Are We There Yet?
Research in Commercial Spoken Dialog Systems

Roberto Pieraccini,
David Suendermann, Krishna Dayanidhi, Jackson Liscombe

SpeechCycle, Inc., 26 Broadway, 11th Fl., New York, NY
{roberto, david, krishna, jackson}@speechcycle.com

Abstract. *In this paper we discuss the recent evolution of spoken dialog systems in commercial deployments. Yet based on a simple finite state machine design paradigm, dialog systems reached today a higher level of complexity. The availability of massive amounts of data during deployment led to the development of continuous optimization strategy pushing the design and development of spoken dialog applications from an art to science. At the same time new methods for evaluating the subjective caller experience are available. Finally we describe the inevitable evolution for spoken dialog applications from speech only to multimodal interaction.*

Keywords: *spoken dialog, speech recognition, language understanding, rich phone applications, multimodal interaction. .*

1 Introduction

Four years ago, a review of the state of the art of commercial and research spoken dialog systems (SDS) was presented at the 2005 SIGdial workshop on Discourse and Dialog in Lisbon, Portugal [2]. The main point of the review was that research and commercial endeavors within the SDS technology have different goals and therefore aim at different paradigms for building interactive systems based on spoken language. While usability, cost effectiveness, and overall automation rate—which eventually affect the ROI, or Return on Investment, provided by the application—have always been the primary goals of commercial SDS, the research community has always strived to achieve user's interaction naturalness and expression freedom. We also established that the latter—naturalness and expression freedom—do not necessarily lead to the former, i.e. usability, automation, and cost effectiveness. One of the reasons of that is that today's technology cannot imitate the human spoken communication process with the same levels of accuracy, robustness, and overall effectiveness. Speech recognition makes errors, language understanding makes errors, and so user interfaces are far from being as effective as humans. For all of this technology to work, one has to impose severe limitations on the scope of the applications, which require a great amount of manual work for the designers. These limitations are often not perceived by the users and that often creates a problem. The more anthropomorphic the system, the more the user tend to move into a level of comfort, freedom of expression, and naturalness which is eventually not supported by the application, creating a disconnect between the user and the machine that leads to an unavoidable communication breakdown.

This progression would follow a curve similar to that predicted by the uncanny valley [1] theory[1], proposed in 1970 by Masahiro Mori. Michael Phillips proposed a similar trajectory, shown in Figure 1, for the usability/flexibility curve in his keynote speech at Interspeech 2006.

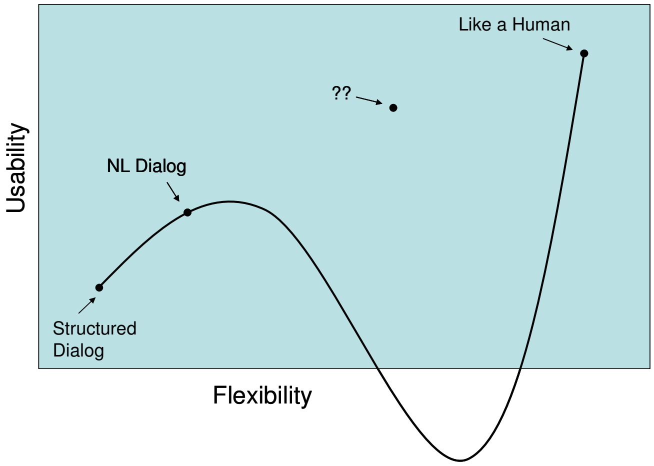

Figure 1: The uncanny valley of usability[2]

Practically, as the flexibility (naturalness and freedom of expression) of a spoken dialog system increases, so does its usability, until a point when usability starts to decrease, to increase again when the system becomes almost indistinguishable from a human. Reaching the point identified by the question marks is very difficult, practically impossible, with today's technology, and the apparent naturalness provided by advanced research prototypes—often demonstrated in highly controlled environments—inevitably drives spoken dialog systems towards the uncanny valley of usability. According to the curve of Figure 1, today we are still in a place between structured—also known as directed—and natural language (NL) dialog.

One of the conclusions from the 2005 review is that, in order to fulfill the necessary usability requirements to drive automation and customer acceptance,

[1] The uncanny valley theory states that humans respond positively to machines with an increased human likeness up to a point where it would start causing an increasing feeling of repulsion. Repulsion will start decreasing again after a certain degree of humanness is achieved by the machine, after which a positive and empathic response is produced again. The interval of human likeness in which machines provoke a sense of repulsion is called the *uncanny valley*.

