

Uma Ferramenta para Projeto de Sistemas de Diálogos para *Call Center* Baseados em Asterisk

Estillac Borges Filho¹, Pedro Batista¹ e Aldebaro Klautau¹

¹Laboratório de Processamento de Sinais – Universidade Federal do Pará (UFPA)
Rua Augusto Correa, 1, 66.075-110 – Belém – PA – Brasil
{estillacbf, pedosb}@gmail.com, aldebaro@ufpa.br

Abstract. *This article presents DialogBuilder, an open-source Java tool that provides to its user an interface for designing dialog systems and exporting them to code that can be deployed in Asterisk, the most popular VoIP framework. DialogBuilder offers to the novice users a wizard so that they can design their own dialog system without learning the intricacies of programming for Asterisk. The software separates the stages of dialog conception and its programming and is positioned to make technically and economically viable, even for small businesses, to construct and maintain dialog systems for telephony applications.*

Resumo. *Este artigo apresenta o aplicativo DialogBuilder, uma ferramenta open-source escrita em Java que disponibiliza ao usuário uma interface para projeto de sistemas de diálogo e exportação deste para implantação no Asterisk, o mais popular framework VoIP. DialogBuilder disponibiliza um wizard para que o usuário leigo possa projetar seu sistema sem precisar aprender a programar para Asterisk. O software separa a fase de concepção do diálogo de sua codificação e se posiciona para tornar técnica e economicamente viável, mesmo para pequenas empresas, construir e manter sistemas de diálogo para aplicações telefônicas.*

1. Introdução

Softwares capazes de dialogar com os clientes são de grande valia para o *call center* de uma empresa. E a eficiência deste tipo de atendimento é proporcional à sua capacidade de processamento de linguagem natural, campo da inteligência artificial que busca viabilizar a comunicação falada entre homem e máquina [Jurafsky e Martin 2009].

Um dos maiores problemas para que uma empresa inclua em seu *call center* aplicações de diálogo como as citadas anteriormente é que não existem ferramentas simples para que elas próprias possam projetar e implantar tais sistemas, mesmo quando já existem ferramentas como o Asterisk, que fornece, inclusive, meios gratuitos para implementação e hospedagem de tais tipos de aplicação.

A contribuição deste artigo é apresentar um aplicativo desenvolvido com esse cenário em vista e uma importante característica: prover uma forma intuitiva, simples e com auxílio de reconhecimento de fala, para que o próprio usuário possa projetar e construir um sistema de diálogo a ser hospedado no *software* Asterisk. O intuito é que a ferramenta evolua e seja capaz de prover suporte a sofisticadas estratégias de

processamento de linguagem natural em ambientes de *call centers*. Deve-se notar, no entanto, que o trabalho objetiva qualificar o projeto do plano de discagem e não o atendimento telefônico, o que poderá ser feito em etapa posterior.

2. Sistemas de Diálogo e o Asterisk

Sistemas de diálogo são sistemas de computador que conversam com um ser humano, sendo capazes de reconhecer informações faladas por este e gerar informações dinâmicas para ele, seguindo o fluxo de uma conversa [Zue e Glass 2000].

O *software* livre Asterisk implementa em *software* os recursos de um PABX convencional, sendo uma poderosa e eficiente arquitetura de voz sobre IP [Qadeer e Imran 2008] [Iseki et al. 2011] [Qin 2011]. Com o Asterisk, é possível controlar um *call center* em termos de infraestrutura, através do uso de uma linguagem chamada Dialplan. Esta linguagem é poderosa, mas não é amigável, o que torna necessário desenvolver um sistema mais simples para que o leigo possa projetar seu diálogo e gerar os *scripts* para o Asterisk. Além disso, é importante ressaltar que a linguagem Dialplan sozinha não traz recursos suficientes para processamento de linguagem natural, porém através da interface AGI (*Asterisk Gateway Interface*) é possível executar código de diferentes linguagens de programação no Asterisk, o que amplia bastante suas funcionalidades.

