A Scalable Architecture for Enterprise Level Call Center Design

Jorge Morales Irizarry
Master of Engineering in Computer Engineering
Jeffrey L. Duffany, Ph.D.
Electrical & Computer Engineering and Computer Science
Polytechnic University of Puerto Rico

Abstract *— This article presents a project module of a larger Information Technology initiative as purposed by Company X. The module illustrates research efforts into potentially viable solution options through the use of network protocols. Also presented is the architecture and software design framework of the chosen method of implementation for this project. Prior to any in depth conversation, this article will include explanations of basic concepts not explicitly covered as a part of the project domain. This is because although not the projects focus, these concepts are deemed pertinent to understand in order to facilitate the reader's appreciation of the more detailed, non proprietary topics covered. Research findings, results and recommendations are also mentioned.*

Key Terms *— †Automatic Call Dialers, Cisco GED-188 Network Protocol, Computer Telephony Integration (CTI), †Global Outbound Solution.*

INTRODUCTION

In the telecommunications industry, there are several business aspects that are imperative to hone, develop and continuously improve upon in order to distinguish yourself from your competitors. These can include but aren't limited to network coverage, network availability, and call quality just to name a few. Their importance are generally easily justified. In many cases, these business aspects are the very same measures utilized by consumers to gage one provider over another and the deciding factors employed in establishing which provider would adequately suit their needs.

Although the aforementioned business facets are undeniably essential, there is however one common factor that cannot be understated in its importance; the consumer experience.

Consumer experience is one of many contributing themes that comprises an overall indication of customer satisfaction. More often than not, this measure specifies to a company how well their clients perceive its ability to provide them with a superior product, support and value for their money spent. A universally accepted truth is that the greater your customer satisfaction, the greater the possibility you can retain your customer base. This, in turn enhances the possibility your services are referred to other prospective consumers by your customer base.

Company X whole heartedly believes in the relevance of the customer experience. It attributes its achievements and successes; both as a telecommunication provider and in its respective market, to their focus on and excellence in customer satisfaction and the customer experience. Thus in their pursuit of continuous amelioration of the customer experience, an initiative was founded referred to as the †*Global Outbound Solution*. It is this broad spanning proposal that gave birth to this project as well as various others that once completed will encompass the entire proposal plan.

Company X's †*Global Outbound Solution* is an information technology based concept that aims to leverage their network capabilities in an attempt to exploit the capacity possessed through their interactive voice response (IVR) equipment assets. It's anticipated that this plan will facilitate more efficient and effective business practices through cost saving measures that will ultimately yield an even greater customer experience.

BACKGROUND

Company X's early successes as a telecommunication provider was based in large part to the emergence of the mobile phone market. In order to "catch up" to consumer demand, they acquired many smaller companies to assist in providing their service. Although fine for a time, now they are faced with a peculiar situation.

Through their attainment of other companies, they came to not only possess business processes but also any systems utilized by these acquired companies. Thus arose a nuance. There exist some internal company departments that use certain systems that do not coincide with other company departments support services. These conflicting and uncoordinated systems and their corresponding limitations have lent themselves to inefficient and ineffective practices that undermine certain departments efforts to provide particular support and services. This is because they rely on other internal department's ability to supply them with the information required for them to function, so that they can extend their services to customers. This ultimately has a potentially negative and direct effect on the overall customer experience.

The Collection Services department is one prime example of a departmental service affected by other conflicting system limitations.

Collection services is the department charged with interacting with those customers who have been designated as grossly negligent in terms of paying their monthly bill in a timely fashion. Once they apply their business process to determine which customers need to be contacted, a list of phone numbers to dial is comprised. This list is then made available to an IT IVR team that must work in conjunction with aspects of network components supported by a network team to take the appropriate steps to begin what is known as an †*agent based outbound call campaign*. In this task, the IT IVR team must interface with the network to obtain all the necessary information to then configure and optimize IVR usage as best as possible once the campaign is set to take place. It is here is where cog presents itself.

The IVR team employs a separate vendor to manage the physical connections of the IVR to the network components. The vendor is also tasked with the retrieval of certain communication messages between the network components so that the IVR's can be configured in accordance to the call campaign's execution. As call volume into the IVR & number of outbound campaigns increased; reaching its current maximum of around 5 million calls per day, difficulties arose with the vendors ability to adequately and efficiently manage the IVR systems as required by Company X. Along with this, the inefficiencies drove the company cost per call connected to an estimated $12.00. Due to vendors own specific limitations in their management system, Company X ensued to investigate other possible vendors. As non were found who could satisfactorily comply with the desired requirements, Company X decided that it possessed the network capability and sufficient IVR capacity to customize their own proprietary solution to ensure their stringent customer experience requirements were met. It was determined that the system should be developed in java, given the strong movement to move most development projects to a centralized java based platform.

