Optimizing the Design of IVR Systems, as a Special Case of Self-Services

M.Sc. Research Proposal

Nitzan Yuviler

Advisors: Prof. Avishai Mandelbaum
Prof. Haya Kaspi

The Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology
August 29, 2012
Contents

List of Acronyms ............................................. 3

1 Introduction .................................................. 4

2 Literature Review ........................................... 5

   2.1 Methodologies for evaluating IVRs............................ 5
   2.2 Designing and optimizing IVRs........................................... 6
   2.3 Modeling a Call Center with an IVR................................. 9
       2.3.1 "Performance Analysis of a Call Center with Interactive 
            Voice Response Units".................................................. 9
       2.3.2 "Designing a Call Center with an IVR"............................ 10
       2.3.3 "Analysis of Customer Patience in a Bank Call Center"............ 11
       2.3.4 "Robust Design and Control of IVR Systems in Call Centers".... 12

3 Problem definition and research objectives .................... 13

4 Methodology .................................................. 25

5 Data Source .................................................. 33
List of Acronyms

CDVC Care Delivery value chain
CSR Customer Service Representatives
HFE Human Factor Engineering
IVR Interactive Voice Response
PASTA Poisson Arrivals See Time Averages
QED Quality and Efficiency Driven
SEE Service Enterprise Engineering
1. Introduction

In today's economy, call centers play a very important role, as they serve as the main customer contact channel in many different enterprises, public or private, product-based or service-based [1]. Call centers are highly labor-intensive, some employing hundreds or even thousands of Customer Service Representatives (CSRs), also known as agents, to handle incoming calls. Typically, 60%-80% of the overall call center operating expenses derive from agents employment costs [1]. Reducing the number of agents handling calls, without degrading the service level, is thus of interest for all call centers. Enabling customers self-service is one of the basic means for doing so. As customers self-serve, the agents workload is being reduced, and less agents are required in order to maintain a certain service level. Interactive Voice Response (IVR) systems, also known as Voice Response Units (VRU), are one of the main self-service channels [12], along with Internet websites and designated smart-phone applications.

IVR systems, if properly designed, can increase customer satisfaction and loyalty, cut staffing costs and increase revenue by extending business hours and market reach [2]. Poorly designed IVR systems, on the other hand, will cause the opposite effect and lead to dissatisfied customers, increased call volume and might even increase agent turnover, as agents ought to serve frustrated customers [6].

A Purdue University study showed that more than 90% of US consumers form their image on a certain company based on their experience with its call center. Furthermore, more than 60% stopped using the products of a company in which they had a negative call center experience [6]. Since the IVR system is the “front gate” of most call centers, having an effective, efficient and customer friendly IVR system is extremely important.

Analyzing IVR transactions is important in order to assess IVR service quality and efficiency. For example, are customers satisfied with the IVR service or do they opt out for agent assistance, or perhaps, leave the IVR without getting any relevant information? Are all IVR capabilities actually being used? [12].

The goal of our proposed research is to model and analyze customers flow within the IVR. The optimization of IVR system design, management and performance can be achieved through system modeling and careful analysis of the data supporting the model.
In this research, we shall try to model the customer flow within the IVR using a stochastic model, and based on real data.

Customers’ transactions in the IVR consist of varying sequences of tasks. Fitting a stochastic model to the IVR process will help analyze the various IVR tasks and their role and effect on process duration and success. The designing of an IVR system must take into account different aspects such as: customer needs, company preferences and human factors [12]. We believe that a stochastic model of customers flow within the IVR will contribute to better designing and modeling IVR systems, thus supplementing existing research from other fields such as Human-Factor-Engineering.

This research proposal is arranged as follows: in Section 2 we present a brief literature review. In Section 3 we introduce the problem and our research objectives. Our proposed methodology for solving the problem is presented in Section 4. In Section 5 we present the data source on which we shall be relying on in our research.

2. Literature Review
2.1 Methodologies for evaluating IVRs.

Suhm and Peterson [21, 22] presented a comprehensive methodology for IVR usability evaluation and re-design. It is claimed there that the standard IVR usability tests and standard IVR reports are not sufficient to assess the true performance and usability of an IVR. Most of the reports being used by call center managers are based on measures related to IVR utilization, for example, the percentage of customers that left the IVR without seeking a live agent, where this percentage is interpreted as the success rate of the IVR service. After analyzing thousands of end-to-end calls, which included the interaction with the IVR and the customer-agents dialogs, it was discovered that although 30% of the customers completed their service in the IVR, only 1.6% of the customers actually got relevant service.

