities. Figure 5.10 shows a sequence of user interaction with mobile to activate emergency alerts.

  While designing this feature, a difficulty to implement data transfer over voice is the limited support from API offered from Android to generate custom audio tones and send them directly over the GSM uplink. An indirect way to achieve this is to store required audio tones of desired frequencies and play them over the phone that will be simultaneously capture by the microphone to send over the uplink. However, the downfall of this approach is that microphone capture all other noise along with the audio tone played over the speakers. Currently, android supports only the generation of DTMF tones that too only at the time of call initiation using "*android.intent.action.CALL*" API. These audio tones are used as

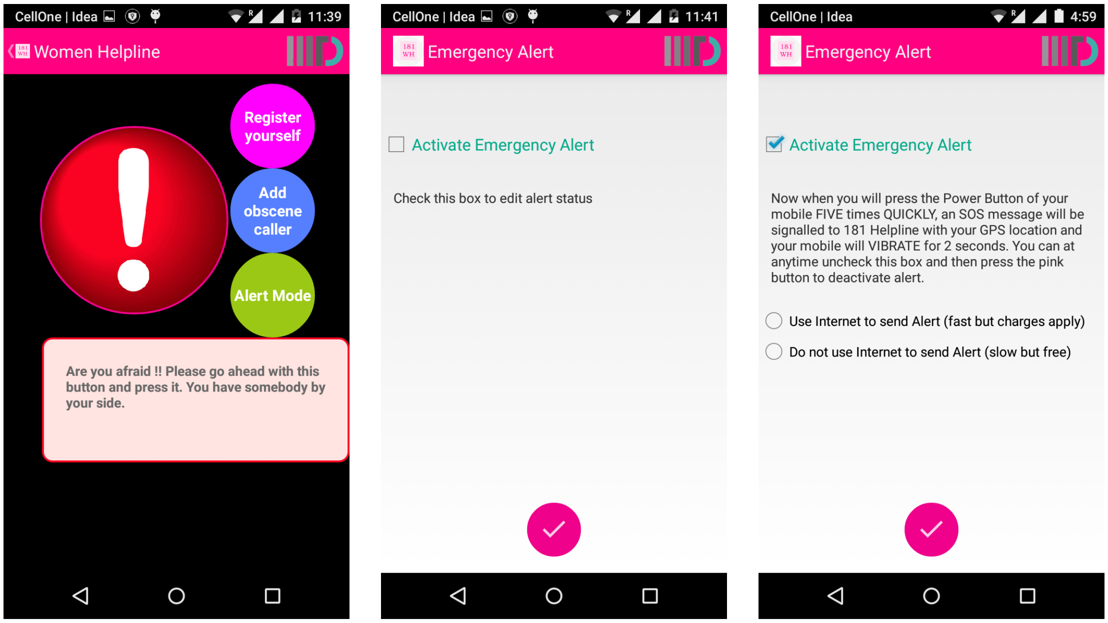


Figure 5.10: *Selecting SOS alert to be send via voice or the Internet*

acoustic symbols that are decoded at the telephony server using DTMF decoders.

5.6 Summary

In this Chapter, we explored the challenges of achieving error-free data transmission over the GSM voice channel. We investigated the potential use of the underutilized reverse call link of the same call session and proposed an adaptive framework that improved the error rate while transmitting the data. Extensive experiments have displayed the error fluctuations existing over the received data that cannot be corrected using the conventional error-correcting methods. Thus, the use of the reverse call link for sending Error metric as feedback improved the effective data rate performance system. We have also developed and designed the Android implementation of Data over Voice for women in distress.

We will also like to bring the known limitation of this work. Proposed a mechanism for sending parity at a delay is not suitable for transmission of real-time traffic such as audio and video. Simulation of the adaptive framework did not account for latency and synchronization issues of the real world. In the current implementation of MFEC only benefits through error-free frames are achieved. However, we observed the presence of several data frames with few errors that could have been used to reduce the overhead of error-correcting codes but requires the development of variable length error control codes.

Chapter 6

Conclusion

In this dissertation, we have provided methods to evaluate and improve the performance of different mobile ICT systems. We discussed three forms of mobile ICT, i.e., 1) Helpline Systems, 2) Automated Voice Applications, and 3) Smartphone Application. Here, we summarize our contribution towards these mobile ICT systems and direction for future research in this domain.

- Helpline System: Through our field deployment at 181- Chief Ministers Helpline, we identified that long waiting time in the telephonic queue has resulted in calls where caller hang-up the call before they were answered by the call executive of the helpline. We reviewed the literature around waiting caller and found the use of temporal metaphors for helpline with limited call executive could be a good solution. We found two concrete research gaps in literature:
  - 1) Need for an objective and quantitative method for evaluation of temporal metaphors.
  - 2) Comparison of metaphors designs to build effective metaphors for helpline caller.

