

# Manual Testing and Quality Monitoring of Interactive Voice Response (IVR) applications

Akhil Mittal

Technology Lead, Infosys Technologies Limited

## ABSTRACT

Do you know which is never busy? Your answer would be Watch. Yes, you are absolutely correct. Time is one of the precious factors in our life and to persevere with time there is another dimension called IVR which is never busy and always set to assist their customers, 24 hours a day, 7 days a week. In today's world customer relationship hold the key to success or we would say survival also. Thanks to the globalization where challenge is just 800 number away. Various organizations are currently using more than hundred call centre [1] related applications to support its external and internal communication interfaces. These IVR applications truly symbolize the brand of a company by seamlessly offering complex self-service features like credit card processing, employee benefit surveys, order processing and status, inventory confirmation, bill inquiry and submission, account status and stock quotes and orders etc. to their customers. To constantly stay ahead of business, organizations need to adopt innovative technologies, migrate their IVR applications from legacy platforms to enterprise platforms to improve operational efficiencies & enhance customer experience..

Investment such as in Migrating IVR applications from one platform to another platform are huge and involve lots of risk. Organizations cannot go ahead and do proof of concept (PoC) [2] as migrating applications from one platform to another is a big initiatives with full of unknowns and include software development costs, license fee and infrastructure cost etc. Organizations really have to de-risk and plan their development and more importantly their testing strategies by careful planning and execution approach. Organizations need to decide success factors beforehand. This paper outlines importance of manual testing and use of automation tools to facilitate organizations in making good testing strategies for an improved end-user satisfaction – to help customer retention and bring loyalty.

This paper aims to give an insight on what to consider while choosing between Manual Testing or Automation Testing to test IVR applications, Why automation is difficult in testing IVR applications and use of automation tools in monitoring IVR applications.

## Keywords

IVR, CTI, DNIS, call centers, Manual testing, automation testing, automation tools, monitoring

## 1. IVR Basics

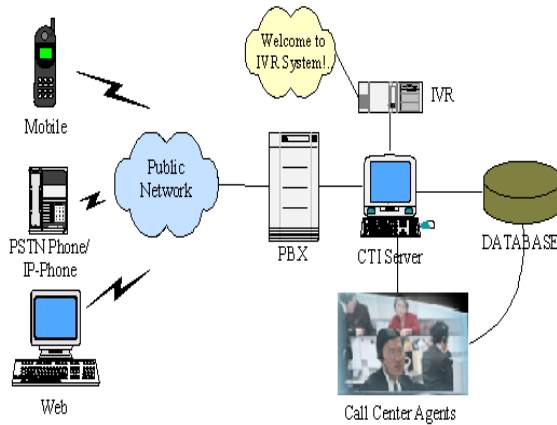
**IVR** [3] – Interactive Voice Response applications allows customers to call and navigate through various instructions to check on account, pay bills, order service etc. IVR applications interacts with the caller to determine (via collection of customer entered dtmf digits) who is calling and the reason for the call. The IVR can also provide automated services to the customer e.g. Automated Bill Inquiry. The IVR acts as a play-and-collect box where the only call routing logic exists to provide fall back routing capability. The IVR performs a database lookup based upon the collected digits and passes the customer data to the CTI [4] infrastructure associated with the call.

## 2. IVR Architecture

The IVR is connected to:

- CTI Server (for attaching customer entered digits and data dips, getting DNIS [5], performing transfers). It is used for collecting useful information from a caller before the call is transferred to an agent.
- Customer profile database or other customer data sources (for performing the database lookup) to get the customer personal as well as other information like billing information.
- Integrated Reporting Server (for storing data related to reporting) for analysts to analyze the call patterns like average duration of calls, type of callers, call hang-ups etc.

Following is the simplest IVR architecture:



For a customer call to be successful, all system components like telecom switch, IVR application, database, network, and CTI must work together seamlessly. To ensure seamless performance, most organizations test their system in the pre-production environment before deploying to production environment and production environment prior to large scale deployment. Yet, for multiple reasons, including time or expertise constraints to poor planning or execution, these tests fail to capture all the problems and truly prove that the voice system can meet the business objectives it was designed to support. As a result, costs go up, and customer Quality of Experience suffers. Problems are left for the customers to detect and for technical teams to scramble to resolve. Proper testing techniques and execution can help improve the performance of any step in the call flow of IVR applications. Note, however, that ensuring quality does not end with successful pre-production and production testing. Because a contact system will undoubtedly undergo software and hardware changes during production, ongoing monitoring is also very important. It will immediately alert contact center operations personals when problems occur, providing them with real-time performance data they can use to resolve these problems quickly basis. This document aims to provide some key inputs to assess the current testing practices and recommend industry best practices to improve effectiveness of IVR applications and enhance end-user experience as well-tuned applications can result in savings of millions of dollars per year.