Suhm and Peterson [18] also presented a new measure for quantifying IVR usability and cost-effectiveness. This measure is defined as the agent time being saved by handling the call, or part of the call, in the IVR, compared to handling the call only by a live agent. It was also suggested that, by using User-Path Diagrams, one can identify usability problems, such as nodes in the IVR that are rarely visited, nodes with
high volume of abandonment and nodes with high volume of customers seeking live agent assistance. By analyzing end-to-end calls, one can also compare the categorical distribution of original call reasons, which is revealed through the customer-agent dialogs, to the call-type distribution that arose from the IVR logs. Differences between those two categorical distributions indicate that customers are not navigating correctly within the IVR. In order to identify menu navigation problems, Suhm and Peterson [19] suggested a chart that shows IVR options chosen by customers (IVR routing) as columns. Then, they break down these columns by the original call reasons, as revealed through the customer-agent dialog. In this chart, menu options that are chosen correctly are shown as columns that consist of just the matching reason, while columns with many different components indicate menu options that are frequently selected incorrectly. Identifying menu options which incur usability problems, and analyzing the menu wording, will infer the solutions for improving menu navigation. For example, the option "Leave Message", in Figure 1 below, might often be chosen incorrectly because it suggests an opportunity of speaking to an agent.

![Figure 1: Analysis of menu navigation accuracy. Columns represent menu selection, broken down by call reason [19].](image)

### 2.2 Designing and optimizing IVRs

The subject of improving and optimizing IVR design has been addressed from different aspects. One of the main issues that appears in Human-Factor-Engineering (HFE) research is the IVR architecture – mainly, comparing broad (shallow) designs and deep (narrow) designs.
Schumacher et al. [15] recommended that IVRs with command-like options are to be limited to four or less items in each menu. The authors claim that since all the output is auditory, broad designs, with more than four options per menu, place a heavy demand on the working memory and therefore should be avoided. Marics and Engelbeck [13] also recommend that menus are to be limited to four commands (not including global commands such as Help and Exit). Both papers are relying on Miller's work from 1956 which basically argued that the number of objects an average human can hold in working memory is 7 ± 2. Marics and Engelbeck [13] also stated that menu items should be ordered according to frequency of use, unless the menu choices have a distinct natural or functional order.

Commarford et al. [5], on the other hand, argued that broad designs do not overload the working memory. The user does not need to remember all the options in the menu but to hold the best option, compare it with the new one and then save the better option. Therefore, the user only needs to hold up to two options in the working memory. Creating a deep, divided design in order to limit the number of options in every menu can increase the complexity and create confusion because it might not fit the user's mental model. Commarford et al. [5] have conducted an experiment which showed that users who used broad IVR design performed tasks faster and with greater satisfaction than users who used deep IVR design. Huguenard et al. [9] mentioned that the top level menu options of a deep and narrow IVR design ought to be less semantically similar to the terminal options, as compared to the top level menu options of a broad IVR design. They examined two menu designs - a deep design with four levels of menus, each menu containing three options, and a broad design with two levels of menus, each menu containing nine options. The broad design was created by simply using the second and fourth-level menus from the deep design, hence, both structures has the same 81 terminal options. The authors show that the choice error rates were significantly higher in the deep menu design. Suhm, Freeman and Getty [17] compared the response rate (choosing a legal option from the menu), routing rate (being routed to a specific group of agents) and re-prompt rate (need to listen to the menu again) in two menus – one containing seven specific, well defined options (will be called “long menu”) and the other containing four broader, frequently selected options (will be called “short menu”). The two designs were
compared using thousands of live calls to a commercial customer service center. Calls entering the call center were routed to a simulation of a modified call flow and then were returned to the call center to complete the call handling (this procedure did not disturb the ongoing call center operation). From the recording of the entire call, the authors could infer the final call routing and determine whether the menu selection by the caller in the simulation was correct. They have found that the long menu can identify the reason for a call more efficiently than the short menu and that the re-prompt rate for the short menu was significantly higher than for the long menu. This result also challenges the belief, which relies on Miller (1956), that menus should contain fewer than 5 items.

More researches address the issue of optimizing IVR design by presenting algorithms to reduce the service times in the IVR. Salcedo-Sanz et al. [14] introduced an evolutionary algorithm to optimally design IVRs, based on Dandelion encoding. Specifically, the IVR menu is considered to be a service tree, where each announcement (sub menu) is a node and each service is a leaf. The algorithm assumes that the time spent in each announcement is linearly related to the number of options in each announcement, and that a customer listens to all the options in the announcement before making a selection. The algorithm aims to reduce the average time to reach a desired service. If M is the number of services, t_i is the time required to reach a certain service i and p_i is the probability that a customer will ask for service i, then the optimal design of the IVR will minimize the function

\[ f(T) = \sum_{i=1}^{M} t_i p_i. \]