  Towards this, we propose and demonstrated the use of Survival Analysis as a statistical method for objective and quantitative assessment of temporal metaphors' design. To the best of our knowledge, Survival Analysis is the first objective method proposed for evaluation of temporal metaphors. We also presented the findings of a real-world study conducted over helpline with distress callers. The performance difference between baseline system and proposed design was statistically significant. The comparison of different designs forms a good basis to design effective temporal metaphors for other helpline systems.
- Automated Voice Application: We deployed Automated Voice application system at IIIT-D to provide information to admission applicant in an automated manner. Data collected through this deployment suggests that a good amount of time of callers is wasted in navigating the menu of a voice application. Through a literature survey, we established that navigation is still an open problem in the area of voice-based system design. We propose the use of data-driven mechanisms which can improve the existing navigation technique by mining knowledge out of system usage data. We demonstrated the use of data-driven methods to build adaptive interfaces for voice menu that can reduce navigation time by predicting caller information need and arranging menu options in a particular order to satisfy that need as early as possible. We provided findings of comparing designs and algorithms for adaptive interfaces through real world study and the results were statistically

significant. We also presented the design and development of IVR testing tool named as Maareech.

- Smartphone application: We presented the findings of rural deployment where due to unreliable data connection has either delayed the data synchronization from field to centralized server MVoice or used SMS to send the data to MVoice. The literature survey validated the problems exists in several parts of developing regions and researchers have proposed the reliable data transmission through the use of voice channel . We identified the potential area of improvements over existing methods to transfer data through the voice channel. The proposed framework improves the data transfer rate by minimizing the overhead of parity bit in data transmission, use of the unutilized reverse channel, and adaptive data transfer mechanism. We conducted a real-world experiment to collect performance data of adaptive transfer mechanism. Calculations done over collected data shows the performance improvement of proposed framework. Based on this mechanism, we designed and developed an Android application for sending SOS signal from users mobile without the use of the Internet.

6.1 Future Work

Our work provides answers to many challenges related to developing and deploying a mobile ICT. However, several extensions of this research deserve further consideration that is listed below:

- For Helpline system, we developed a statistical method to assess the performance of temporal metaphor. We believe in future, several other methods for time and duration analysis can be compared against the proposed method in this dissertation. In future, method to accurately estimate expected waiting time can also be investigated. Future research could also extend it to study how an objective method like survival analysis correlate with subjective methods proposed in the literature. Further, the real world study conducted for temporal metaphors had mostly women users. A comparative study with the different type of users and scenarios can enable to comment better on the generalizability of proposed method.
- We studied the adaptive interfaces for voice application that increases IVR usability by reducing navigation time for callers but we find adaptive interfaces may be confusing to some repeated callers. Future work in this direction to develop adaptive interfaces for repeated callers can further increase the applicability of such systems.

  Also, in our studies, users were mainly from urban areas and were well educated. We believe a similar study with rural and illiterate population can open up new challenges for automated voice application system.
- We try to mitigate the problem of data connectivity through a multimodal system and acoustic modems. We believe future research can combine acoustic modem with multimodal connectivity into a single system to build the novel solutions for many rural applications like health, education, and banking. The work can also be expanded as Android middle-ware library.

Mobile phone based ICT has shown promising results for developing regions and resource-limited settings. With improving infrastructure and growing capabilities of the mobile phones, there will be a further boost to its usage as our gateways to human knowledge.

List of Publications

Published/Accepted

1. Konstantinos Kazakos, **Siddhartha Asthana**, Madeline Balaam, Mona Duggal, Amey Holden, Limalemla Jamir, Nanda Kishore Kannuri, Saurabh Kumar, Amarendar Reddy Manindla, Subhashini Arcot Manikam, GVS Murthy, Papreen Nahar, Peter Phillimore, Shreyaswi Sathyanath, Pushpendra Singh, Meenu Singh, Pete Wright, Deepika Yadav, Patrick Olivier, "A Real-Time IVR Platform for Community Radio", *Proceedings of the 34th Annual ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'16)*, 2016. [**Not included in thesis**]
2. **Siddhartha Asthana**, Pushpendra Singh, Parul Gupta "Survival Analysis: Objective assessment of Wait Time in HCI", *Proceedings of the 33rd Annual ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'15)*, 2015.
3. **Siddhartha Asthana**, Pushpendra Singh "Data Driven Usability: A Case for Adaptive Interfaces in Voice Based Menu Systems"' *To appear in proceedings of the 15th New Zealand Conference on Human-Computer Interaction (CHINZ 2015), Hamilton, New Zealand*, 2015
4. **Siddhartha Asthana**, Pushpendra Singh "Maareech: Usability Testing Tool for Voice Response System using XML based User Models", *To appear in proceedings of the 17th International Conference on Human-Computer Interaction (HCII), LNCS (Springer), USA*, 2015.
5. **Siddhartha Asthana**, Pushpendra Singh, Shraddha Jain "Adaptive Framework for Data Transmission over GSM Voice Channel for Developing Regions", *To appear in proceedings of the 7th IFIP International Conference on New Technologies, Mobility and Security (NTMS'2015), IEEE Communication Society*, 2015.
6. **Siddhartha Asthana**, Pushpendra Singh, "MVoice: A Mobile Based Generic ICT Tool", *Proceedings of the 6th International Conference on Information and Communication Technologies and Development (ICTD)*, 2013.**[Note]**
7. Vivek Srinivasan, Vibhore Vardhan, Snigdha Kar, **Siddhartha Asthana**, Rajendran Narayanan, Pushpendra Singh, Dipanjan Chakraborty, Amarjeet Singh, Aaditeshwar Seth, "Airavat: An Automated System to Increase Transparency and Accountability in Social Welfare Schemes in India", *Proceedings of the 6th International Conference on Information and Communication Technologies and Development (ICTD)*, 2013.**[Note]**
8. **Siddhartha Asthana**, Pushpendra Singh, Amarjeet Singh, "Assessing Designs of Interactive Voice Response Systems for Better Usability", *Proceedings of the 15th International Conference on Human-Computer Interaction (HCII), LNCS (Springer), USA*, 2013.
9. **Siddhartha Asthana**, Pushpendra Singh, Amarjeet Singh, "Exploring Adverse Effects of Adaptive Voice Menu", *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 2013.
10. **Siddhartha Asthana**, Pushpendra Singh, Amarjeet Singh, "Design and Evaluation of