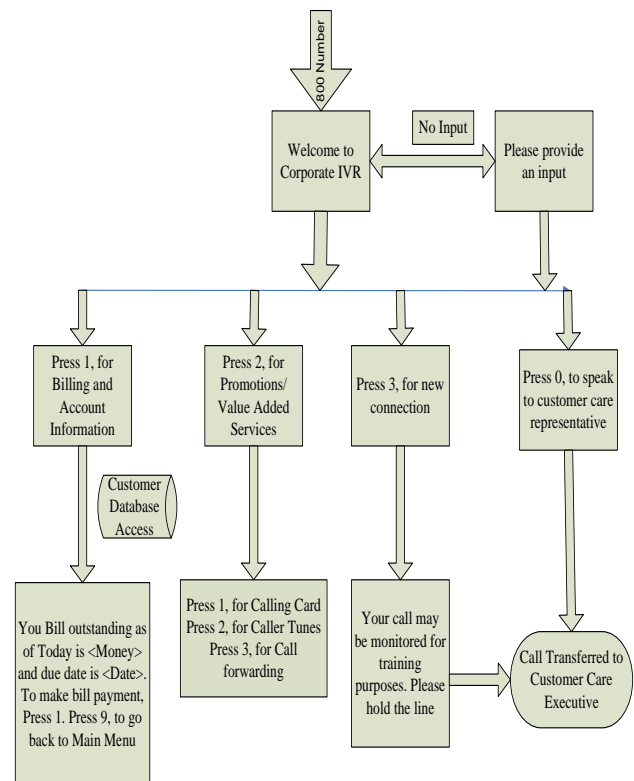
### 3. Main business drivers for undertaking this study:-

- An improved end-user satisfaction – to help customer retention and bring loyalty
- Bring cost benefits to companies using IVR applications
- Improved effectiveness of the applications

- Exploring the use of new technologies to enhance customer experience

The most often encountered problem in a company when it comes to testing is when to automate and when to manually test. We know that various companies are consolidating DTMF/legacy/regional applications into enterprise applications using state of the art technologies such as Speaker Verification, Magic Word/ hot Word Recognition like “**Complaint**”, Text to Speech (system which converts normal language text into speech) and State Recall but still there are lots of legacy/DTMF applications that are written in proprietary languages and very difficult to migrate due to multiple factors like cost benefits, language supported, application design complexity, language mix, expensive proprietary platforms etc. There is no industry standard development environment to develop IVR applications. It’s a known fact that it is very critical to incorporate customer’s needs and expectations into the IVR applications but to test these IVR applications is more challenging and cumbersome job to ensure that customer needs are met. While development focus should be towards callers who are adverse to the self-service usage but testers should understand the issues they face while using IVR’s.

Below is basic of IVR for telecom service provider:



Every IVR is in the form of call flow diagram. This diagram shows all possible paths that can be traversed through the system by a caller. For example, Telecom service provider can broadcast value added service details wherein information such as billing information, apply for new connection and others information can be listen in real time environment.

There are tools available in market to test IVR applications but all/most of the tools interpret each and every flow/guideline literally, without bearing any other thought to what else is in application.

Testing IVR applications is more concerned with **HOW TO TEST** NOT **WHAT YOU TEST**...as from switch to database/backends, Middleware and CTI interaction, testing needs to be done to ensure system performance, identification of bottlenecks and immediately test the system/applications to make sure that problems are resolved and application is back to normal working condition.

Many developers, and in particular those that are new to call centre domain and telephony, have experienced difficulties while testing IVR applications.