The basic idea behind the algorithm is to associate the most frequently requested service with the shortest path. The problem is equivalent to assigning code words to a set of messages to be transmitted, such that the mean code word length is minimized. The suggested algorithm was tested in a real call center of an Italian mobile telecommunications company and in synthetic experiments. The results showed that the algorithm is able to improve the results of Huffman approach and obtain results which are very close to a lower bound for the problem, although it was not proved that the algorithm will bring to optimal results. The lower bound of the problem was derived using the noiseless coding theorem which states that “the average codeword length \( \bar{l} \), using an alphabet of k symbols, is always larger than the uncertainty measure
represented by the entropy of the system”, which, in this case means: \( \bar{T} \geq -\sum_{i=1}^{M} p_i \log_k(p_i) \).

Therefore, the average time to reach a desired service, for a k-ary tree with M services whose probabilities are \( p_1 \leq p_2 \leq \ldots \leq p_M \), maintain: \( \bar{T} = k d_{\text{ann}} \geq -k d_{\text{ann}} \sum_{i=1}^{M} p_i \log_k(p_i) \), where \( d_{\text{ann}} \) is the common duration of each option in the announcement. Huffman coding provides near optimal performance for the source coding problem. Its inefficiency, i.e., the difference between the average time to reach a desired service and the optimum (which is given by the entropy), when \( k \) is the number of options in each announcement is bounded by:

\[
\Delta T = \sigma_k + p_M \frac{k}{\ln k}, \quad \text{where} \quad \sigma_k = \log_k(k-1) + \log_k(\log_k(e)) - \log_k(e) + \frac{1}{k-1}.
\]

**Modeling a Call Center with an IVR.**

2.3.1 "Performance Analysis of a Call Center with Interactive Voice Response Units"

Srinivasan, Talim and Wang [16] used a Markovian model of a call center with an IVR to determine the number of trunk lines (N) and agents (S) required to meet a certain service level, which is defined by the probability of a wait for an agent (after the IVR service) and the probability that an arriving call will be blocked (all lines being busy). The model assumptions are as follows: The arrival rate is a Poisson process with constant rate \( \lambda \). If a call arrives to the call center and all trunk lines are occupied, the call is lost. Otherwise, the call spends some time in the IVR and then can either request an agent service with probability \( p \), or leave the system with probability \( 1-p \). If there is no available agent the call will join the queue. The IVR processing times are i.i.d exponential random variables with rate \( \theta \). The agent service times are i.i.d exponential random variables with rate \( \mu \). The model in [16] does not include abandonment from queue. The model can be addressed as a Flow-Controlled Jackson Network (FJN) and can be converted into a three-nodes closed Jackson network, which has a product form solution for its stationary distribution [21]. From the stationary probability \( P_k \) (for \( k \leq N \)), which means that there are exactly \( k \) calls in the system (in the IVR, in the queue or being
served by an agent), one can derive the probability that all the lines are occupied - $P_N$, which, according to the PASTA property is also the loss probability. The next step is finding the distribution function of the waiting time - $W(t) := P(T \leq t)$, $t \geq 0$. With $P_N$ and $W(\cdot)$ in hand, and specified values of $\lambda$, $\theta$, $p$, and $\mu$, one could find the number of trunk lines ($N$) and the number of agents ($S$), subject to $P_N \leq \varepsilon_1$ and $P(T \leq t) \geq \varepsilon_2$, for pre-defined $\varepsilon_1$, $\varepsilon_2$ and $\tau$.

![Figure 2: Call Center Model with N Trunk lines (VRU/IVR) and S Agents [16].](image)

2.3.2 "Designing a Call Center with an IVR"

Khudyakov, Feigin and Mandelbaum [11] used the model presented by Srinivasan, Talim and Wang [16] and performed asymptotic analysis in the QED regime. The asymptotic analysis provided QED approximations of frequently used performance measures (such as the waiting probability, the probability of a busy signal and the mean waiting time given waiting). Those approximations can help solve the staffing and trunking problem and be used to support the operation management of a call center. The approximations were validated against data taken from a large US bank call center [7]. The comparison of the approximate performance measures derived from the model to the real data gave satisfactory results. In many intervals, the theoretical values of the model were very close to the real data.
In addition, Khudyakov, Feigin and Mandelbaum [11] expanded the model presented by Srinivasan, Talim and Wang [16] and added customer impatience to the model. Srinivasan and Wang [21] also expanded their model [16] by adding abandonment from queue. The model assumption is that each call abandons the queue after an independently exponentially distributed amount of time. Naturally, when adding abandonment to the model, the number of required lines (N) and required agents (S) will be smaller. Furthermore, customers’ abandonment can decrease the probability of a busy signal and the waiting time.