Adaptive Interfaces for IVR Systems", *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 2013.

11. **Siddhartha Asthana**, Pushpendra Singh, Amarjeet Singh, "MockTell: Exploring challenges of user emulation in Interactive Voice Response Testing", *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE)*, 2013. **[Poster]**
12. **Siddhartha Asthana**, Pushpendra Singh, Amarjeet Singh, "Exploring the usability of Interactive Voice Response System's design", *Proceedings of the 3rd ACM Annual Symposium on Computing for Development (DEV)*, 2013. **[Poster]**
13. **Siddhartha Asthana**, Pushpendra Singh, Amarjeet Singh, "A Case for Adaptive Interface for Interactive Voice Response System", *Proceedings of the 3rd ACM Annual Symposium on Computing for Development (DEV)*, 2013.**[Poster]**
14. **Siddhartha Asthana**, Pushpendra Singh, Ponnurangam Kumaraguru, Amarjeet Singh, Vinayak Naik, "Tring! Tring! - An Exploration and Analysis of Interactive Voice Response Systems", *Proceedings of the 4th International Conference on Human Computer Interaction (IndiaHCI)*, 2012.

Technical Report

1. **Siddhartha Asthana**, Pushpendra Singh "Automated testing of Interactive Voice Response applications", *IIITD-TR-2014-001* , 2014.
2. **Siddhartha Asthana**, Pushpendra Singh, Amarjeet Singh, Vinayak Naik "Context-aware IVR: Challenges and Opportunities", *IIITD-TR-2011-008* , 2012.
3. **Siddhartha Asthana**, Ponnurangam Kumaraguru, Pushpendra Singh, Amarjeet Singh "Tring! Tring! usability study of a voice based information system", *IIITD-TR-2011-011* , 2012.

Bibliography

[1] Global mobile statistics 2011. Article, July 2011. http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats#subscribers. 1

[2] National Bureau of Statistics of China, Joint Statistical Publication by BRICS Countries 2011, April 2011. http://www.stats.gov.cn/english/statisticaldata/otherdata/brics2011/P020110412517148811510.pdf. 1

[3] Oded Agam. Voice over IP Testing-A Practical Guide. *RADCOM White Paper, Bitpipe Inc*, 2001. 4.5

[4] Sheetal Agarwal, Arun Kumar, Amit Anil Nanavati, and Nitendra Rajput. The world wide telecom web browser. In *Proc. of the 17th International Conference on World Wide Web*. ACM, 2008. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367686. URL http://doi.acm.org/10.1145/1367497.1367686. 3.2.1

[5] B.T. Ali, G. Baudoin, and O. Venard. Data transmission over mobile voice channel based on m-fsk modulation. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pages 4416–4421, April 2013. 5.3.1

[6] Lorraine G. Allan. The influence of the scalar timing model on human timing research. *Behavioural Processes*, 44(2):101 – 117, 1998. ISSN 0376-6357. doi: http://dx.doi.org/10.1016/S0376-6357(98)00043-6. URL http://www.sciencedirect.com/science/article/pii/S0376635798000436. 2.1.3

[7] LorraineG. Allan. The perception of time. *Perception & Psychophysics*, 26(5):340–354, 1979. ISSN 0031-5117. doi: 10.3758/BF03204158. URL http://dx.doi.org/10.3758/BF03204158. 2.1.3

[8] Gad Allon and Achal Bassamboo. The impact of delaying the delay announcements. *Operations research*, 59(5):1198–1210, 2011. 2.1.2

[9] Gad Allon, Achal Bassamboo, and Itai Gurvich. "We will be right with you": Managing customer expectations with vague promises and cheap talk. *Operations research*, 59(6): 1382–1394, 2011. 2.1.2

[10] Mor Armony, Nahum Shimkin, and Ward Whitt. The impact of delay announcements in many-server queues with abandonment. *Operations Research*, 57(1):66–81, 2009. 2.1.2

[11] B. Arons. Hyperspeech. In *Proc. of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*. ACM, 1993. ISBN 0-89791-575-5. doi: 10.1145/169059.