[2] This chart published with permission of Mike Phillips, Vlingo.

commercial systems need to have completely defined Voice User Interfaces (VUI) following a principle called VUI completeness. An interface is VUI complete when its behavior is completely specified with respect to all possible situations that may arise during the interaction, including user inputs, and backend system responses. Today, in order to guarantee VUI completeness, it is best practice to come to a complete specification of the interface during an initial design phase. The only dialog management paradigm today that allows for producing VUI complete interfaces in an economical and reproducible way is the functional one [2] embodied by the finite state controller representation known as *call flow.* Unfortunately, other more sophisticated mechanisms, such as those based on inference or statistical models, would require much higher a complexity and costs in order to guarantee VUI completeness.

In the rest of this paper we will describe the advancements we have witnessed in spoken dialog technology during the past four years. We will put particular emphasis on complex and sophisticated systems deployed for commercial use with a large volume of interactions—i.e. millions of calls per month. The main trend in this area is an increased use of data for improving the performance of the system. This is reflected by intense usage of statistical learning—even though with quite simplistic mechanisms—at the level of the dialog structure as well as at the level of the individual prompts and grammars. To that respect, the availability of large amounts of data and the ability to transcribe and annotate it at a reasonable cost, suggests deprecating the use of traditional *rule-based grammars*, widely used in the industry, and substituting them with statistical grammars. We will then discuss the use of data as the basis for the evaluation of subjective characterization of dialog systems, such as caller experience and caller cooperation. Finally, we will conclude with a mention of the current trends, including the evolution of SDS towards multimodal interaction.

2 RPA: Rich Phone Applications

There is not an established measure of the complexity of a dialog system. As software is often measured in *lines of code,* if a system is built using the traditional call flow paradigm one can use its size, in terms of nodes and arcs, as a measure of complexity. Most of the call flows built today are based on the notion of dialog modules. A dialog module is an object, typically associated with a node of the call flow, designed to collect a single piece of information, such as a date, a number, a name, a yes/no answer, or a free form natural language input. Dialog modules include retry, timeout, and the confirmation strategy necessary to prompt for and collect the information from the user even in presence of speech recognition errors, missed input, or low confidence. The number of dialog modules used in a system is a fair indicator of its complexity.

As discussed in [3], during the past decades, we have seen dialog systems evolve through three generations of increasingly more sophisticated applications: informational, transactional, and problem-solving, respectively. Correspondingly, their complexity evolved towards deployed systems that include several hundreds of dialog modules and span dozens of turns for several minutes of interactions. Yet, the call flow paradigm is still used to build these sophisticated and complex systems.

We associate the most sophisticated spoken dialog application today to a category called Rich Phone Applications, or RPAs. RPAs are characterized by a certain number of features, including:

a) **Composition and integration with other applications**. Non integrated, or blind, spoken dialog systems provide very limited automation at the expense of poor user experience, especially for customer care applications, in analogy with human agents who do not have any access to caller information, nor to account management tools, diagnostic tools, or external knowledge sources. The development cost of providing integration of a dialog system with external backends is often higher than that of building the voice user interface. For that purpose, an orchestration layer that allows the management of external object abstractions characterizes the most advanced dialog systems today, and it is a fundamental feature of RPAs.

b) **Asynchronous interaction behavior**. Since RPAs can interact with multiple external backends, and at the same time interact with a user using spoken language, there is no reason not to carry on all these interactions, whenever appropriate, simultaneously. So, an RPA can, for instance, retrieve the caller's account from a database, check the status of his bill, and perform diagnostics on his home equipment while, at the same time, ask a few questions about the reason of the call. That will greatly speed up the interaction creating the basis for improved user experience and automation.

c) **Continuous tuning.** Spoken dialog systems, once deployed, generally do not offer the best possible performance. The main reason is that their design is generally based on a number of assumptions which are not always verified until large amounts of data are available. First, the voice user interface is often based on what is considered *best practice,* but the reality of the particular context in which the application is deployed, the distribution of the user population, their peculiar attitude towards automated systems and the provider, their language, and many other variables may be different from what was envisioned at design time. The designed prompts may not solicit exactly the intended response, and the handcrafted rule-based grammars may not be able to catch the whole variety of expressions used by the callers. Besides that, things may be changing during the course of deployment. New situations may arise with new terms—e.g. think of marketing campaigns introducing products with new names and callers requesting assistance with that—and changes made to the system prompts or logic—it is not uncommon to have deployed spoken dialog systems updated monthly or more often by the IT departments who owns them. That may impose changes on the user interface that could invalidate the existing rule-based grammars. Thus, a feature of RPAs is the ability to undergo continuous data-driven tuning with respect to the logic and the speech resources, such as prompts and grammars.