Kamm e Walker [Kamm e Walker 1997] discutem o projeto e a avaliação de sistemas de diálogos, enquanto Goel e Bhattacharya [Goel e Bhattacharya 2010] apresentam uma visão teórica e de implementação de um sistema de diálogos no Asterisk, como aqueles gerados com o auxílio da ferramenta proposta neste artigo. Min-Jen Tsai [Min-Jen Tsai 2005] apresenta um sistema de diálogo que se utiliza do *VoiceXML*, linguagem padrão do consórcio *W3C (World Wide Web Consortium)* para o desenvolvimento de sistemas de diálogos, mas que não é utilizada pelo Asterisk. Raux e Eskenazi [Raux e Eskenazi 2007] propõem uma arquitetura, denominada Olympus, para sistemas de diálogos, diferente daquela disponibilizada pelo Asterisk.

3. Aplicativo DialogBuilder

O aplicativo DialogBuilder desenvolvido é uma aplicação *desktop*, escrita em Java. Ele provê uma ferramenta para criação e manutenção de um projeto de sistema de diálogo. Para cada projeto, o usuário deve definir e configurar um fluxo de diálogo, onde cada passo contém um módulo que representa uma ação do *software* durante o diálogo. Cada módulo deve ser configurado com as informações necessárias.

Estando o fluxo de diálogo criado e configurado, o usuário solicita que o sistema de diálogo seja gerado e o DialogBuilder gera o código do sistema, em linguagem Dialplan, a ser implantado no Asterisk. Alguns dos módulos disponíveis para uso estão mostrados na Tabela 1, sendo que novos serão adicionados ao projeto no futuro.

Tabela 1. Alguns módulos disponíveis.

Módulo	Descrição
Executar Mídia	Executa uma mídia para o cliente, como uma mensagem falada.
Pergunta Simples	Realiza uma pergunta aberta. Exemplo: “Qual o seu nome?”.

Pergunta com Múltiplas Respostas	Realiza uma pergunta com opções de resposta. Exemplo: “Deseja realizar ou consultar um pedido?”.
----------------------------------	--

Além da interface gráfica, há outra forma para que o usuário crie um projeto de sistema de diálogo. Essa alternativa é a contribuição principal do DialogBuilder: um *wizard* que utiliza reconhecimento de fala. Dessa forma, um usuário leigo que busque desenvolver um sistema de diálogo simples para o Asterisk pode executar o *wizard* do aplicativo DialogBuilder e, através de comandos de voz, definir e configurar seu projeto. Ao fim do *wizard*, o grafo representativo do sistema de diálogo é apresentado ao usuário e este pode, caso deseje, realizar novas configurações, ou então selecionar a geração de código para o Asterisk normalmente. Para reconhecimento de voz, utilizou-se o *software* Coruja, do grupo de pesquisas FalaBrasil [Neto et al. 2010]. Vale dizer que o uso de reconhecimento de voz no *wizard* não deve ser confundido com o uso de reconhecimento de voz durante a operação do sistema, como um bloco anterior ao processamento de linguagem natural, que se busca para o sistema de diálogo final a ser implantado no Asterisk e utilizado nos *call centers*. Por último, deve-se ressaltar que usar interface com voz no *wizard* é uma aposta: mesmo que outros tipos de interfaces possam ser atualmente mais eficientes, as interfaces de voz vêm evoluindo muito rapidamente, sendo tópico de diversas pesquisas de vanguarda e largamente utilizadas em tecnologias de ponta como o Google Glass.

A ideia do *wizard* é manter uma conversa com o usuário e questioná-lo a respeito do fluxo do diálogo que deseja construir. Os questionamentos seguem o fluxo desde o início e, em cada passo, as opções são oferecidas para o usuário. Um exemplo simples de execução do *wizard* desenvolvido é transcrito a seguir:

[DialogBuilder] **Olá! O que você gostaria que o sistema fizesse ao iniciar um atendimento para seu cliente? Diga “mensagem” para o sistema executar uma mensagem de boas-vindas, ou diga “pergunta” para o sistema realizar uma pergunta.**

[Usuário] **Mensagem.**

[DialogBuilder] **Ok. Por favor, selecione o arquivo a ser executado.**

[Usuário seleciona o arquivo].

[DialogBuilder] **Ok. Em seguida, diga “pergunta” para o sistema realizar uma pergunta e diga “encerrar” para o sistema encerrar o atendimento.**

[Usuário] **Encerrar.**

[DialogBuilder] **Ok. O projeto de seu sistema será gerado em seguida.**

4. Testes de Validação

A validação será baseada em sistemas de diálogo reais, em duas macroetapas: primeiro a validação da exportação do projeto do DialogBuilder para o Asterisk; e, segundo, a validação do *wizard*, incluindo o reconhecimento de fala. Para cada macroetapa, várias iterações serão realizadas, com níveis crescentes de complexidade de diálogo.