To better garner an general feeling for what an IVR is, "Reference [1] informs us that the purpose of an Interactive Voice Response (IVR) system is to take input from a caller, perform an action based on that input (commonly, looking up data in an external system such as a database), and return a result to the caller " Madsen, L.

PROBLEM STATEMENT

As a part of the project, there were guidelines instilled in the development process to ensure that each aspect of the project was correctly adhered to and verified. These guidelines of this project are briefly discussed in the subsequent sections.

Research

The research portion of this project is intended to investigate all possible protocol alternatives to the current vendors provided services. All choices must be compatible with Cisco based network components. They also must be capable of retrieving the information identified as essential to this initiatives success.

Presentation

This presentation is to be made to Information Technology, Business and Network Integration

management teams. The presentation should mention the projects overall goals and how they intertwine with the larger †Global Outbound Solution initiative. It must also include a mention of all alternatives that where researched along with their advantages and disadvantages.

Framework Development

The framework must illustrate preliminary architectural design topology in a top - down manner as well as a class diagram of the intended software development plan. Processes used are summarized through the use of sequential diagrams.

Implementation

Implementation is to consist of a rudimentary proof of concept of the recommended protocol. The goals of the prototype include:

- An explicit demonstration of the prototypes ability to interact with the network components including the acquisition and requesting of communication messages between network components.
- Display of message filtering use organize and distinguish between messages.
- Illustration of how the communication messages obtained can be obtained, utilized and processed.

PROJECT DESIGN

The approach taken for the research aspect of the project was to first inspect the different network components involved in the communication process as documented by the current vendor. This provided the opportunity to isolate the key network components and how they are exploited to obtain the required information. These key components were identified as the Peripheral Gate (PG) and the Intelligent Contact Management (ICM).

From this point, the research into viable alternatives entailed which protocols feasibly could be employed to interface with the identified network components. Secondly, how readily available would the data be from said components and how easily it could be obtained was another important aspect considered.

Design Alternatives

Various different alternatives to the current solution in place where investigated. Many addressed certain aspects of the requirements instilled by both the *†Global Outbound Solution* and the need to be attuned with the network components in place. However, due to the specific and peculiar nature of these requirements and compatibility needs there were few protocols that plausibly be applied.

The GED-188 protocol is a Cisco based procedure that patches into a network component charged with managing the communications of each automatic call dialer on Company X's system arrangement. Each automatic call dialer is tied to a call center and handles the accumulation of call agent data generated by that call center. This protocol supplied two different implementation methods. These are known at the Computer Telephony Integration (CTI) Server and the Computer Telephony Integration Object Server (CTI OS).

Reference [2] explains to us that "Computer Telephony Integration (CTI) is the set of software and hardware components that allows a computer to manage telephone calls and integrate telephony services into desktop computers, servers, PBX devices and other computing equipment" Yarberry, W. Thus the CTI Server would be hardware equipment of the overall CTI concept.

Another possible protocol was the GED-125 protocol also offered by Cisco. This procedure makes use of a different network component than the GED-188 protocol. It employs the Intelligent Contact Management (ICM) component. The ICM is considered to be the brain of the network, being that it is tied; directly or indirectly, to all of the other network components. Hence it's name, the ICM is what coordinates all of the network activities and diverse communications along the network channels.

Each method has its own specialty and manner in which the information from the automatic call

dialers are obtained. The CTI Sever is a low level, direct communication method between the PG and the automatic call dialers. The CTI Server actually sits on the Peripheral Gate. The CTI OS is an object oriented method of communication between the same two network components. The Cisco GED-125 protocol uses the ICM network component rather than the PG to obtain the necessary information from the automatic call dialer. Each method has its pros and cons and will be discussed in the subsequent sections.

GED-188 CTI Server

The CTI Server is a low level, direct communication method between the PG and the automatic call dialers. "Reference [3] indicates that the CTI Server provides an interface between ICM Software and client CTI applications where the CTI Server runs at a call center site either on an ICM Peripheral Gateway with automatic call dialer interface software or on a dedicated CTI Gateway platform", Cisco Systems Inc. Figure 3 at the end of this section is provided by Cisco and illustrates a general design architecture of the CTI Server and how it interacts with the other network components. This is shown in Figure 1.