169517. URL http://doi.acm.org/10.1145/169059.169517. 3.2.2

[12] B. Arons. Speechskimmer: A system for interactively skimming recorded speech. *ACM Transactions Computer-Human Interaction*, 4(1), March 1997. ISSN 1073-0516. doi: 10.1145/244754.244758. URL http://doi.acm.org/10.1145/244754.244758. 3.2.2

[13] Siddhartha Asthana and Pushpendra Singh. Mvoice: a mobile based generic ict tool. In *Proceedings of the Sixth International Conference on Information and Communications Technologies and Development: Notes-Volume 2*, pages 5–8. ACM, 2013. 4.2

[14] Siddhartha Asthana, Pushpendra Singh, Ponnurangam Kumaraguru, Amarjeet Singh, and Vinayak Naik. Tring! tring!-an exploration and analysis of interactive voice response systems. *4th International Conference on Human Computer Interaction*, 2012. 4.2

[15] Siddhartha Asthana, Pushpendra Singh, and Amarjeet Singh. Design and evaluation of adaptive interfaces for ivr systems. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013. ISBN 978-1-4503-1952-2. doi: 10.1145/2468356.2468663. URL http://doi.acm.org/10.1145/2468356.2468663. 1

[16] Siddhartha Asthana, Pushpendra Singh, and Amarjeet Singh. Exploring adverse effects of adaptive voice menu. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013. ISBN 978-1-4503-1952-2. doi: 10.1145/2468356.2468494. URL http://doi.acm.org/10.1145/2468356.2468494. 3.6.5

[17] Siddhartha Asthana, Pushpendra Singh, and Amarjeet Singh. Exploring the usability of interactive voice response system's design. In *Proceedings of the 3rd ACM Symposium on Computing for Development*, page 36. ACM, 2013. 4.2

[18] Etienne Barnard, Laurens Cloete, and Hina Patel. *Electronic Government: Second International Conference, EGOV 2003, Prague, Czech Republic, September 1-5, 2003. Proceedings*, chapter Language and Technology Literacy Barriers to Accessing Government Services, pages 37–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-45239-3. doi: 10.1007/10929179_6. URL http://dx.doi.org/10.1007/10929179_6. 3.2.1

[19] J Martin Bland and Douglas G Altman. The logrank test. *BMJ*, 328(7447):1073, 2004. 2.3.1, 2.3.5

[20] Florian Block and Hans Gellersen. The impact of cognitive load on the perception of time. In *Proc. of the 6th Nordic Conference on Human-Computer Interaction*, NordiCHI '10, pages 607–610. ACM, 2010. ISBN 978-1-60558-934-3. doi: 10.1145/1868914.1868985. URL http://doi.acm.org/10.1145/1868914.1868985. 2.1.3

[21] R.A. Block. Prospective and retrospective duration judgment: The role of information processing and memory. In *Time, Action and Cognition*, volume 66, pages 141–152. Springer, 1992. ISBN 978-90-481-4166-1. doi: 10.1007/978-94-017-3536-0_16. URL http://dx.doi.org/10.1007/978-94-017-3536-0_16. 2.1.3

[22] Richard A. Block. *Models of psychological time*. Lawrence Erlbaum, 1990. 3

[23] Richard A Block. *Cognitive models of psychological time*. Psychology Press, 2014. 2.1.3

[24] Richard A. Block, Peter A. Hancock, and Dan Zakay. How cognitive load affects duration judgments: A meta-analytic review. *Acta Psychologica*, 2010. ISSN 0001-6918. doi: http://dx.doi.org/10.1016/j.actpsy.2010.03.006. URL http://www.sciencedirect.com/science/article/pii/S0001691810000594. 2.1.3, 2.3

[25] RichardA. Block and Dan Zakay. Prospective and retrospective duration judgments: A meta-analytic review. *Psychonomic Bulletin & Review*, pages 184–197, 1997. ISSN 1069-9384. doi: 10.3758/BF03209393. URL http://dx.doi.org/10.3758/BF03209393. 2.1.3

[26] M. Boloursaz, AH. Hadavi, R. Kazemi, and F. Behnia. A data modem for gsm adaptive multi rate voice channel. In *East-West Design Test Symposium, 2013*, pages 1–4, Sept 2013. doi: 10.1109/EWDTS.2013.6673152. 5.3.1

[27] M. Boloursaz, R. Kazemi, D. Nashtaali, M. Nasiri, and F. Behnia. Secure data over gsm based on algebraic codebooks. In *East-West Design Test Symposium, 2013*, pages 1–4, Sept 2013. doi: 10.1109/EWDTS.2013.6673148. 5.3.1

[28] Anna Bouch, Allan Kuchinsky, and Nina Bhatti. Quality is in the eye of the beholder: Meeting users' requirements for internet quality of service. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 297–304. ACM, 2000. ISBN 1-58113-216-6. doi: 10.1145/332040.332447. URL http://doi.acm.org/10.1145/332040.332447. 2.1.3

[29] Russell J Branaghan and Christopher A Sanchez. Feedback preferences and impressions of waiting. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 51(4):528–538, 2009. 2.1.3, 2

[30] Business Wire. Global contact center market to exceed usd 9.7 billion by 2019, according to technavio. http://www.businesswire.com/news/home/20160329005128/en/Global-Contact-Center-Market-Exceed-USD-9.7. 2