2.3.3 "Analysis of Customer Patience in a Bank Call Center"

Feigin [8] presented a statistical analysis of customer patience in a call center of a US bank. The system considered was the queue of customers waiting to receive agent service, after completing the the IVR and the post-IVR phases. The post-IVR phase may consist of announcements, made by the system, which are designed to warn the customers of heavy load in the system. In times of heavy system load, the system announces an expected waiting time ($<1$ min, 1 min, 2 min, etc.) and recommends that the customer returns to the IVR. It was shown that less than 1.5\% of the customers who heard the announcement actually returned to the IVR.

One of the factors that may affect customers patience is how much time they have already invested in the call, namely, how much time they spent in the IVR and the post-IVR phases. Feigin compared the patience of customers who spent short time in the IVR (less than 100 sec) with the patience of customers who spent longer time in the IVR, by analyzing their survival probability. There was a clear separation between the two survival curves. Customers who have invested more time in the IVR were more patient while waiting for an agent service (see Figure 3). This fact has some consequences on the operational management of the system, especially when aiming to minimize abandonment. It suggests that the priority of customers entering the agent queue should be inversely related to the time they had already spent in the IVR and in the post-IVR phase. This means that customers who tend to be less patient in the IVR and post-IVR
phases will be given a higher priority in the agents queue and will wait less, while customers who tend to be more patient would be allowed to wait longer.

Histograms and QQ-plots of the IVR time distribution and the total invested time distribution (IVR + post-IVR), revealed that there is quite a good fit of the IVR time to the lognormal distribution but the fit of the total invested time to the lognormal distribution is not adequate.

![Figure 3: Survival curves for customers who invested short time (<100 sec) in the IVR and post-IVR stages compare to those who invested 100 sec or more [8].](image)

2.3.4 "Robust Design and Control of IVR Systems in Call Centers"

Behzad and Tezcan [3] proposed a model of a call center with flexible IVR system. They assumed that each call can be routed into one of two IVR designs. The specific design of each system was not considered. To quantify the differences between the two designs they focused on three performance measures: call resolution probability – proportion of calls completing the service in the IVR; Opt-out probability – proportion of calls that were transferred to a live agent; Abandonment probability – proportion of calls abandoning from the IVR. The IVR performance measures are effecting the call center staffing. For example, if the call resolution is high, then fewer agents are needed and vice versa. Therefore, IVR design must be synchronized with call center staffing. A two-stage
stochastic program was formulated in order to determine the optimal staffing level and the proportion of calls that should be routed to each IVR system, in a way that will minimize total costs, where the total costs consist of agents cost and abandonment cost.

The most interesting scenario is when one IVR system is more efficient than the other, meaning that the call resolution probability is higher and the abandonment probability is higher. This scenario is reasonable when there is one IVR system in which it is more difficult to reach the option of opting out to an agent. In this system, it is more likely that customers will find the answer to their problem while navigating through the IVR but may also lead to higher abandonment probability. In this scenario, there are two critical values of arrival rate \( \lambda_1, \lambda_2 \) which determine the routing to each one of the IVR systems. When the arrival rate is lower than \( \lambda_1 \), calls will be routed only to the less efficient IVR system. When the arrival rate exceeds \( \lambda_2 \), calls will be routed only to the more efficient IVR system. When the arrival rate is between \( \lambda_1 \) and \( \lambda_2 \), the routing should bring to a full utilization of the agents. The numerical experiments showed that using the presented model can bring to an average reduction of 8% in the total costs.

3. Problem definition and research objectives

We consider a call center with an IVR system. A customer usually starts the service in the IVR and then, if necessary, continues to an agent service. The call may either be connected immediately or queued. If the customer's waiting time in the queue exceeds the customer's patience she will abandon. After receiving a service from an agent, the customer might be transferred to a different agent, back to the IVR or simply exit the system. A customer service in the IVR usually constitutes several "events" such as identification and different queries.