d) **Multimodal interaction.** Most of the issues related to the usability of an imperfect technology such as speech recognition would be greatly alleviated if the interaction would evolve with the aid of a second complementary modality. This has been known for a long time, and multimodal interaction has been the object of study for several decades. Most recently, W3C published a recommendation for a markup language, EMMA[3], targeted at representing the input to a multimodal system in a unified manner. However, the commercial adoption of multimodal systems with speech recognition as one of the modalities has been very limited. First of all the convenience of speech input on a PC with a regular keyboard is questionable. Speech can be useful indeed when small and impractical keyboards are the only choice, for instance on a smartphone or PDA. However, only recently, the adoption of powerful pocket devices became ubiquitous together with the availability of fast wireless data networks, such as 3G. Wireless multimodal interaction with speech and GUIs is a natural evolution of the current speech-only interaction.

3 Managing complexity

One of the enablers of the evolution of complex and sophisticated RPAs is the availability of authoring tools that allow building, managing, and testing large call flows. We can say that the most advanced tools for authoring spoken dialog systems today are those that create a meta-language description that is eventually interpreted by a runtime system, which in turn generates dynamic markup language (i.e. VoiceXML) for the speech platform. Other types of authoring models, such as for instance direct VoiceXML authoring, does not allow managing the complexity necessary to create sophisticated systems of the RPA category. The question is how did authoring evolve to allow the creation and maintenance of call flows with hundreds of processes, dialog modules, prompts, and grammars. The answer to this question relies on the consideration that a call flow is not just a graph designed on a canvas, but it is software. At the beginning of the commercial deployment of spoken dialog systems, in the mid 1990s, the leading model was that of using very simplistic *state machine* representations of the call flow, associated with very poor programming models. Most often, the call flow design and development tools did not even offer hierarchical decomposition, relying on pasting together several pages of giant monolithic charts linked by *goto* arcs. Rather, if you consider a call flow as a particular embodiment of the general category of modern procedural software, you may take advantage of all the elements that allow reducing and managing the complexity of sophisticated applications. For instance, call flow development tools can borrow most, if not all, of the abstractions used today in modern languages, such as inheritance, polymorphism, hierarchical structure, event and exception handling, local and global variables, etc. The question then is why not develop dialog applications in regular Java or C# code, rather than by using the typical drag-n-drop graphical paradigm which is being adopted by most call flow development tools. There are several answers to that. First, call flows are large hierarchical state machines. Programming large finite state machines in regular procedural languages, like Java or C#, results in an unwieldy nest—each state corresponds to a nested

[3] http://www.w3.org/TR/emma/

statement—of case statements, which is quite unreadable, unintuitive, and difficult to manage. In addition to that we have also to consider that the modern development lifecycle calls for a reduction of the number of steps required creating, deploying, and maintaining spoken dialog applications. A few years ago, it was common practice to have a full dialog specification step, generally performed by a VUI designer on paper, and followed by an implementation step, generally carried on by a software developer. That is impractical today for complex RPAs. Rather, advanced VUI designers today fully develop the application on rapid development tools on their own, while software developers just provide connectivity with the backend systems. The availability of powerful visual development tools with embedded software abstractions perfectly fit this development paradigm.

4 Continuous Tuning

Even though relying on the best design practices and the most powerful authoring tools, designers need to make assumption that may not be verified in practice. Because of that, dialog systems typically underperform when deployed for the first time. Also, as we stated early, there are environmental changes that may happen during the lifecycle of a deployed system and that can affect its performance. Because of that, the ability to perform continuous tuning is an essential feature of RPAs. There are at least two elements of a dialog system that can be tuned, namely the VUI and the speech grammars.

4.1 Data Driven VUI Tuning

With the introduction of Partially Observable Markov Decision Processes (POMDPs) [4] there has been a fundamental evolution in dialog policy learning by taking into consideration the uncertainty derived by potential speech recognition errors. While dialog learning research is developing potential breakthrough technology, commercial deployment cannot yet take advantage of it [5] because of the inherent impracticalities of learning policies from scratch through real user interactions—in fact, research is using simulated users—and the difficulty of designing reward schemas, as opposed to explicitly specifying the interaction behavior as in standard call flow development.

