AUTOMATING THE VMEALS.COM ORDER CONFIRMATION SYSTEM

Student team: Li Chen, Andrea Cook, John Cox, James Robertson

Faculty Advisors: Don Brown, Louis Pittard
Department of Systems Engineering

Client Advisors: Lester Pearlstein,
Vmeals.com
Charlottesville, VA
E-mail lpearlstein@vmeals.com

KEYWORDS: Computer Telephony Integration (CTI), Interactive Voice Response (IVR), Cold Fusion, Sequential Querying Language (SQL), Visual Basic.

ABSTRACT

Vmeals.com is an online catering company that brokers food transactions between vendors and consumers. The order confirmation process formerly involved receiving a fax back from the vendor with a management signature, manually updating the order status in the database, and placing reminder calls to vendors two hours before order delivery times. After developing goals for the system and evaluating several alternatives, the Capstone team chose to utilize interactive voice response, IVR, computer telephony integration, CTI, software to build an automated order confirmation system that allows vendors to confirm orders over the phone. The system updates the order status in Vmeals’ database based on user input from the system. In place of a signature, the system records the name of the user at the end of the call. The “outgoing call” component of the system automatically places delivery reminder calls to vendors. In addition to the IVR solution, the team designed a dynamic web page that allows customer service agents to search the database using various query criteria. Information on the web page is drawn directly from Vmeals’ database and is therefore updated simultaneously. The team used current estimates of time spent, and the projected time it takes to use the system to evaluate its effectiveness. At current volume of 6 orders per day, Vmeals.com should expect to save $200 per month in wages, and $12 per month in fax outsourcing costs. At Vmeals.com’s projected order volume of 1000 orders per day, they should expect to save $34,000 per month in wages, $2,000 per month in fax costs, without a large increase in staff.

INTRODUCTION

Vmeals.com is an online meal-catering firm serving individuals and corporations in several different markets across the nation. Currently, the confirmation process for each order at Vmeals requires employees to visually check each confirmation fax from restaurants and manually enter confirmation information into the system. Currently, when customers place orders, the orders are processed in following steps (Figure 1):
1. Customer submits order via internet
2. Vmeals employee reviews order, generates and faxes order form to restaurant
3. Order form is sent to an outsourced fax bureau and faxed to appropriate restaurant
4. Restaurant manager signs confirmation page and faxes back to Vmeals.
   a. Any questions regarding an order are handled via telephone
5. Confirmed fax passes again through bureau, sent to Vmeals as e-mail with attached image of fax document (pdf format)
6. Vmeals employee visually confirms order form with signature, manually updates confirmation status using Excel Spreadsheets
7. Agent makes reminder calls to restaurant reminding customer of delivery
8. Meal is delivered to customer by provider

Figure 1: Current Order Confirmation System Flowchart
This process takes up a total of an hour of the customer service agent’s time each day. This figure is for a total of six orders per day. Vmeals hopes to run at a capacity of 1000 orders per day. At this volume, the amount of time spent on confirmations and delivery reminders is not feasible for the amount of staff on hand. This process not only exposes the Vmeals operation to human error and mishandling of information, but also hinders the opportunity for expansion because of this time-limiting factor. The Capstone team was brought in to evaluate this problem and provide a solution.

SYSTEM ANALYSIS

Systems engineering methodologies are a framework for analyzing a given problem and producing the best solution. The first step of systems analysis is to determine the descriptive (current) and normative (ideal) scenarios [Gibson]. The current scenario is outlined in the previous section. We brainstormed as a group and had several interviews with the Vmeals.com staff. Several goals were determined to form a normative scenario: eliminate any manual entry of database information, eliminate manual verification of confirmation (i.e. checking for signatures), simplify the checking of order status, and automate the delivery reminder calls. Because the current system has been in place for so long, Vmeals.com encouraged the continued use of the fax for placing orders. After a thorough brainstorm session with the Capstone team, Vmeals.com decided to evaluate two types of solution: automation of the fax-back confirmation system, and automation of a phone confirmation system.