Modeling and analyzing the customer flows within the IVR is important in order to optimize IVR design – discover usability problems, shorten service durations, raise the proportion of customers completing the service successfully, decrease the proportion of customers seeking an agent service and improve routing to the agents so that customer actions in the IVR will route them directly to the right group of agents.
One way to analyze customer flows within the IVR is via user-path diagrams as described by Suhm and Peterson [18]. Similarly, Kaplan and Porter [10] suggested a new cost measurement approach for the Healthcare system. Their approach consists of specifying the care delivery value chain (CDVC), which charts the main activities involved in a patient's care and then develops a detailed process maps for each activity in the CDVC. Those process maps present patients’ paths within the Healthcare system. They include all the resources (personnel, facilities and equipment) involved at each process, as well as the estimated time each resource spends with a patient at each step in the process. Similarly to the user-path diagrams of Suhm and Peterson and the process maps of Kaplan and Porter, Khudyakov et al. [12] constructed a flow chart describing the transition in the IVR, based on one day data from an Israeli commercial call center (see Figure 4). The chosen day was a typical Friday in August 2008. Only the first visit of each customer to the IVR was taken into account (as opposed to a returning visit). There were 50,085 such calls on that day. Each customer transaction in the IVR is referred as an “event”, and there are 14 different “event” types. The arrows in the chart denote transitions from one event to another. The numbers on the arrows state the number of transitions between the two relevant events, during the Friday being analyzed. A distribution of the number of visits to different events, the mean and standard deviation of the event durations are presented in Table 1 (which is also taken from Khudyakov et al. [12]).

The flow chart in Figure 4 and Table 1 reveals interesting issues that could be addressed. First, we observe that some events occur much more frequently (2000 times and up) than others (less than 50). There are several explanations for events with small number of visits: the services represented by these events have low demand; customers do not know about these services; customers prefer to receive these services in another way (via the website, face to face, with a live agent on the phone, etc); the way to reach these services in the IVR system is complicated [12]. Each one of these reasons may be a trigger to make a change in the IVR design. Second, almost 10% of the customers end their call directly after the "Identification" phase. Those customers leave the system without getting any useful information, because they are, probably, unable to identify
themselves. This is also a problem that should be considered in the IVR design. Moreover, when calculating the fraction of calls completing their service in the IVR, these calls cannot be defined as completed in the IVR only, but are actually uncompleted. Third, we note that a large number of customers reach event 5, which records transition to an agent. Some customers reach this event without visiting any other event (298 instances) or directly after the identification (4346 instances). As agents’ salaries being typically 2/3 of the overall call center operating expenses, one of the goals in the IVR design is to reduce the transitions to agents as much as possible, without degrading the service level and customer satisfaction.
Figure 4: Flow chart diagram of a call transitions in the IVR.
Table 1 shows that some of the events could be exponentially distributed, as their coefficient of variation (CV) is quite close to one (for example events 15 and 7), while others are certainly not.

IVR service durations and customer paths in the IVR depend on various factors, such as the customer type or priority, the service the customer requires, the customer familiarity with the system and so on. These factors also affect the durations of IVR events.

The following graphs demonstrate the dependency of the IVR service durations as well as the duration of specific IVR events (Identification, Query, etc.), upon various factors. The data was taken from an Israeli commercial call center, from April 2007 to June 2009. Figures 5 shows the distribution of service times for calls served only by the IVR, as opposed to calls that received agent service after the IVR, as shown in Figure 6. Figure 5 and 6 reveals that customers with different priorities have different service distributions. Figure 7 presents the distribution of service time provided by the IVR for customers requesting different types of service. It is clear that the IVR service time distribution is influenced not only by the customer types, but also by the service they are requesting.
Figure 5: Distribution of service time for calls served only by the IVR, for each type of customers.
Figure 6: Distribution of service time for all calls served by the IVR, for each type of customers.
Figures 7 presents the distribution of event durations for different types of customers. It depicts that different events have different distributions and that, within each event, different customer types have different distributions. It is also clear that the event durations, for most of the events, are not exponential. Some event durations, for example Query and Agent, have an empirical distribution which suggests a discrete or even a deterministic distribution.
Figure 8: Distribution of event duration – Identification, for different types of customers
Figure 9: Distribution of event duration – Query, for different types of customers
Figure 10: Distribution of event duration – Agent, for different types of customers
Figure 11: Distribution of event duration – Stock Market, for different types of customers
In our research, we seek to first create a stochastic model for customer flow within the IVR. In the second step, the model will be applied to support IVR design, in order to minimize opt-out rate, abandons from IVR and agent service times. So far, most research on operational design has been deterministic. A comparison between the deterministic and stochastic-based design is thus of interest, and perhaps new design methodologies would arise. An interesting research direction would be to incorporate known HFE recommendations regarding audio menu structures, such as a preference of one menu structure over the other in the sense of working memory load and user mental model. This could lead to a better, more comprehensive methodology to optimally design IVRs.

We shall then consider the following question: Could the stochastic model, which was applied to IVR design, could also be applied to different service systems, especially other self service systems, such as internet websites, smart-phones applications, self service stations at banks and supermarkets, etc.?

4. Methodology

Our objective in this research is to first create a stochastic model for customer flows within the IVR. We shall then use this model, combined with HFE principles and, possibly, known deterministic algorithms results, to seek optimal IVR designs, in the sense of minimizing abandons from the IVR, opt outs to an agent and agents service time.