However, there is a lot of interest to explore *partial* learning of VUI in the commercial world of spoken dialog systems. This interest spurs from the fact that there is a clear dependency between the performance of a system (e.g. overall automation rate, caller experience, abandon rate, etc.) and the specific VUI strategy and prompts. This dependency has been always considered an expression of the VUI designer *art*, rather than a *science*. Designers have a baggage of best practice experience on how to create prompts and strategies that would solicit certain behavior, but often they have to choose among different possibilities, without being certain which is the one that would perform best in a given situation. Using data is an effective approach to this issue. Whenever alternative strategies are possible from the design point of view, one could alternate among them in a random fashion, and then measure the individual contribution to an overall evaluation criterion, such as automation rate or caller experience. One can also make the random selection of alternative strategies depend on the current interaction parameters, such as time of the day, day of the week, or some type of caller's characterization (e.g. type of subscription account, area code, etc.). Once enough data is collected, one can decide which strategy offers the best performance for each set of parameters or chose an online optimization strategy that changes the frequency of usage of each alternative strategy based on an exploration/exploitation criterion similar to that used in reinforcement learning theory.

4.2 Speech Recognition Continuous Tuning

What makes a spoken dialog system a *spoken* dialog system is its capability to interact with a user by means of input and output speech. While output speech can be produced with highest intelligibility and quality using pre-recorded prompts combined using concatenative speech synthesis, the performance of speech recognition and understanding of these systems is often subject to criticism and user frustration.

The reason for non optimal performance of commercially deployed systems is based on the common practice to consider speech recognition and understanding as an art rather than science. Voice interaction designers and "speech scientists" design rules representing utterances they expect callers to speak at the various contexts of the interaction. This rule set is generally referred to as *rule-based grammar*. Sometimes, the designers and speech scientists may listen to some live calls, collect and transcribe a limited number of utterances and tweak the grammars, pronunciation dictionaries and noise models, set thresholds, sensitivities, and time-outs most often based on anecdotic experience or best practice.

In contrast to this common *handcrafting* approach, only recently did we present a framework [8] systematically replacing all rule-based grammars in spoken dialog systems by statistical language models and statistical classifiers trained on large amounts of data collected at each recognition context. This data collection involves partially automated transcriptions and semantic annotation of large numbers of utterances which are collected over the lifecycle of an application. The statistical grammars, that replace the hand written ones, are constantly retrained and tested against the baseline grammars currently used in the production system. Once the new grammars show significant improvement, they are released into production, providing improvement of its performance over time. Figure 2 shows an implementation of this continuous improvement cycle which also involves steps to assure highest reliability and consistency of the involved semantic annotations [6].

As an example, we implemented this continuous tuning cycle for a complex deployed system consisting of three different technical support applications interconnected via an open-prompt call router. The overall accuracy went from an average initial semantic classification accuracy of 78.0% to 90.5% within three months, based on the processing of more than 2.2 million utterances.