[31] T. Chmayssani and G. Baudoin. Data transmission over voice dedicated channels using digital modulations. In *Radioelektronika, 2008 18th International Conference*, pages 1–4, April 2008. doi: 10.1109/RADIOELEK.2008.4542682. 5.3.1

[32] Russell M. Church and Hilary A. Broadbent. Alternative representations of time, number, and rate. *Cognition*, 37(12):55 – 81, 1990. ISSN 0010-0277. doi: http://dx.doi.org/10.1016/0010-0277(90)90018-F. URL http://www.sciencedirect.com/science/article/pii/001002779090018F. Special Issue Animal Cognition. 2.1.3

[33] Destination CRM. Ivr market to reach $2.78 billion by 2017. http://www.destinationcrm.com/Articles/CRM-News/CRM-Featured-News/IVR-Market-to-Reach-\$2.78-Billion-by-2017-78870.aspx. 3

[34] Aditya Dhananjay, Ashlesh Sharma, Michael Paik, Jay Chen, Trishank Karthik Kuppusamy, Jinyang Li, and Lakshminarayanan Subramanian. Hermes: Data transmission over unknown voice channels. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 113–124, New York,

NY, USA, 2010. ACM. ISBN 978-1-4503-0181-7. doi: 10.1145/1859995.1860010. 5.3.1, 5.4.1, 5.4.3

[35] Wieland Eckert, Esther Levin, and Roberto Pieraccini. User modeling for spoken dialogue system evaluation. In *Proc. IEEE ASR Workshop*, pages 80–87, 1997. 4.6

[36] Bradley Efron. Logistic regression, survival analysis, and the kaplan-meier curve. *Journal of the American Statistical Association*, 83(402):414–425, 1988. 2.4.5

[37] Paul D. Feigin. Analysis of customer patience in a bank call center. Technical report, The Technion, Haifa, Israel, 2006. 2.1.2

[38] Peter Fröhlich. Dealing with system response times in interactive speech applications. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1379–1382. ACM, 2005. ISBN 1-59593-002-7. doi: 10.1145/1056808.1056921. URL `http://doi.acm.org/10.1145/1056808.1056921`. 2.1.3, 2

[39] Daniel Gaonac'h and Pascale Larigauderie. *Mémoire et fonctionnement cognitif: la mémoire de travail*. Armand Colin, 2000. 2.1.3

[40] GRAMVAANI. Jharkhand Mobile Radio. Website, 2012. `http://www.gramvaani.org/?page_id=343`. 3.2.1

[41] Simon Grondin. Timing and time perception: a review of recent behavioral and neuroscience findings and theoretical directions. *Attention, Perception, & Psychophysics*, 72 (3):561–582, 2010. 2.1.3

[42] Parul Gupta, Pushpendra Singh, and Vikram Goyal. Women safety and technology: analysis of women distress helpline for efficient technological intervention. 2015. 2

[43] Chris Harrison, Brian Amento, Stacey Kuznetsov, and Robert Bell. Rethinking the progress bar. In *Proc. of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 115–118. ACM, 2007. ISBN 978-1-59593-679-0. doi: 10.1145/1294211.1294231. URL `http://doi.acm.org/10.1145/1294211.1294231`. 2.1.2, 2.1.3

[44] Chris Harrison, Zhiquan Yeo, and Scott E. Hudson. Faster progress bars: Manipulating perceived duration with visual augmentations. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1545–1548. ACM, 2010. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753556. URL `http://doi.acm.org/10.1145/1753326.1753556`. 2.1.2, 2.1.3

[45] Sandra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, Advances in Psychology. North-Holland, 1988. doi: http://dx.doi.org/10.1016/S0166-4115(08)62386-9. URL `http://www.sciencedirect.com/science/article/pii/S0166411508623869`. 2.1.3

[46] Thomas Hornstein. Telephone Voice Interfaces on the Cheap. In *Union Bank of Switzerland, UVK Informatik*, pages 134–146, Zurich, 1994. 3.2.2

[47] Christophe Hurter, Benjamin R. Cowan, Audrey Girouard, and Nathalie Henry Riche. Active Progress Bar: Aiding the Switch to Temporary Activities. In *Proc. of the 26th*

Annual BCS Interaction Specialist Group Conference on People and Computers, pages 99–108. British Computer Society, 2012. URL http://dl.acm.org/citation.cfm?id=2377916.2377928. 2.1.3

[48] Zhezhen Jin, DY Lin, LJ Wei, and Zhiliang Ying. Rank-based inference for the accelerated failure time model. *Biometrika*, 90(2):341–353, 2003. 2.3.6

[49] Anirudha Joshi, Mandar Rane, Debjani Roy, Shweta Sali, Neha Bharshankar, N. Kumarasamy, Sanjay Pujari, Davidson Solomon, H. Diamond Sharma, D. G. Saple, Romain Rutten, Aakash Ganju, and Joris Van Dam. *Design Opportunities for Supporting Treatment of People Living with HIV / AIDS in India*, pages 315–332. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-23771-3. doi: 10.1007/978-3-642-23771-3_24. URL http://dx.doi.org/10.1007/978-3-642-23771-3_24. 5, 5.2.3

[50] Oualid Jouini and Alex Roubos. On multiple priority multi-server queues with impatience. *Journal of the Operational Research Society*, 2013. 2.1.1