Our analysis will be based on real data from a call center with an IVR (described in Section 5). As suggested in the deterministic algorithm of Salcedo-Sanz et al. [14], we shall start by considering the IVR menu as a tree, where services available by the IVR are represented by the tree leaves, and selection menus are represented by nodes. Therefore, in order to construct a stochastic model, we would like to distinguish between two types of IVR events:

- Navigation events - in which a customers only listen to a set of choices from which they should choose the desired one. These events will be represented by the tree nodes (not the leaves). We assume that, unlike the deterministic model, the time spent in each of these events is not constant and is not necessarily linearly related to the number of options in the menu. This is derived from the fact that a
customer can choose the desired option as soon as it appears and does not have to wait until all options are presented. Therefore, the time spent in the menu is influenced by the positions of the options in the menu and the probability that a specific option will be chosen. Another aspect that should be considered is the limitations of the human working memory. As mentioned before, on one hand, according to Miller (1956), one can store up to $7 \pm 2$ items in the working memory, hence menus with more options might place too heavy of a demand on the working memory. This can lead to confusion and force the customers to re-listen to the menu. On the other hand, deep menu structure may divide the menu in a way which does not fit the user's mental model and increase reaction time and error rate.

- Content events - in which a customer gets an actual service, for example, hearing her account balance or the stock market state. These events will be represented by the tree leaves.

The observation between those two types of events can be done by fitting a distribution to the IVR events. As mentioned before, it seems like some of the events have a discrete distribution, and thus will be good candidates for navigation events. We shall use the SEEStat program (see Section 5), in order to fit known distributions, or mixture of known distributions to the empirical ones. In our data, some events include sub events. In Figure 12 we see an example of how one event – "Change password" - actually consists of two sub events – "change password" and "selected password". One way to fit a distribution to an event is to construct it from the distribution of its sub-events. Using CV (Coefficient of Variation) calculations and comparing it to the CV of the event itself, one can identify the relationships between the sub-events, i.e. do they appear in a sequence, in parallel (i.e. each one of the events can be chosen with a certain probability). Analyzing the connection between the different events can be done via the same method, meaning, calculating the CV of a group of events and deducing whether they are connected in sequence, in parallel or in any other way.

A more realistic model will take into consideration the option of loops. From each node and each leaf in the IVR tree one can return to the main menu. The two other options which are available from any state in the IVR (nodes and leaves) is ending the
service in the IVR or opting out to receive service from an agent. Selecting these options may result from "good" or "bad" reasons. On one hand, customers may choose to end the service in the IVR (and leave the system or opt out) after receiving all the needed information they could receive from the IVR, this will be considered as a "good" reason. On the other hand, customers can leave the IVR without getting any useful information.

Figure 12: Distribution of the duration of the event "Change password" and the distribution of its sub events duration.

Our goal in this research is to reduce opt-out rate, abandonment rate from the IVR and agents’ service time. Identifying abandonments and opt outs which result from unsatisfactory IVR service is thus of interest. In order to do so we shall analyze customer paths within the IVR and will try to distinguish between paths that lead to a successful completion of the IVR service as opposed to paths that lead to unsatisfactory results such
as abandonment and opt out that could be avoided. Identifying abandonments could also be done using structural estimation. In this method one assumes a specific model to customer's behavior and then predicts the model parameters using the actual results as they appear in the data. SEEStat program enables us to construct flowcharts of customer paths within the IVR directly from the data itself. An example of such flowchart is presented in Figure 13. The green triangles represent starting points, the green dots represent ending points, the blue rectangles represent the IVR events and the numbers on the arrows are the number of the matching transitions that occurred during the period from which the data was taken (in Figure 13, one day).

![Figure 13: Customer flows within the IVR during 2.5.2008.](image)

When analyzing customer paths we will first sort the data according to different categories such as: customer types; customer call count for a specific period of time (e.g. first customer calls, second customer calls, etc., for a specific day); the call result – whether the call was completed in the IVR or transferred to an agent; and the service type the customer received from an agent. This approach relies on the work of Kaplan and Porter [10] and Bohmer [4] in the healthcare system. Kaplan and Porter, in
their suggestion for a cost measurement method in healthcare, mentioned that the first step that should be done, before defining the CDVC and developing the process maps, is dividing the patient population by their medical condition. After doing so, a different process map is to be developed for each patient type. Bohmer [4] also stated the importance of separating heterogeneous patient populations into subgroups (according to disease type, severity or risk of complication) and finding each subgroup's distinct pathway, as different patient subgroups have different paths of care.