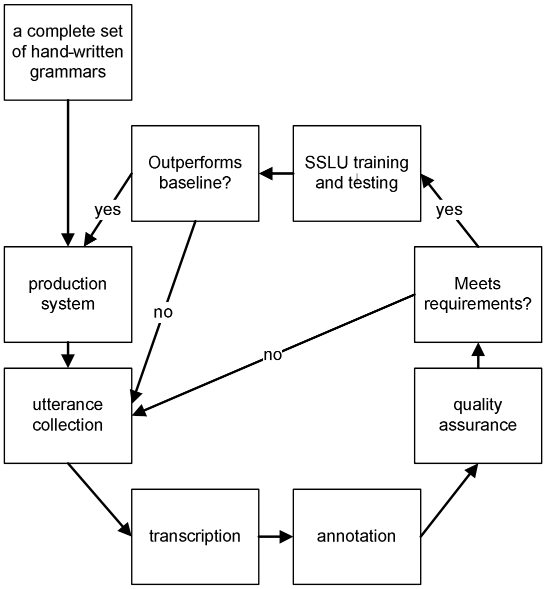

Figure 2: The continuous speech recognition tuning cycle

5 Evaluating Caller Experience

While a data-driven approach to dialog optimization is much more powerful and complete than an anecdotic approach based on human experience and best practices, it does require an objective criterion that can be easily measured. Call duration, successful automation, and utterance classification accuracy are suitable measures that can be effectively calculated without the need of human labor. However this type of objective criteria is often not sufficient to completely characterize a system's behavior with respect to the subjective experience of the user. Poorly designed systems may exhibit high completion at the expense of a poor user experience. For instance they may ignore requests for live agents, or often re-prompt and confirm the information produced by the caller leading to user frustration. They may also extensively offer DTMF options, presenting numerous levels of menus and yes/no questions, which may be very annoying for the user. One of the ultimate goals of designing and building spoken dialog systems is therefore the optimization of the *caller experience*. In contrast to impractical and often biased implementations of caller surveys we, introduced a method based on subjective evaluation of call recordings by a team of listeners. Each sample call is scored on a scale between 1 (very bad) and 5 (excellent). The average of these ratings across several calls and several listeners expresses the overall caller experience [7].

As an example, we tracked the caller experience of a complex call routing application over a three-month tuning cycle and showed that the average score improved from 3.4 to 4.6 as reported in Figure 3.

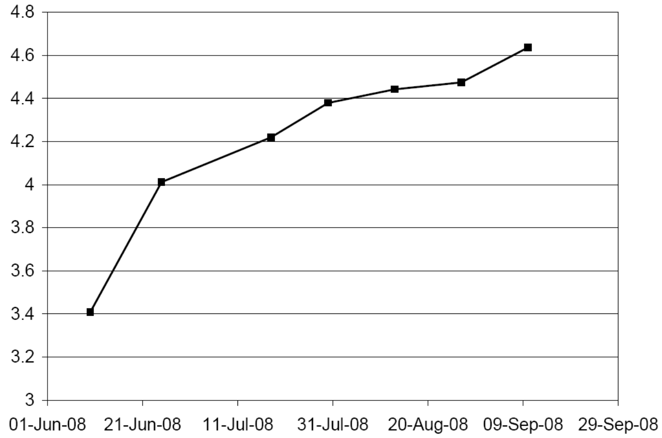

Figure 3: Evolution of caller experience over multiple versions of a spoken dialog system

However, the nature of the caller experience score as introduced above being based on subjective ratings makes this measure impractical for automated system tuning or real-time reporting. To overcome this issue, research is directed towards the prediction of caller experience based on available objective and measureable parameters such as the number of no-matches, operator requests, call duration, etc. Experiments show that automatic prediction may achieve accuracy almost as high as human experts considering substantial inter-expert disagreement.

6 Wireless Multimodal Interactions

The trend is uncontestable. Soon everyone or nearly everyone will use a smartphone as their primary means of communication. Although this becomes immediately evident by walking around or riding the subway of a city like New York, the trend is supported by analyst research. For instance, a report by Harris Interactive on telephony usage for US adults shows that between Q4 2007 and Q1 2008 89% the adoption of mobile phones is 89%, compared with 79% of landline, and 15% of internet telephony. However, the most interesting piece of data is that, still according to Harris Interactive, 14% of the adult population is composed of *cordcutters*: people who do not have a fixed landline phone at home, and for which a mobile device is their only access to the telephone network. Another report shows this number to be 18% by the end of 2008.

As POTs, or plain old telephones, are quickly replaced by powerful pocket size devices with voice, data, and display capabilities, spoken dialog systems cannot dwell on the same old "*please press or say one*" voice + DTMF interaction paradigm. The ability to compress voice on the device, send it to a remote server through a fast 3G data link, and display information at the same time will shift the spoken dialog interaction paradigm towards wireless multimodal communication with machines. It is interesting to notice that just by adding a visual component to a speech recognition system one can reduce or remove most of the issues that are present in today spoken dialog interaction systems. As we know, speech recognition errors are one of the main causes of the slow adoption of speech systems, and copying with them is part of the VUI design of an application. Given the narrow bandwidth of the speech channel, information has to be presented to users in a sequential manner. Even though the accuracy of speech recognition may be close to 100% when considering alternative ranked hypotheses from the recognizer—what is typically called N-best—they have to be presented one by one, making the user experience quite unpleasant, and giving raise to the typical negative perception acquired by speech recognition interfaces. If you wanted to take advantage of the N-best list, the only possibility in a speech-only interface is to present the various alternatives in a sequential manner, as in the following example:

Caller: Boston
System: I think you said Austin, is that correct?
Caller: No, I said Boston
System: My mistake. Was it Houston?
Caller: No … Boston!
System: Sorry. I seem to have trouble understanding you. Maybe it was Boston?
Caller: YES!