The team researched price points for both types of system. Signature scanning software for automating the fax-confirmation cost $5,000 or more. Software/hardware packages for phone systems were more numerous, and less expensive, on the order of $1000-3000. In addition, Vmeals.com already used a phone system called Altiserv, which could meet some of the goals for the system. Vmeals.com’s budget and the simpler approach of phone-based confirmation dictated that we automate confirmations through the phone.

To rate this software based on its feasibility to accomplish the goals of the system, the team came up with a list of criteria the software should have.

1. Can communicate with SQL database
2. Phone menu system for confirmation
3. Text to speech for creation of messages
4. Outbound calling for reminder messages
5. Scheduled events
6. Scripting capabilities for event handling
7. Redirection to live representatives
8. Voice mail

Phone-Based Automation

Automatic call response systems have enjoyed widespread success because they have proved efficient and easy to use. Along with this, the necessary hardware already exists at companies in the form of telephones, PCs, and the networks that connect them together. In the mid 1990’s, automated call response systems were limited to playing a greeting and instructions, accepting caller touchtone input, correlating that input with an extension, and transferring the call. [Deixler,103] This helped to route calls more effectively, but did not significantly alleviate the number of personnel required to support the system.

The type of software that allows for interaction with a menu and database is called IVR, or Interactive Voice Response. Interactive Voice Response was introduced in the late 1990’s and created the ability for users to request or change information about themselves without speaking with a customer service representative (CSR). The earliest forms of IVR used “grunt detection” whereby four “grunts” indicated the number four and so on. [Bajorek, 145] IVR has since been improved to enable the caller to interact with information stored in a database using their touchtone phone to provide their personal identification or account number and to perform certain operations.

On the front-end, the system guides the user through the interface with a series of voice prompted commands and receives touchtone responses from the caller. These responses include choices made by the user to navigate to another part of the system and information about the user. Using the IVR to request information from the user enables the back-end of the IVR system to pass this information to a database where the user’s account information is stored. [Kasahara, 59] The user is now connected to his account through the IVR and can request or change information, or is routed to the CSR most capable of resolving the user’s requests.

Evaluating Alternatives

The team evaluated several different IVR software packages using its criteria, including the Altiserv package already in use by Vmeals.com. Of Pronexus, Callmaster, Cisco IVR, SynreVoice, Altiserv, and several other packages, the top two choices based on features that could contribute to the system’s goals were...
Callmaster and Pronexus. Both offered all of the features listed in the criteria.

Callmaster used a table-based system and a coding language called “CallBasic” which was a proprietary scripting language, and was offered for $1,295. Pronexus used a GUI-based system and Visual Basic as a scripting language, which meant there were many resources for its use. A screenshot of the Pronexus GUI is below, in Figure 2. Pronexus was a bit more expensive at $1,495. Both of these systems required the purchase of a voice card such as the Dialogic/4, for about $500. Because of the Pronexus system’s GUI system and VB-based scripting, it was the team’s choice on which to develop the system.

The process model provided the team with a view of how to best break up the project into its component parts. Each goal of the system, with the exception of automatic SQL database updates, was given its own component: incoming calls for confirmation of orders, outgoing calls for delivery reminder calls, and a website for the display of order status information. The remainder of this paper will explain the implementation of these three components, and present the effectiveness of the system.

INCOMING CALLS

On the first ring of an incoming call, the system is engaged and plays a welcome greeting. The call is then routed to the main menu that plays a greeting requesting that the user choose whether to confirm an order or to speak directly with a customer service representative. At anytime during the call, the user is able to interrupt the menu messages with a keystroke so that they may progress through the system more quickly if they know the command for the desired operation.

Single Order Confirmation

If the user chooses to confirm an order, a message prompts the user to enter in the confirmation code on the fax containing the order that is to be confirmed. Within the database, a unique confirmation code is assigned to each fax and each order, unless multiple orders are sent on a single fax. In this case, each order on that fax is assigned the same code. If not found, an error message is played to the user, prompting for the confirmation code to be entered again.

If the fax contains only a single order, the call is automatically routed to GetOrderNum, which prompts the user to key in the order number. Even though the system is able to determine the order number from the confirmation code, this step is required to ensure that the confirmation code corresponds with the order to be confirmed. This pairing of the code and order number provides security and accuracy by ensuring that the confirmation code was entered correctly as an incorrectly entered code may be, although unlikely, valid for another fax. When a valid confirmation number and order have been matched, the user is routed to the final step of the confirmation process.