[51] Oualid Jouini, Zeynep Akin, and Yves Dallery. Call centers with delay information: Models and insights. *Manufacturing & Service Operations Management*, pages 534–548, 2011. doi: 10.1287/msom.1110.0339. 2

[52] Oualid Jouini, Ger Koole, and Alex Roubos. Performance indicators for call centers with impatient customers. *IIE Transactions*, 45(3):341–354, 2013. doi: 10.1080/0740817X.2012.712241. URL http://www.tandfonline.com/doi/abs/10.1080/0740817X.2012.712241. 2.1.1

[53] Konstantinos Kazakos, Siddhartha Asthana, Madeline Balaam, Mona Duggal, Amey Holden, Limalemla Jamir, Nanda Kishore Kannuri, Saurabh Kumar, Amarendar Reddy Manindla, Subhashini Arcot Manikam, GVS Murthy, Papreen Nahar, Peter Phillimore, Shreyaswi Sathyanath, Pushpendra Singh, Meenu Singh, Pete Wright, Deepika Yadav, and Patrick Olivier. A real-time ivr platform for community radio. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 343–354, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858585. URL http://doi.acm.org/10.1145/2858036.2858585. 1.1

[54] Polyna Khudyakov, Paul D Feigin, and Avishai Mandelbaum. Designing a call center with an IVR (interactive voice response). *Queueing Systems*, 66(3):215–237, 2010. 2.1.1

[55] David G Kleinbaum and Mitchel Klein. *Survival analysis*. Springer, 1996. 2.3, 2.3.1, 2.3.1, 2.3.1, 2.3.5, 2.3.6, 2.4.5

[56] Aaron Koehl and Haining Wang. m. site: efficient content adaptation for mobile devices. In *Middleware 2012*. Springer, 2012. 3.6.1

[57] Zahir Koradia, Goutham Mannava, Aravindh Raman, Gaurav Aggarwal, Vinay Ribeiro, Aaditeshwar Seth, Sebastian Ardon, Anirban Mahanti, and Sipat Triukose. First impressions on the state of cellular data connectivity in india. In *Proceedings of the 4th Annual Symposium on Computing for Development*, ACM DEV-4 '13, pages 3:1–3:10, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2558-5. doi: 10.1145/2537052.2537064. 1.1.3, 5, 5.3.1

[58] Philip Kortum, S. Camille Peres, and Kurt Stallmann. Extensible auditory progress bar design: Performance and aesthetics. *Int'l Journal of Human-Computer Interaction*, 27(9): 864–884, 2011. doi: 10.1080/10447318.2011.555310. URL `http://dx.doi.org/10.1080/10447318.2011.555310`. 2.1.2, 2.1.3, 2.3

[59] Bojan Kotnik, Zdenko Mezgec, Janja Svecko, and Amor Chowdhury. Data transmission over {GSM} voice channel using digital modulation technique based on autoregressive modeling of speech production. *Digital Signal Processing*, 19(4):612 – 627, 2009. ISSN 1051-2004. 5.3.1

[60] C.K. LaDue, V.V. Sapozhnykov, and K.S. Fienberg. A data modem for gsm voice channel. *Vehicular Technology, IEEE Transactions on*, 57(4):2205–2218, July 2008. ISSN 0018-9545. doi: 10.1109/TVT.2007.912322. 5.3.1

[61] Carine Lallemand and Guillaume Gronier. Enhancing User eXperience During Waiting Time in HCI: Contributions of Cognitive Psychology. In *Proc. of the Designing Interactive Systems Conference*, DIS '12, pages 751–760. ACM, 2012. ISBN 978-1-4503-1210-3. doi: 10.1145/2317956.2318069. URL `http://doi.acm.org/10.1145/2317956.2318069`. 2.1.2, 2.1.3

[62] Martha Larson, Nitendra Rajput, Abhigyan Singh, and Saurabh Srivastava. I want to be sachin tendulkar!: A spoken english cricket game for rural students. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 1353–1364, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1331-5. doi: 10.1145/2441776.2441928. URL `http://doi.acm.org/10.1145/2441776.2441928`. 1

[63] Elisa T Lee and John Wang. *Statistical methods for survival data analysis*, volume 476. John Wiley & Sons, 2003. 2.4.5

[64] Kwan Min Lee and Jennifer C. Lai. Speech Versus Touch: A Comparative Study of the Use of Speech and DTMF Keypad for Navigation. *International Journal of Human Computer Interaction*, 19(3):343–360, 2005. URL `http://dx.doi.org/10.1207/s15327590ijhc1903_4`. 3.2.2

[65] Helga Lejeune. Switching or gating? the attentional challenge in cognitive models of psychological time. *Behavioural Processes*, 44(2):127 – 145, 1998. ISSN 0376-6357. doi: http://dx.doi.org/10.1016/S0376-6357(98)00045-X. URL `http://www.sciencedirect.com/science/article/pii/S037663579800045X`. 2.1.3

[66] Kevin A. Li, Patrick Baudisch, and Ken Hinckley. Blindsight: eyes-free access to mobile phones. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1389–1398, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: http://doi.acm.org/10.1145/1357054.1357273. URL `http://doi.acm.org/10.1145/1357054.1357273`. 3.2.2