After doing so, an interesting statistical analysis will be to find a correlation between IVR paths and IVR durations to service durations. We shall look at groups of calls that resulted in agent service in which all the parameters are the same and will investigate whether specific IVR paths result in lower agent service durations or whether there is a negative correlation between IVR durations and agent service durations.

We expect that all the above will help us in constructing a stochastic model for customer flows within the IVR, validating the model and solving the optimization problems of minimizing opt out rate, abandonment rates and agents service time, subject to the model assumptions and constrains.

A simple example of a formulation of such a problem:

Our goal will be to minimize the opt out rate and the abandonment rate. We shall make the following assumptions:

1. There are M available services in the IVR.
2. There are J customers passing through the IVR.
3. The IVR is constructed as a directed tree, no loops allowed. The services are the tree leaves.
4. The IVR menus are the tree nodes (not including the leaves). In each menu there is also an option to opt out to an agent service and an option to leave the system (hanging up without choosing any option, will be called abandonment). The opt out options and the abandonments will be represented as tree leaves.
5. If there are M services, the total number of menus (nodes, not including leaves) could vary from 1 (the root node leads directly to all the services - broad design) to M-1 (deep design). The number of leaves representing abandonments and the number of leaves representing opt out options is equal to the number of menus.
6. All the IVR services, menus, opt out options and the abandonments will be considered as the tree nodes where some are leaves (services, opt out options and abandonments) and some are not leaves (menus). The IVR services will be numbered by 1, 2, ..., M. The IVR menus will be numbered by M+1, M+2, ..., 2M-1, Where node number M+1 is the root nodes (first IVR menu). The opt out options will be numbered by 2M, ..., 3M-2. The leaves representing abandonments will be numbered by 3M-1, ..., 4M-3.

Definitions:

\[ X_{i,j} = \begin{cases} 
1 & \text{customer } j \text{ reached leaf } i \\
0 & \text{o.w.} 
\end{cases} \quad i = 1, \ldots, 4M-3 \\
\quad j = 1, \ldots, J 
\]

We will use the notation:

\[ S = \{1, \ldots, 4M-3\} \]
\[ A = \{1, \ldots, M\} \]
\[ B = \{M+1, \ldots, 2M-1\} \]
\[ C = \{2M, \ldots, 3M-2\} \]
\[ D = \{3M-1, \ldots, 4M-3\} \]

T(k) = Time spent in node k.

Decision Variables:

\[ Y_{i,k} = \begin{cases} 
1 & \text{node } k \text{ is a direct predecessor of node } i \\
0 & \text{o.w.} 
\end{cases} \quad i, k = 1, \ldots, 4M-3 
\]

Note: "direct predecessor" means that there is a direct transition from node k to node i. If node k is a direct predecessor of node i then node i is a direct successor of node k.

Index(i) = If node k is a direct predecessor of node i and of a total number of N nodes (not including leaves which represent abandonments) then index(i) is the specific position of node i in the menu which is being heard at node k (node i can be either a service, an option to opt out or a menu).
Objective function:

\[
\min \sum_{i \in C} \sum_{j=1}^{J} X_{i,j} \quad \Leftrightarrow \quad \min \sum_{i \in C} \sum_{j=1}^{J} X_{i,j} ;
\]

Minimizing the number of customers reaching the opt out or abandonment leaves.

Problem constrains:

\[
\sum_{i \in S} X_{i,j} = 1, \quad \forall i \in j ; \quad \text{Every customer reaches exactly one leaf.}
\]

\[
\sum_{i \in S} Y_{i,k} = 0, \quad \text{for } k \in S \setminus B ; \quad \text{The leaves are not predecessors of any other node.}
\]

\[
\sum_{i \in S} Y_{i,k} = 1, \quad \text{for } i \in A ; \quad \text{Each service is a direct successor of exactly one node.}
\]

\[
\sum_{i \in S} Y_{i,M+1} \geq 4 ; \quad \text{The root node has at least 4 direct successors (2 services, abandonment and opt out).}
\]

\[
\sum_{i \in S} Y_{i,M+1,k} = 0 ; \quad \text{The root node has no predecessors.}
\]

\[
\sum_{i \in S} Y_{i,k} \notin \{1,2,3\} , \quad \text{for } k \in B ;
\]

All the IVR menus (except the root node) should have at least 4 direct successors (2 services, abandonment and opt out) or none.

\[
\sum_{i \in S} Y_{i,k} = \min \left\{ 1, \sum_{i \in S} Y_{i,j} \right\} , \quad \text{for } i \in B ;
\]

Each IVR menu (except the root node) can have only one direct predecessor. However, if this menu is not a predecessor of any other node it could not be a successor of any other node (and is actually does not exist in the tree).