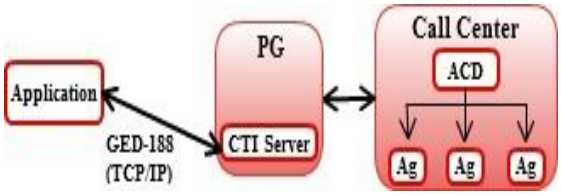

Figure 1
CTI Server Flow

The CTI Server accomplishes it's direct communication through the use of messages consistent with Publish & Subscribe and Request & Acknowledge paradigms. This means that the CTI Server can subscribe to specific message services broadcasted by the automatic call dialer or it request specific messages from the automatic call dialer. This is possible because the CTI Server institutes a set number of message categories. These categories are; session management, call events, agent events, client control and miscellaneous.

Each of the message categories follows a strict and specific message format. There are three main portions to the message format; message header, fixed-length fields and floating fields. The message header is used to separate and classify which message is being sent to or received by which automatic call dialer. The fix portion fields is designated for the developer to indicate to the intended automatic call dialer which message services from it is explicitly requesting if such a feature is desired. The floating field portion of the message format is to indicate specific messages that are being requested by the CTI Server. It is important to note that message header and fixed portion of the format are required. The floating portion is optional. The general message format is shown in Figure 2.

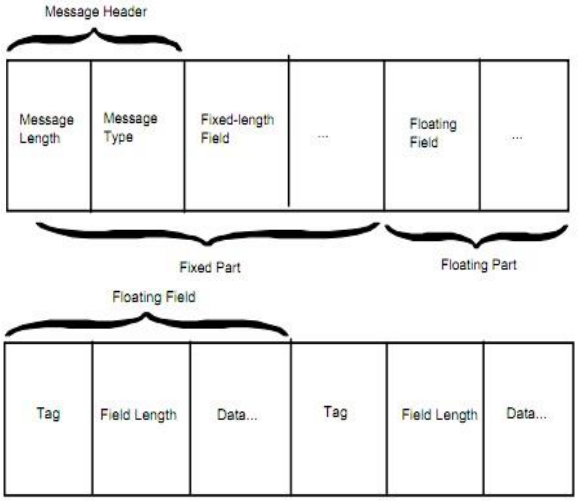

Figure 2
CTI Server Message Format

One major advantage of the CTI Server method lies within the design of how messages are passed back and forth. Having the liberty to either subscribe to generic messages broadcasted by the automatic call dialer or requesting specific messages provides the developer with a healthy amount of information. at their disposal. Another advantage of this method is the complete control over the message content, assuming the previously indicated format is followed.

There are however inherent disadvantages associated with the CTI Server method. The first clear disadvantage is in the connection management.

The CTI Server requires 2 connection types be active when being utilized. These connections are (1) physical connection of the CTI Server on the Peripheral Gate and (2) connection between the CTI Server and the automatic call dialers. The physical connection is a standard TCP/IP connection that must be constantly monitored for fail over and valid connection procedures . The link between the CTI Server and the automatic call dialer must be valid, thus the programmer must periodically validate that the link is open. This becomes a bit cumbersome because as mentioned earlier, there is one automatic call dialer per call center. Thus, there is a potential for managing various different automatic call dialer connections since each call center could potentially be using a different brand or manufacturer automatic call dialer. Both these connection types must be managed by the developer.

The second disadvantage lies in the message construct themselves. This is in reference to the possibility of different automatic call dialer models. Each automatic call dialer model and brand has their own particular format. This implies that the CTI Server message construct must coincide with the automatic call dialers message construct. Thus, the possibility for management of various different message constructs is evident.

Lastly, the nature of the Publish / Subscribe method implemented by the automatic call dialers presents difficulty. This is because although you can indicate specific message services, you can not specifically target specific aspects of the subscribed messages without the use of message filtering. Thus the possibility of receiving unwanted published messages is high.