[67] David H Maister. *The psychology of waiting lines*. Harvard Business School, 1984. 2.4.1, 11

[68] Avishai Mandelbaum and Sergey Zeltyn. Data-stories about (im) patient customers in tele-queues. *Queueing Systems*, pages 1–32, 2013. 2.1.1

[69] Mckinsey. Offline and falling behind: Barriers to internet adoption. http://www.mckinsey.com/industries/high-tech/our-insights/offline-and-falling-behind-barriers-to-internet-adoption, September 2014. 5.3.1

[70] Zdenko Mezgec, Amor Chowdhury, Bojan Kotnik, and Rajko Svečko. Implementation of pccd-ofdm-ask robust data transmission over gsm speech channel. *Informatica*, 20(1): 51–78, January 2009. ISSN 0868-4952. 5.3.1

[71] ShridharMubaraq Mishra, John Hwang, Dick Filippini, Reza Moazzami, Lakshminarayanan Subramanian, and Tom Du. Economic analysis of networking technologies for rural developing regions. In Xiaotie Deng and Yinyu Ye, editors, *Internet and Network Economics*, volume 3828 of *Lecture Notes in Computer Science*, pages 184–194. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-30900-0. doi: 10.1007/11600930_19. 5.3.1

[72] mobiThinking. Internet world stats. Article, July 2011. http://www.internetworldstats.com/top20.htm. 1

[73] Preeti Mudliar, Jonathan Donner, and William Thies. Emergent practices around cgnet swara, voice forum for citizen journalism in rural india. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, ICTD '12, pages 159–168, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1045-1. doi: 10.1145/2160673.2160695. URL http://doi.acm.org/10.1145/2160673.2160695. 3.2.1

[74] Nira Munichor and Anat Rafaeli. Numbers or apologies? customer reactions to telephone waiting time fillers. *Journal of Applied Psychology*, 92(2):511, 2007. 2.1.2

[75] Fiona Fui-Hoon Nah. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163, 2004. 2

[76] NETMARKETSHARE. Mobile / tablet operating system market share. https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1, July 2016. 5.5.1

[77] Jennifer A Newberry, Swaminatha Mahadevan, Narendrasinh Gohil, Roma Jamshed, Jashvant Prajapati, G Ramana Rao, et al. Using an emergency response infrastructure to help women who experience gender-based violence in gujarat, india. *Bull World Health Organ*, 94(5):388–392, 2016. 2

[78] Sumaru Niida, Satoshi Uemura, Hajime Nakamura, and Etsuko Harada. Field study of a waiting-time filler delivery system. In *Proc. of the 13th Int'l Conference on Human Computer Interaction with Mobile Devices and Services*, pages 177–180. ACM, 2011. 2.1.2

[79] Neil Patel, Sheetal Agarwal, Nitendra Rajput, Amit Nanavati, Paresh Dave, and Tapan S. Parikh. A comparative study of speech and dialed input voice interfaces in rural India. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 51–54. ACM, 2009. ISBN 978-1-60558-246-7. doi: http://doi.acm.org/10.1145/1518701.1518709. URL http://doi.acm.org/10.1145/

1518701.1518709. 3.2.2

[80] Neil Patel, Deepti Chittamuru, Anupam Jain, Paresh Dave, and Tapan S. Parikh. Avaaj otalo: a field study of an interactive voice forum for small farmers in rural india. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 733–742, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: http://doi.acm.org/10.1145/1753326.1753434. URL http://doi.acm.org/10.1145/1753326.1753434. 1, 1.1, 2, 3.2.1

[81] SC Peres, P Kortum, and K Stallmann. Auditory progress bars: Preference, performance and aesthetics. In *Proc. of the International Conference on Auditory Display (ICAD2007)*, 2007. 2.1.2

[82] Saverio Perugini, Taylor J. Anderson, and William F. Moroney. A study of out-of-turn interaction in menu-based, IVR, voicemail systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 961–970, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: http://doi.acm.org/10.1145/1240624.1240770. URL http://doi.acm.org/10.1145/1240624.1240770. 3.2.2

[83] Richard Peto and Julian Peto. Asymptotically efficient rank invariant test procedures. *Journal of the Royal Statistical Society. Series A (General)*, pages 185–207, 1972. 2.3.5, 9

[84] A. Srinivasa Rao. 95 per cent people below poverty line in andhra pradesh. http://indiatoday.intoday.in/story/95-per-cent-below-poverty-line-andhra-pradesh/1/164651.html, December 2011. 5.2.1

[85] Paul Resnick and Robert A. Virzi. Skip and scan: cleaning up telephone interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '92, pages 419–426, New York, NY, USA, 1992. ACM. ISBN 0-89791-513-5. doi: http://doi.acm.org/10.1145/142750.142881. URL http://doi.acm.org/10.1145/142750.142881. 3.2.2, 3.2.3

[86] Sancho Salcedo-Sanz, Maurizio Naldi, Angel M Perez-Bellido, Jose A Portilla-Figueras, and Emilio G Ortíz-García. Evolutionary optimization of service times in interactive voice response systems. *IEEE Transactions on Evolutionary Computation*, 14(4), 2010. 3.6.1