\[
Y_{i,j-(M-1)} = \min \left\{ 1, \sum_{i \in S} Y_{i,j-(M-1)} \right\} , \quad \text{for } i \in C
\]

\[
Y_{i,j-2(M-1)} = \min \left\{ 1, \sum_{i \in S} Y_{i,j-2(M-1)} \right\} , \quad \text{for } i \in D ;
\]

Each leaf which represents abandonment or opt out is associated to a specific menu. If this menu is not a predecessor of any other node (and therefore does not appear in the tree) then those leaves will not appear in the tree either.
\[ M + 2 \leq \sum_{n \in S} \sum_{k \in S} Y_{i,k} \leq 4M - 4; \]

The total number of nodes that can be reached from any other node is between \( M + 2 \) (broad design) and \( 4M - 4 \) (deep design).

\[
\text{Max\_index}(i) = \sum_{k \in S} \left[ Y_{i,k} \cdot \sum_{k \in S \setminus D} Y_{i,k} \right], \quad \forall i;
\]

If node \( i \) is a direct successor of node \( k \), then the maximal index that node \( i \) can get is the total number of direct successors to node \( k \), not including the abandonment leaves.

\[
\text{index}(i) = 0, \ldots, \text{Max\_index}(i), \quad \forall i;
\]

\[ T(k) = f \left( \sum_{i \in S} Y_{i,k}, \text{index}(i) \text{ (of all nodes } i \in S \setminus D \text{ for which } Y_{i,k} = 1) \right), \quad \text{for } k \in B; \]

\[ f(0) = 0; \]

The time spent in each node which is not a leaf (a node that represents a menu) is a function of the number of its direct successors (the number of options in this menu) and their specific position (the order of the options in the menu). If this node has no direct successors it is not included in the tree and the time spent there is 0. This function should be found theoretically or via our data and the SEEStat program.

The time spent in the services could be from some continuous distribution, for example:

\[ T(i) \sim \exp(\lambda_i) \quad \text{for } i \in A \]

This distribution could be found using our data and the SEEStat program.

The time spent in the leaves representing abandonments or opt outs is considered to be 0:

\[ T(i) = 0, \quad \text{for } i \in C, D \]

This problem representation, using integer programming, is only an example of one possible way to represent the problem, when assuming a finite number of customers and not considering the dynamic in time. If we will consider the case of a stream of customer arriving to the IVR then a dynamic programming is in need.
5. Data Source

Our data was provided by the Center for Service Enterprise Engineering (SEE). From the SEE center website¹: “The Center for Service Enterprise Engineering (SEE) was established in February 2007, within the Faculty of Industrial Engineering and Management, at the Technion, through the generous support of Hal and Inge Marcus. The goal of SEE is the development of engineering and scientific principles that support modeling, design and management of Service Enterprises, for example: financial services (banking, insurance); health services (hospitals, clinics); government and teleservices (telephone, internet). Presently, SEE’s main activity is designing, maintaining and analyzing an accessible repository of resources and data from telephone call-centers”.

The graphical user interface, developed at the SEE Lab, called SEEStat, provides real-time statistical analysis, spanning seconds-to-months resolutions, with typical few seconds response time. The SEEStat tool also enables statistical analysis of imported raw data. The SEEStat tool is available at:

http://seeserver.iem.technion.ac.il/see-terminal/.

In order to model an IVR, we shall analyze data from a call center with an IVR, using DataMOCCA and its user interface, SEEStat. We shall analyze data from a call center of an Israeli Bank. This database fits our needs since it covers more than two years data of daily IVR logs, including the customer type; the service required; how many calls the customer had already made that day; the different events the customer experienced in the IVR and their durations. The database includes also complete calls data and agents data.

Table 2 presents a numerical summary of the database, from April 2007 to June 2009, including all weekdays in that period. Table 3 and 4 present, for the same time period, a distribution of the average number of calls for each service type and each customer type respectively.

¹ http://ie.technion.ac.il/labs/serveng
<table>
<thead>
<tr>
<th>Table 2: Overall summary of calls.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td># Requesting agent service</td>
</tr>
<tr>
<td># Served only by IVR</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3: A distribution of the average number of calls for each service type, for customers requesting agent service after the IVR service.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Type</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>private</td>
</tr>
<tr>
<td>Securities</td>
</tr>
<tr>
<td>Internat</td>
</tr>
<tr>
<td>Other language</td>
</tr>
<tr>
<td>Loans</td>
</tr>
<tr>
<td>Solutions</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4: A distribution of the average number of calls for each customer type, for customers that were served only by IVR.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Type</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>High priority</td>
</tr>
<tr>
<td>Medium priority</td>
</tr>
<tr>
<td>Low priority</td>
</tr>
<tr>
<td>Unidentified</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>
References