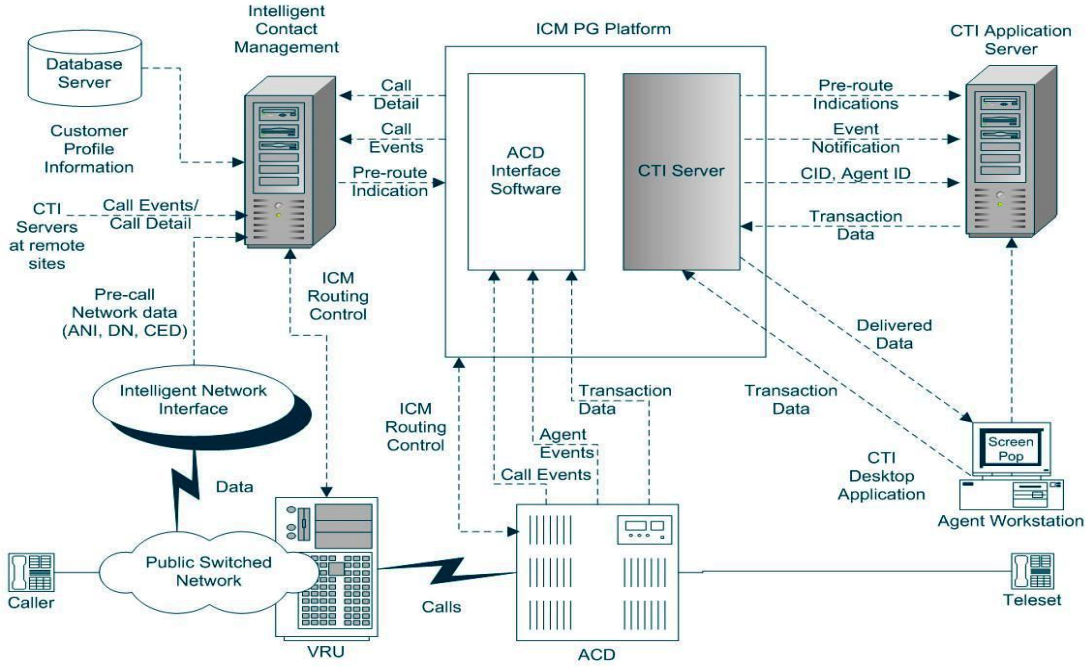

Figure 3
CTI Server

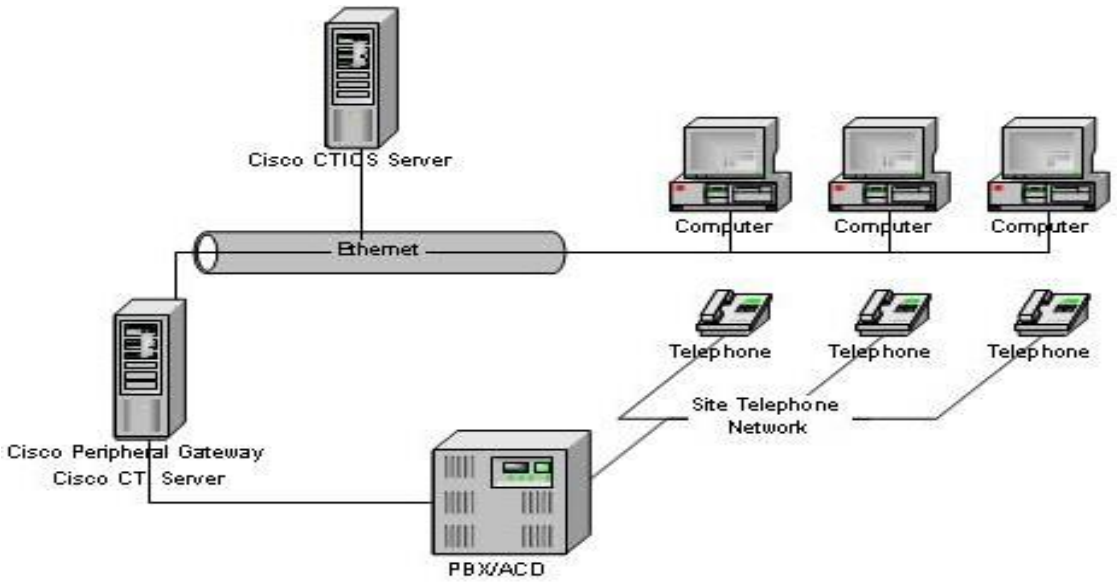

Figure 4
CTI Object Server

GED-188 CTI Object Server

"Reference [4] tells us that the Computer Telephony Integration Object Server (CTI OS) is an object oriented integration platform that is provided through a software development toolkit", Cisco Systems Inc. Essentially, the CTI OS implementation method is an aggregate to the earlier mentioned CTI Server. It combines the use of a separate object server in conjunction with the CTI Server. It implies Figure 4 at the beginning of this section displays the general overview of the CTI OS architecture.