To complete confirmation, the user is prompted to say his name so that Vmeals will have a record of who confirmed the order. If no audio is detected, an error message is played requesting that the user say his name again. Once a successful recording is made, the DataChange node, ConfirmOrderChange, then locates
the order within the database, updates its status to confirmed, and inputs a timestamp to specify the date and time that order confirmation was completed. In addition, the name of the voice file corresponding to that order is included in the database so that it can be located for future reference. The user is informed that order confirmation was successful and the call is passed to the final menu that gives the user options to confirm another fax, transfer to customer service, or terminate the call by hanging up.

**Multiple Order Confirmation**

When a user keys in a confirmation code that returns multiple orders on a single fax, the user is given the option to confirm a single order from the fax or all orders. The user is requested to key in any order number on the fax. As in the single order scenario, this is to ensure that the confirmation number and orders on the fax successfully match. After successfully matching the confirmation code, order numbers, and the voice recording is complete, each order is updated to confirmed status, each receives a timestamp, and they are all assigned the same voice file. The user is then given the same post-confirmation options as above. Below is a flowchart of this process (Figure 4), which is outlined below the figure.

**Figure 4: Incoming Call Flowchart**

1. Customer submits order via internet
2. Order form is sent to an outsourced fax bureau and faxed to appropriate restaurant
3. Restaurant employee with permission to confirm order contacts Vmeals 1-800 number, connects to IVR system that guides callers through the confirmation process with a series of prompts and system updates information to database order after successful confirmation
   a. Any calls regarding questions are passed by the IVR system to a Vmeals agent
4. Three hours prior to delivery, IVR system calls restaurant to remind of delivery
5. Reminder call is placed to customer to tell them a delivery will be made.
6. Meal is delivered by provider to customer

**Customer Service**

The user is informed at the main menu that at anytime during the call, even if in the process of entering order information, the '*' key can be pressed for automatic transfer to a customer service representative. If '*' is detected by the system, the call is routed to TimeSwitch1. At this TimeSwitch node, the current time is checked to determine whether it is regular business hours or not. If the current time is during that of operating hours, the caller will be routed to the first available CSR. Otherwise, the system informs the caller that it currently is not operating hours for Vmeals. The user is then informed of normal operating hours and given the option to leave a voicemail, return to the main menu, or disconnect.

**AUTOMATED REMINDER CALLS**

Customer service agents at Vmeals.com have the added task of reminding their vendors of pending orders. These teleminder calls ensure that the vendor knows of orders with enough time to cook them and deliver them on time. Currently, customer service agents make the teleminder calls throughout the day, two hours in advance of an order’s delivery time. Agents then update the results of this call on an Excel-based “Teleminder Sheet”. It was decided that automation of this process would be beneficial to the customer service agents.

This part of the system periodically checks for orders pending that are within 3 hours of delivery. It does this using a tiered system for filtering the records. First, the system selects all the orders that are to be delivered on the current day that have not already had a successful reminder call, or more than 5 tries for the reminder call. It then selects, from this record set, the orders that have already been confirmed by the vendor. The system then checks how long it has been since the last time it called regarding this order. If it has been less than 5 minutes, it proceeds to the next order. If 5 minutes have passed, it then determines the location of the vendor using location information in the database.

Using this location, VB code determines the offset for the time zone in which the vendor is located. The next control calculates how much earlier the current time is before the delivery time. If the result is less than 3 hours, control is passed to the call section of the
program. If not, the next record in the record set is checked.

The database is updated at this point to record that a call has been placed. The system then dials the number of the vendor related to the current order. If there is no answer, it is busy, or an answering machine answers, the system hangs up and returns control to the record set filter. When someone answers, the prerecorded greeting is played, which informs the vendor of the order number and delivery time for that day. To confirm that this message was received, the system asks the vendor to enter the confirmation code on the fax for that order. If they do not have this information, the vendor can hang up, and the system will return control back to the record set filter. The vendor will receive another call in about 5 minutes. If the confirmation number is entered, the database updates the field that keeps track of reminder call status. The system then moves on to the next order.