Following are some of the application technical aspects which should be measured while deciding on testing strategies:

- What all functionalities do you want IVR should provide?
- What all databases/stored procedure/mainframe systems need to be read from/written to by the IVR?
- How IVR application will connect/communicate to other applications/interfaces
- What should be the input to tester in case of Manual and input to tool in case of Automation testing?
- **CTI interface** – Application developer need to see if existing application business logic/platform will talk to new/migrated platform without change in any business logic of current IVR application. Is there any need to change the CTI logic as well? If the answer is yes, then tester needs to test one complete end to end flow for CTI interaction
- **Server specific processes:** Does your existing application uses any server specific proprietary features (e.g. system shell scripts, CTI scripts etc.). If yes, consider moving these scripts to the new platform and testing of same. Sometimes especially in legacy

applications there are some scripts which are specially designed to interact with database. Testing strategy should include migrating and testing these scripts to new platform or re-writing these scripts for new platform and test accordingly. Also do consider and check if your application doesn't fail due to run time environment like jvm [4], oracle version etc.

- **Alarms:** Legacy applications use alarm mechanism to report events/failure scenarios. They use proprietary alarm mechanism. Testing strategy should include creating some failure scenarios and generate alarms to test the failure situations.

An IVR application testing life cycle involves thorough testing of the application prior to its installation in a production environment in which IVR processes inbound phone calls, plays pre recorded messages including information extracted from databases and potentially routes calls to call centre representatives. It also involves testing of program flow, error conditions, database access and updated tests, CTI interaction, load testing etc.

Automated test tools are powerful support to improving the return on the testing investment when used wisely but for IVR applications there is a need of human judgment to assess the accuracy of the result or extensive, ongoing human intervention to keep the test running. Automated testing is an enormous investment, one of the biggest that organizations make in testing. Tool licenses can easily go upto six to seven figures with training, consulting, and expert contractors can cost more than the tools themselves. Then there's maintenance of the test scripts, which generally is more difficult and time consuming than maintaining manual test cases. Moreover, the organizations do enable isn't compatible with other vendors' products. Automation solutions doesn't allows to leverage already developed some of already existing scripting work, difficult to enabling those scripts and programs to be run through and stored in automation framework as well.

#### 4. Factors that influence manual testing over automation testing

- **SPECIALIZED SKILL:** - Developer/Tester needs to be well versed of application architecture/functionality. They should have deep understanding of application from unit test perspective and testing methodology. Sometimes specialized skill is required in integration/performance testing and debugging of errors.
- **TEST DATA SETUP:** - Big challenge in setup of test data. Organizations usually have multiple back-end system interfaces (e.g. CORBA, MQ, Web services etc) of

interfaces to their back-end systems (billing, CRM etc). Testing these interfaces using automation tool is very difficult and would not be effective specially if your back-end is behind some firewall.

- **DEBUGGING:-**The major issue is also associated with debugging wherein tool should be able to generate sophisticated call log files for further analysis. Most of the DTMF applications running today either running on proprietary tool or their own environment (environment specific processes). Here it is very difficult/costly to integrate these tools with proprietary tools. Here we need to see the tradeoff between benefits v/s costs involved.
- **LACK OF PROFESSIONAL TOOLS:** - Lack of professional automation/open source tools to test IVR applications specially DTMF applications. They are usually costly, very complex to setup and operate and needs specialized skill to use these tools. Sometimes tools are developed to solve traditional problems but these days with technology advance and complex domains, it become inappropriate to deploy automation tools. The Automation tool should be flexible and maintainable according to user needs which leads to lack of professional's who have in and out idea of how to use particular automation software. Careful consideration should be given to the ability to integrate third party products, to add custom programming and to further customize the standard controls provided by the tool and not proprietary to the selected tool.
- **END TO END TESTING:** - Testing end to end flow is impossible with any automation tool without human intervention. Here the challenge is, we have to decide till what point we can automate the testing. But a broad usability assessment of IVRs is based on end-to-end recordings of calls so the only complete records of caller and system behavior in IVRs are complete calls.
- **CALLER PROFILE:** Manual testers need to test the behavior of the IVR system with different caller profiles so that they can simulate the behavior of actual callers. Changing the executable test scripts is very tricky with Automation tools. Moreover, the tool must be capable of sending any sequence of DTMF tones as calls are generally recognized using DTMF tones.
- **CHANGE IN SCOPE:** - It is very difficult to incorporate testing requirements in case of any change in business requirement due to complexity in test planning/execution both from unit/regression/system test perspective and time constraint. Also due to complex menu structures and difficult caller navigation.
- **TECHNOLOGY RESTRICTION:** - IVR applications are developed using proprietary tools so it is very difficult to have an automation tool that can simulate/run the test case for proprietary tool in client specific environment. Clients have their own proprietary tool and middleware implementation e.g. AT&T has developed its own middleware implementation called datagate services.
- **WASTAGE OF EFFORT:** - Lots of effort will go if application development methods are changed, making it difficult to manage the test tools and test scripts.
- **ENVIRONMENT CONSIDERATIONS:** For automated testing a stable environment is required. Testers preset parameters and then run the automatic scripts, but if the environment changes in between (a new release/patch takes place) then scripts may not behave in the expected way and testers would need to modify the script each time. Thus the effort that goes into updating scripts is as bad/ more than the effort that goes into writing and running manual scripts.
- **CUSTOMIZATION OF TOOL:** - Deciding test case pass/fail is very difficult in testing IVR applications as it might be possible that automation tool is able to execute end to end application flow but from testers need to verify backend data/different logs/exceptions. Again Automation tool can serve the purpose by giving you all the information but tool needs to be customized for particular /company/applications/functionalities/domain etc.
- **GRAMMER ISSUES:** In case of testing voice applications, automation tools are built on some pre assumptions. If the assumption is if grammar files contain special characters like “,” , “-“, capital letters and digits(1,2,3) , these will lead to compilation error. This would lead to effort wastage in modifying all the grammar files they want to use for automation.
- **TESTING IN PRODUCTION:** - The IVR applications need to be monitored after each deployment for any defect. Developer need to make some test calls to check end to end call flow. Here we cannot use Automation tool as application will get customer details from production database (Oracle/proprietary tables/middleware etc.). To test end to end call flow in production, one cannot set test data in automation tool. Also tester needs to verify that failure of one module/application should not stop/hold the entire system/other applications.