[87] Nithya Sambasivan, Julie Weber, and Edward Cutrell. Designing a phone broadcasting system for urban sex workers in india. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 267–276, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: http://doi.acm.org/10.1145/1978942.1978980. URL http://doi.acm.org/10.1145/1978942.1978980. 3.2.1

[88] Jeff Sauro and James R. Lewis. Average task times in usability tests: What to report? In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2347–2350. ACM, 2010. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753679. URL http://doi.acm.org/10.1145/1753326.1753679. 2.3.5, 2.4.5

[89] Dominique L Scapin and JM Christian Bastien. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & Information technology*, pages

220–231, 1997. 2.1.2

[90] A Shahbazi, AH. Rezaie, A Sayadiyan, and S. Mosayyebpour. A novel speech-like symbol design for data transmission through gsm voice channel. In *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on*, pages 478–483, Dec 2009. doi: 10.1109/ISSPIT.2009.5407541. 5.3.1

[91] Asheesh Sharma, Manveen Kaur, Zahir Koradia, Rahul Nishant, Sameer Pandit, Aravindh Raman, and Aaditeshwar Seth. Revisiting the state of cellular data connectivity in india. In *Proceedings of the 2015 Annual Symposium on Computing for Development*, DEV '15, pages 149–157, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3490-7. doi: 10.1145/2830629.2830649. URL `http://doi.acm.org/10.1145/2830629.2830649`. 5, 5.3.1

[92] A. Sharma Grover, O. Stewart, and Lubensky. Designing interactive voice response (IVR) interfaces: localisation for low literacy users. In *Proc. of Computers and Advanced Technology in Education*, 2009. 3.2.2

[93] J. Sherwani, N. Ali, S. Mirza, A. Fatma, Y. Memon, M. Karim, R. Tongia, and R. Rosenfeld. Healthline: Speech-based access to health information by low-literate users. In *Information and Communication Technologies and Development, 2007. ICTD 2007. International Conference on*, pages 1 –9. IEEE, dec. 2007. doi: 10.1109/ICTD.2007.4937399. 1

[94] Abhishek Shrivastava and Anirudha Joshi. Effects of visuals, menu depths, and menu positions on ivr usage by non-tech savvy users. In *Proceedings of the India HCI 2014 Conference on Human Computer Interaction*, IndiaHCI '14, pages 35:35–35:44, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3218-7. doi: 10.1145/2676702.2676707. URL `http://doi.acm.org/10.1145/2676702.2676707`. 3.2.2

[95] Raj Srinivasan, Jérome Talim, and Jinting Wang. Performance analysis of a call center with interactive voice response units. *Top*, 12(1):91–110, 2004. 2.1.1

[96] Bernhard Suhm. *Human Factors and Voice Interactive Systems Second*, chapter IVR Usability Engineering Using Guidelines and Analyses of End-To-End Calls. Springer, 2nd edition, 2008. ISBN 978-0-387-25482-1. 3.2.2

[97] Govt. of India The Department of Agriculture & Cooperation (DAC), Ministry of Agriculture. Kisan Call Center. website. `http://agritech.tnau.ac.in/kisan/kisan.html`. 1.1

[98] TRAI. Press Release June11. Information Note to the Press, August 2011. `http://www.trai.gov.in/WriteReadData/trai/upload/PressReleases/835/Press%20Release%20June11.pdf`. 1

[99] Michel Treisman. Temporal discrimination and the indifference interval: Implications for a model of the "internal clock". *Psychological Monographs: General and Applied*, 77 (13):1, 1963. 2.1.3

[100] Jinting Wang and R Srinivasan. Staffing a call center with interactive voice response units and impatient calls. In *Service Operations and Logistics, and Informatics, 2008.*

IEEE/SOLI 2008. IEEE International Conference on, volume 1, pages 514–519. IEEE, 2008. 2.1.1

[101] Stephen B Wicker. Reed-solomon error control coding for rayleigh fading channels with feedback. *Vehicular Technology, IEEE Transactions on*, 41(2):124–133, 1992. 5.4.2

[102] wiki. Network simulator-2. Article, January 2012. `http://nsnam.isi.edu/nsnam/index.php/Main_Page`. 4.5

[103] Dan Zakay. Gating or switching? gating is a better model of prospective timing (a response to switching or gating? by lejeune). *Behavioural Processes*, 50(1):1 – 7, 2000. ISSN 0376-6357. doi: http://dx.doi.org/10.1016/S0376-6357(00)00086-3. URL `http://www.sciencedirect.com/science/article/pii/S0376635700000863`. 2.1.3

[104] Dan Zakay. Attention and duration judgment. *Psychologie Bulletin Francaise*, 50:65–79, 2005. 2.1.3

[105] Dan Zakay and Richard A Block. Temporal cognition. *Current Directions in Psychological Science*, pages 12–16, 1997. 2.1.3

[106] Ferdinand Rudolf Hendrikus Zijlstra. *Efficiency in Work Behaviour: A Design Approach for Modern Tools*. Delft University Press, 1993. ISBN 9789062759187. URL `http://books.google.co.in/books?id=kbSCAQAACAAJ`. 2.1.3