CONFIRMATION STATUS WEBSITE

We have developed a dynamic web page for the customer representatives to monitor order confirmation status. Cold Fusion was used in creating the web page because it provided us a set of tools to quickly build dynamic, interactive, data-driven web page. The web page allows users to check confirmation status in several ways; users have the options to view all unconfirmed orders, by specifying a location, a range of order numbers, or time of delivery. Upon hitting the submit button, a query is executed to extract corresponding confirmation status from Vmeals’ database through an ODBC and the results are returned. In addition, each record has a link that will open up a detailed page containing information such as customer contacts, items ordered, and all other relevant information concerning that specific order. The structure and connectivity of the web page is shown below in Figure 5.

TESTING

We used a five-part test procedure for the entire system. The first two parts are set up procedures, the second two parts involve simulation on the incoming call interface, and the last part involves simulation on the outgoing call interface.

There are a series of six tests for the incoming call interface. Each test has four parts and a series of steps for the tester to follow. The first three tests run simulation through the incoming call queue to ensure that the system functions properly at the most basic level, with no user mistakes and correct user input at all prompts. The second three tests simulate errors in user input at each of the prompts in the queue.

The accuracy and effectiveness of the outgoing call interface was examined by monitoring the placement and status of reminder calls as the system progresses through the queue. The call simulator allows the tester to simulate the different scenarios that real life users may encounter. The tester follows the test plan, monitoring the teleminder simulation and verifying that the system is performing correctly.

RESULTS

Vmeals.com currently processes approximately 30 orders a week, which translates to an average of six orders per day. Prior to the automation of the system, the customer service agents spent approximately one hour per day checking order confirmation faxes received, updating order status in the database, and placing reminder calls to vendors for the six orders. Order volume is anticipated to eventually increase more than ten-fold, up to a thousand orders per day. At this projected volume, the customer service agents would have to spend 170 hours or more each day checking order confirmation, updating order status, and placing reminder calls. The implementation of our solution system requires no time from the customer service agents in the order confirmation process outside of vendor assistance, thus decreasing the necessary customer service man hours. General customer service inquiries will increase, however, with the amount of orders. Setting the customer service wage at approximately $10 per hour, estimated labor savings provided by our system are roughly $1,700 per day. This translates to savings of roughly $34,000 per month in wages. Our system also cuts the cost of outsourcing faxes. Vmeals pays $0.10 per fax page received and previously received a one-page confirmation fax for each order. With the implementation of our system, Vmeals no longer receives faxes from vendors, thus,
our system decreases the projected outsourcing fax cost by $100 per day, or $2,000 a month. At the current order volume, the savings are $200 per month in wages and $12 per month in fax outsourcing costs. The implementation of our solution system will incur a minimal new variable cost. The system will be integrated with Vmeals.com’s current phone system, Altiserv, which is reached by a toll-free number. The cost for this number will increase as vendors will exclusively call this toll-free number to confirm orders. Vmeals does not know the exact cost of this number. The Pronexus software purchased to build the solution system was priced at $2,000. Based on the numbers above, it is evident that implementation of the solution system should provide large savings for Vmeals.com. It will also allow the company to expand, at its current number of staff, well beyond 6 orders per day.

REFERENCES


Gibson, John E. “How to Do a Systems Analysis” Ivy: Gibson


BIOGRAPHIES

Li Chen is a fourth-year Systems Engineering student concentrating in computer and information systems. She was primarily responsible for the design and implementation of the cold fusion web pages that are used in viewing order confirmation status.

Andrea Louise Cook is a fourth-year Systems Engineering major from Potomac, MD, concentrating in Business Management and an Economics minor. Her main contribution to this project was in system design and testing. Ms. Cook plans to stay in Charlottesville for the summer and potentially move to San Diego, CA in the fall.

John Walter Cox is a fourth-year Systems Engineering major from Singapore with a minor in Economics. His main contribution to the project was in system design and programming. Mr. Cox plans to spend next year studying and working in Shanghai, China after graduating this May.

James Robertson, coincidentally, is also a fourth-year System Engineering major, with a concentration in Computer Information Systems. His main contribution to the project was system design and programming. After graduation, James plans to remain at UVA to pursue a Master of Science degree in Systems Engineering.