Of course this strategy may lead to unacceptably long dialogs. In fact, most of the VUI designs today avoid using the N-best list and either re-prompt the caller, or move on to a backup dialog. The situation is even more exacerbated when dealing with open prompt input, since there are more semantic possibilities, and the difference between them may be more subtle than the acoustic similarity among words (as in Boston vs. Austin).

So, speech recognition today counts on the first recognition hypothesis to be correct, while we know very well that, with the current technology, the first hypothesis may not be the correct one in a small, but still significant, number of time (e.g. between 5% and 10%). Moving from speech-only to speech+GUI interfaces, we gain the advantage of eliminating the sequentiality constraint, and allow users to *browse* the results, and select the correct response, even though it is not the first best one.

6 Conclusions

Commercial spoken dialog application evolved during the past few years both in complexity and performance. The leading dialog management design paradigm is still based on the call-flow abstraction. However, the most sophisticated design tools have embraced a call-flow-as-software paradigm, where the interaction is still represented in a graphical way as a hierarchical finite state machine controller, but the programming model is enriched with most of the primitives of modern procedural programming languages. This design paradigm allows streamlining the development of complex applications such as those in the category of problem solving. A new category of spoken dialog applications, identified as RPAs, or Rich Phone Applications, is emerging as the voice interaction analog of Rich Internet Applications. The enhanced experience of RPA users is derived by a sophisticated interaction of the application with external services in an asynchronous manner, the capability of continually adapting the system, and the use of multimodal interaction whenever possible. In particular, while traditionally commercial systems were designed based on a best practice paradigm, we see an increased use of data driven optimization of spoken dialog system. The voice user interface can be optimized by selecting among competing strategies by using an exploitation/exploration paradigm. The speech grammars, which are commonly rule-based and built by hand, are being substituted by statistical grammars and statistical semantic classifiers which can be continuously trained from data collected while the system is deployed. Finally we discussed how speech only dialog systems are evolving towards multimodal interaction systems due to the growing adoption of smartphones as a unified communication device with voice, data, and visual display capabilities.

References

1. Mori, Masahiro (1970). Bukimi no tani The uncanny valley (K. F. MacDorman & T. Minato, Trans.). *Energy, 7*(4), 33–35. (Originally in Japanese)
2. Pieraccini, R. Huerta, J., *Where do we go from here? Research and Commercial Spoken Dialog Systems*, Proc. of 6th SIGdial Workshop on Discourse and Dialog, Lisbon, Portugal, 2-3 September, 2005. pp. 1-10
3. Acomb, K., Bloom, J., Dayanidhi, K., Hunter, P., Krogh, P., Levin, E., Pieraccini, R., Technical Support Dialog Systems, Issues, Problems, and Solutions, HLT 2007 Workshop on “Bridging the Gap, Academic and Industrial Research in Dialog Technology,” Rochester, NY, April. 26, 2007.
4. Thomson, B., Schatzmann, J., Young, S. "Bayesian Update of Dialogue State for Robust Dialogue Systems." Int Conf Acoustics Speech and Signal Processing ICASSP, Las Vegas, 2008

5 .Paek, T., Pieraccini, R., Automating spoken dialogue management design using machine learning: An industry perspective, Speech Communication, Vol. 50, 2008, pp 716-729.

6. Suendermann, D., Liscombe, J., Evanini, K., Dayanidhi, K., Pieraccini, R., C^5, in Proc. of 2008 IEEE Workshop on Spoken Language Technology (SLT 08), December 15-18, 2008, Goa, India.
7. Evanini, K., Hunter, P., Liscombe, J., Suendermann, D., Dayanidhi, K., Pieraccini, R., Caller Experience: a Method for Evaluating Dialog Systems and its Automatic Prediction, in Proc. of 2008 IEEE Workshop on Spoken Language Technology (SLT 08), December 15-18, 2008, Goa, India.
8. Suendermann, D., Evanini, K., Liscombe, J., Hunter., P, , Dayanidhi, K., Pieraccini, R., From Rule-Based to Statistical Grammars: Continuous Improvement of Large-Scale Spoken Dialog Systems, Proceedings of the 2009 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP 2009), Taipei, Taiwan, April 19-24, 2009