- **REQUIREMENT SCOPE:-** Testing of IVR applications due to following requirements is very difficult, complex and almost impossible with automation tools:
  - ❖ Database migration
  - ❖ Application Migration from one technology to another e.g. AVAYA to Periphonics (Nortel proprietary tool to develop DTMF IVR applications)
  - ❖ Middleware technology change
  - ❖ Change in hardware/server technology e.g. VPS to MPS or T1 lines to ISDN lines
  - ❖ Customer issues in production - the only way most companies know anything is wrong with their IVR an application is if a customer calls to complain. A single 20-minute outage or slowdown of IVR-based applications in larger call centers can impact thousands of calls at a potential cost in the tens of thousands of dollars.
  - ❖ Testing DTMF applications interacting with speech based applications: Sometimes DTMF/legacy applications has to interact with speech applications as there could be some possible scenario's wherein organization is transforming their DTMF/legacy technology into enterprise structure and some of the functionality has not been migrated.
- **CHECK FOR OTHER PROCESSES:-** Automation tools can't check other required processes/utilities that should run for proper functionality of IVR applications e.g. sqlclnt is a process which should run on Unix servers so that IVR applications can directly interact with oracle database but in case this process is not running or in down status then automation tool will simply fail that test case saying data not found. Again effort has to be put to analyze why test case had fail. Most of the problems are not due to IVR applications/IVR components but they are due to system and network problems external to the IVR like CTI, Middleware's, Mainframe Database down etc. In these scenarios, automation tool while testing IVR applications mark test cases failed. In these scenario's, developer have to have check whether CTI is working or not, Mainframe database is up (Mainframe slowdowns and outages impact database lookups such as authenticate account, bill enquiries, bill payment etc.). Another scenario might be lines on which IVR applications are attached are not in up (running) or in busy state or whole T1 is out of order. Here also, developer/tester needs to manually check line status. Another situation is CTI is down. in this case customers will experience hold times, hang up, and try again later errors. Various network and system management tool will not be able to detect/indicate problems of these types.
- **TOOL STANDARD:-** Automation tools have been built based on some guidelines that make it difficult to test each different scenario's. Here in case of IVR testing, we cannot left out any untested scenario and put the code in production which could result in backout's.
- **ROOT CAUSE ANALYSIS:-** Even after you discover that you have a problem with one or more of your IVRs, finding exactly where the problem is and correcting it can pose a significant challenge. It can take hours just to identify the source and cause of an IVR problem. With Manual testing it is very easy to locate the bottleneck using some trace mechanism or following the call flow.
- **PERSONALIZATION TESTING:-** There is no industry standard tool available in market to test "Personalization". By way of personalization caller shall have the ability to create/modify/delete his own profile. Profile may include information relating to language option, ability to set various billing thresholds, feedback alert mode (SMS, Email, Call back) etc. Small things like personalized greeting for an authenticated customer or specialized greeting in case it happens to be callers Birthday/Wedding. Here tester has to setup/feed the personalized data and test the IVR application based on that.
- **DIFFICULTY IN TESTING SPEECH APPLICATIONS:-** It is very also very difficult to test Speech applications using automation tools due to following factors:
  - ❖ **LACK OF INTEGRATION** – VoiceXML engines, development tools, call processing framework, telephony hardware, web services and object libraries all come from different vendors. Automate testing of speech applications in this integrated environment is not practical as these are not designed to work easily together.
  - ❖ Interacting with the caller through natural language as clear and unambiguous wording of a prompt is a key contributor to application success. To automate the test cases while **SIDE SPEECH** – user's tendency to talk to others while on the call is very complex and required lots and lots of thinking. In these cases, the application will not be able to recognize the user input and goes into a rejection or error state and may take users back to the Main Menu