As mentioned above, the CTI OS is geared towards an object oriented software development environment while still facilitating the lower level connections necessary to communicate along the network. To accomplish this, CTI OS is divided into two main parts, the CTI OS Server and the Client Interface Library (CIL). The CTI OS Server is charged with handling the network connections aspect of the protocol and the CIL is the main catalyst for the object oriented portion.

The Client Interface Library (CIL), in order to provide its object oriented nature, it can be utilized in various different programming languages (.NET, C++, Java). The CIL itself is designed as a 3-tier architecture to fulfill both functionalities as required by the CTI OS Server and the developing environment with a translation tier in between them. These 3 tiers are as shown in Figure 5:

- Object Interface Layer
- Service Layer
- Connection Layer.

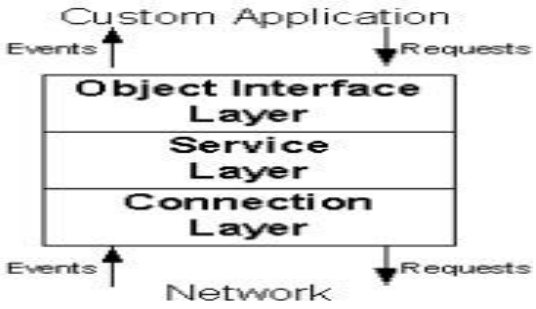

Figure 5
CTI Object Server Client Interface Library

The Object Interface Layer is where the programmer will develop his code in one of the language provided by the CIL that best suits him. The Service Layer is tasked with taking the code as written by the developer and translating it into the message constructs recognized and compatible to both the CTI Server and CTI OS Server and delivering the translated code to the CTI OS Server.

The Connection Layer is the portion of the CIL that manages the passing of message constructs between the CTI OS Server and CTI Server. This architecture is illustrated in Figure 6.

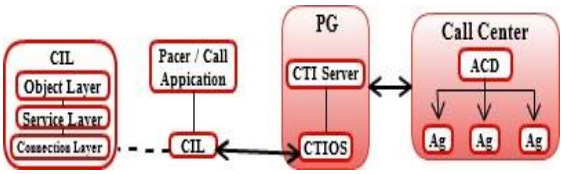

Figure 6
CTI Object Server Flow

It's important to note that the CIL does not reside on the PG as in the case of the CTI Server and CTI OS Server. The CIL, being an development library, is utilized within the actual programming environment.

To facilitate the lower level connections that the CTI OS Server will employ to exchange messages with the CTI Server while still preserving the object oriented methodology, the CTI OS Server segment of the CTI OS method utilizes an object which is illustrated in Figure 7. These objects are the main structure of the CIL and are what will be translated via the Service Layer into the constructs compatible with the CTI Server.

Figure 7
CTI Object Server Session Categories

There are four main objects that are employed in this model: Session, Agent, Call and Skill group.

The Session object is the main object in the CIL's model. It controls the logical session between the application and CTI OS Server. It accomplishes this through providing object management (creation, collection and deletion) and publishing all CIL events. This implies that once a session is created, all other objects can be accessed and implemented to achieve the desired task.

The primary advantages of this method is that all of the capabilities of the direct method are preserved. They are simply offered in a object oriented software development environment that most developers are already accustomed to. Next, through the use of the client interface library's (CIL) architecture, the responsibility of managing the standard TCP/IP connection shifted. The developer is no longer required to handle this connection. The connection layer of the CIL automatically manages this aspect, essentially hiding the backend connectivity from the developer. This also allows for automatic call dialer transparency, where as the CTI Server was automatic call dialer dependent.

The major drawback to this method is the dependency on the CIL. There are some difficulties associated with the use of the client library since every interface that could be used is exclusively found on the CIL. This essentially implies that the developer now has less control over message constructs since he can only use what is permitted by the client library. Also, another drawback is in the only connection that needs to be monitored by the developer. The connection from the client library to the CTI OS Sever must monitored and verified periodically. This is because if not, then no events can be retrieved.

GED-125 ICM Direct Communication

As stated earlier, Cisco's GED-125 protocol utilizes a completely different network component than the GED-188 protocol; the Intelligent Control Management (ICM). The ICM is considered the brain of the network because of the vast connections it possesses to all subsequent components on the network. other words, all activity on the network at some point or another must traverse through the Intelligent Control Management. This implies that the ICM has both direct or indirect knowledge of messages being passed along the network.

GED-125 proved to provide more disadvantages than advantages. Being tied into most aspects of the network, stress and loads on the ICM

were already considerably high. When the estimated addition activities on the network were presented, the Network Team informed us that the additional load would strain the network to the point of failure. In lieu of this fact, the GED-125 was disregarded.

Final Design Choice

The CTI OS portion of GED-188 protocol was what ultimately chose to begin the design and implementation phase of the project. The reasoning for its selection was quite simple.

CTI OS preserved all of the functionality of CTI Server but without the necessity to manage all of the lower level network connections as in with the CTI Server. The connection layer of the CTI OS architecture manages this for the developer. That fact coupled with an object oriented style of development environment caters well to allow the IT team to leverage its current development strengths and capabilities. Admittedly, it took time to fully garner a decent understanding of the Client Interface Library (CIL) as well as which interfaces would be used in the project and how to implement them per CIL guidelines. The documentation provided by Cisco for the CIL through java docs was of great help and was very easy to follow.

Framework / Implementation

It was decided that this project would be implemented in a java framework; coinciding with the organizations internal movement towards a uniform software development platform. This selection coincided smoothly with both potential GED-188 methods. The direct CTI Server protocol; implemented using socket programming, could utilize specific java libraries that catered to this programming style. The CTI OS method could leverage the developers familiarity with Eclipse IDE. Utilizing the java language to perform the object oriented tasks allowed for the use of common design patterns; such as the iterator, singleton and abstract factory design pattern.

FUTURE WORK

The †*Global Outbound solution* is a project undertaking that never has been before attempted in Company X. As is always the case in software development initiatives, there are always unforeseen difficulties. In Company X's case, this could potentially stem from features of the protocol that are not correctly implemented. Also, the possibility of undiscovered functionality necessity are also a risk since this is a joint project between various other departments. Lastly, requirement needs are liable to change once the used protocol is better understood. These certainties spurred deliberations amongst the departments involved with this project. It was determined that it was imperative to fully understand the capabilities and potential short comings of the GED-188 protocol. Thus, it was decided to expand the projects scope.

The projects deadline was extended one fully year. Within this time frame, an addition four phases where added where the following is to be accomplished:

- Phase 2: Rudimentary prototype of GED-188 CTI Server.
- Phase 3: Full beta version of GED-188 CTI OS.
- Phase 4: Full beta version of GED-188 CTI Server.
- Phase 5: Comparison analysis between both protocol implementations.

CONCLUSION

It's my personal believe that Company X's †*Global Outbound Solution* initiative is indicative of the beginnings of paradigm shift amongst the larger corporations in the United States. Slowly, organizations are beginning to realize the importance and inherent benefits of internally developing or expanding their Information Technology infrastructure. In recognition of this, more and more companies are seeking out professionals that are adept in the Computer Science and Computer Engineering fields so they can combine their assets with the personnel knowledgeable in maximizing the efficiency and

utilization of said assets. At the end of the day, more efficient practices translate into cost savings while succeeding in the pursuit of a quality product, a better customer satisfaction measure or overall effective company practices.

NOTE

This project was completed in conjunction with a Fortune 15 telecommunications company whom for the sake of anonymity will be referred to as Company X. Due to certain proprietary business processes & information of Company X, some detailed information must be omitted. In these cases, general knowledge & concepts will be provided. Thusly, any topics or statements that are proprietary in its nature will be indicated by a † label immediately adjacent to its mentioning.

REFERENCES

[1] Madsen, L, *et al.*, "Interactive Voice Response", *Asterisk: The Definitive Guide*, 3rd Ed, April 2011, pp. 389-396.

[2] Yarberry, W, A, "CTI Concepts and Applications", *Computer Telephony Integration*, 2nd ed., 2003, pp. 49-59.

[3] Cisco Systems Inc, "CTI Server Message Reference Guide Version 13", Retrieved on June, 08, 2013 from: http:// www.cisco. com/ en/ US/docs/voice_ip_comm/ cust_contact/contact_center/ctios/cti_server_reference/programming/guide/CTI_Svr_Msg_Ref_Gde_Prot_13.pdf

[4] Cisco Systems, Inc, "CTI OS Developer's Guide for Cisco ICM/IPCC Enterprise & Hosted Edition", Retrieved on June 12, 2013 from: http:// www.cisco. com/en/US/ docs/voice_ip_comm/cust_contact/contact_center/ctios/ ctios7_0/developer/ guide/cti70dev.pdf