❖ Designing, documenting and then incorporating test cases in automation tool is very time consuming and cumbersome task as the user input is not limited to some digits. This will put extra burden on application testing and lots of effort has to put even if for simple change request.

- Lack of inconsistent standards
- Proprietary hardware, and third-party components

➤ **TESTING BIOMETRIC PATTERN:** - Automation tools are not designed to store biometric pattern for the called for Speaker Verification. Speaker Verification means an IVR caller should be able to use the IVR system, even if he doesn't remember his Account number. As IVR can automatically verify the customer based on matching the biometric pattern. This is unique like the finger prints. Investing on features like Speaker recognition.

➤ **TESTING STATE RECALL:** - Automation tools are not designed for Advance features like *State recall* which can help in the event when the transaction was incomplete because of any reason such as User hanged up. (E.g. Caller received a call on his other phone and had to drop the call in between a transaction) System issues (Call drops because of an issue with the IVR itself)

- ✓ *State recall* allows application to recall the last status and avoid user repeating the already completed steps of the incomplete transaction. This feature is very useful in application where a transaction spans across many steps such as payment/billing processing applications.

## 5. IVR Monitoring

### 5.1 Strategic focus on monitoring IVR performance & Call Quality

Did anyone think why IVR monitoring is needed to get the benefits of IVR?

In today's business development most of the organizations rely on their IVR applications to resolve customer queries, handle various transactions etc. But the question is how they would get to know if their customers are having some issues using their IVR system?

The only way to know anything about malfunctioning of IVR is if a customer comes to complaint. This represents a significant potential business risk. Organizations could loose as much as 50-70% their customer or revenues due to unde-

tected IVR transaction problems. A single 30-minute outage or slowdown of IVR applications specially billing applications in larger organizations can impact thousands of calls at a potential cost in the tens of thousands of dollars. This had occurred with lots of telecom service providers around the world.

Have you ever observed IVR problem is not due to IVR application failure but due to external problem like system and network problems like database down, updated patch applied on some operating/software system, backend interface down or due to some application dependencies etc. The most common issue that has encountered with most of the organization is backend database failure. In this case IVR applications will not be able to fetch caller data from database and result in call drop/call hang-up/call transfer to call representative, all result in loss of customer/revenue. From customer point of view, IVR application is not working properly but this is not the case.

Another potential problem could be line on which these IVR applications are running goes into down state. Here caller will hear some music or error message like "All of our customer care representatives are busy in handling another customer. Please call back late "and call will get disconnected.

Even after you realize that you have a problem with one or more of your IVR system, finding exactly where the problem is and correcting it can pose a significant challenge

Identify the serious customer service problems and higher expenses caused by "out of order" IVR applications, some organizations have started manually monitoring their call centre IVR applications with the goal of reducing IVR downtime. A group of people with knowledge of IVR functionality monitors IVR systems by dialing into each IVR application on a planned basis to ensure that everything is working properly and customers are not encountering problems. This approach, though time consuming and expensive, has helped to reduce the impact of undiscovered IVR outages. By finding and fixing IVR problems more quickly, fewer calls overflow from out-of-service IVRs to call center CSRs [7]. Keeping more calls in the IVRs reduces overall transaction costs and more than pays for the cost of physically monitoring the IVR applications.

### 5.2 Key attributes on monitoring IVR performances

Before deciding any strategy on monitoring IVR performances, It's very important to note some key attributes:-

- Focus on IVR Call Flow, errors, hang-ups, drop-out points starting from the design of the IVR application.
- Track transactions and facilitate information from customer point of view (Whether the callers planned goal was accomplished and how it impacted him/her?)
- Spend time to analyse call quality through call quality experts by recording the caller's transaction and try to understand the customer experience with the system from his/her stand point.

Implementation of adequate IVR performance & call quality monitoring tools/processes would help improving it by correcting applications, systems inefficiencies and failures. One can make IVR application less prone to fail by automating the monitoring mechanism and use of proper monitoring tools. Automate monitoring means real time monitoring of IVR calls or make anonymous calls using some predefined call flow after some time slice and send status/alerts in case of any discrepancy with what is expected.

There are tools available in market which can be used for continuous quality monitoring of IVR applications. These tools can be scheduled to dial into IVR application on a planned basis to verify everything is working accurately. After the call gets over, someone from QA team can verify if call was successful or not. They can verify it through log generated by IVR applications. These tools need IVR call flow as an input and some threshold limit so that if the IVR application exceeds the threshold limit, the monitoring team can immediately page to technical support resource to correct the problem.

The professional monitoring/testing tools can be used as an automated testing tool to test/monitor IVR/Speech applications by making calls proactively to identify the issue. These tools take end to end call flow in terms of recorded scripts along with duration, default speaks, duration as an input. Tool then periodically makes the call to production system (execute scripts) and compares recorded behavior to call behavior. In case there is any mismatch, tool sends the alarm to applicable support teams. Based on alarm, support team can analyze the alarm and fix errors. Tools also take call in number, call frequency, support team information, server information (where the call will land) as an input.

If, at any point, the monitoring team does not get the right response from the IVR or the response time exceeds a predefined limit, they can immediately call/page to technical support resources to correct the problem. The page can specify which IVR system/application the problem was encountered in and where in the call flow the problem occurred. This can mean significant time savings in finding and correcting the source of the problem and getting IVR applica-

tions back in service which would in-turn, lead to improved customer satisfaction and cost savings .

These tools can be used to performance/load testing. It requires peak amount of load to be put on production servers but amount of T1 lines/ports are limited. The number of simultaneous calls at any single point of time is directly proportional to number of ports. Moreover blocking all ports at a given time is not viable. Here we can use the load generation tool. It can be used both for DTMF as well as speech applications.

Research has shown that Using automated testing of IVR applications; many organizations have been able to reduce the time it takes to find and solve IVR problems from hours down to minutes result in end-user satisfaction and cost benefits to companies.

## 6. Conclusion

As we have seen that developers/testers, and in particular those that are new to call centre domain and telephony, have experienced difficulties while testing IVR applications especially for above mentioned functionality of IVR applications. There should be some deciding factors while deciding between Automated and Manual testing such as Extent of human interfere, Nature of application, knowledge level of testing team, time constraint, environment etc. It is good to automate the test execution but not on the expense of customer experience, complexity and cost. Customer satisfaction is a key goal of every organization. The real challenge comes in maintaining balance between customer expectations and investment in business to meet expectations. This is completely true for IVRs. By using the right testing strategies and proven best practices as illustrated in the paper, organizations can avoid difficulties that can result in financial losses and customer dissatisfaction.

## 7. References

- [1] [http://en.wikipedia.org/wiki/Call\\_centre](http://en.wikipedia.org/wiki/Call_centre)
- [2] [http://en.wikipedia.org/wiki/Proof\\_of\\_concept](http://en.wikipedia.org/wiki/Proof_of_concept)
- [3] [http://en.wikipedia.org/wiki/Interactive\\_voice\\_response](http://en.wikipedia.org/wiki/Interactive_voice_response)
- [4] [http://en.wikipedia.org/wiki/Computer\\_telephony\\_integration](http://en.wikipedia.org/wiki/Computer_telephony_integration)
- [5] [http://en.wikipedia.org/wiki/Dialed\\_Number\\_Identification\\_Service](http://en.wikipedia.org/wiki/Dialed_Number_Identification_Service)
- [6] [http://en.wikipedia.org/wiki/Java\\_Virtual\\_Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine)
- [7] <http://www.hrville.com/hrjobdesc/CallCenterRep.htm>