Para ambas as macroetapas, já se conseguiu com sucesso estabelecer a validação para um sistema simples onde o *software* atende o cliente, executa uma mensagem e encerra a chamada, garantindo que a arquitetura desenvolvida atende aos objetivos.

Também já se validou o reconhecedor de fala (*software* Coruja) em testes com uma gramática contendo o nome de 76 cidades, repetidos três vezes cada por um locutor em uma chamada de um *softphone* para o Asterisk, utilizando um *headset* popular. Os resultados seguem no gráfico. As três primeiras colunas mostram o percentual de vezes em que a opção falada pelo usuário foi corretamente reconhecida na primeira, somente a partir da segunda ou somente na terceira das três tentativas, respectivamente. As demais colunas mostram o percentual de reconhecimento nas três tentativas, em nenhuma, em somente uma e em somente duas delas, respectivamente. Os valores de 92% de acerto na primeira tentativa e de 80% nas três tentativas são satisfatórios.

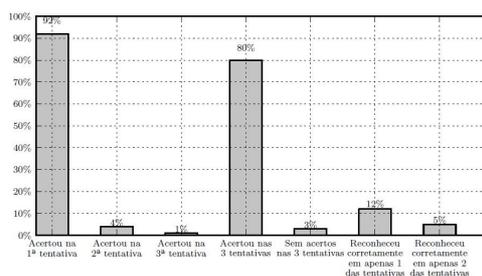


Figura 1. Resultado de teste de validação do reconhecedor de fala.

5. Contexto e Resultados Esperados

Assim como o *software* MS FrontPage rompeu o paradigma inicial de desenvolvimento de portais HTML, possibilitando ao próprio usuário criar suas páginas, sem depender de empresas com este *know-how*, o aplicativo DialogBuilder busca quebrar o paradigma de que só se é possível dotar um *call center* com uma ferramenta de diálogos com processamento de linguagem natural através de profissionais especializados. Com o DialogBuilder, qualquer empresa poderá projetar e desenvolver diretamente sistemas de diálogos a serem hospedados no Asterisk. Há soluções comerciais semelhantes, como o Visual Dialplan, porém o DialogBuilder é a única *open-source* e também possui o diferencial do *wizard*, que visa facilitar o processo, em especial para o usuário iniciante.

Por último, cabe dizer que há um sentimento comum de insatisfação de clientes com *call centers* de maneira geral. Mas, hoje estes centros são indispensáveis e deve-se melhorá-los continuamente, tornando-os mais inteligentes e próximos à linguagem natural. Isso será mais fácil quanto maior o número de ferramentas disponíveis para tal.

6. Conclusão e Trabalhos Futuros

Espera-se que o aplicativo desenvolvido traga uma real opção técnica e econômica às pequenas empresas para utilização de sistemas de diálogo com poder de processamento de linguagem natural em seus *call centers*.

Há de se atentar, contudo, para a relação de compromisso entre a qualidade do sistema projetado com o aplicativo e o baixo custo deste, pois por se tratar de um *wizard*, o produto final possui características gerais que se aplicam à maioria dos casos, mas que poderiam ser otimizadas, por um especialista, em casos específicos. Entretanto, inúmeras aplicações simples de interesse ficariam ao alcance de todo tipo de empresa.

Nas próximas etapas, novos módulos serão disponibilizados na ferramenta e incluídos como opções no *wizard*.

Referências

- Asterisk. <http://www.asterisk.org>. Acesso em Julho, 2013.
- Jurafsky, D., Martin, J. H. (2009), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice Hall, 2nd edition.
- Zue, V., Glass, J. (2000) “Conversational Interfaces: Advances and Challenges”, Invited Paper, Special Issue on Spoken Language Processing, Proc. IEEE, vol. 88, no. 8, pp. 1166-1180, August 2000.
- Qadeer, M. A., Imran, A. (2008) “Asterisk Voice Exchange: An Alternative to Conventional EPBX”, International Conference on Computer and Electrical Engineering, pp. 652-656, December 2008.
- Iseki, F., Sato, Y., Kim, M. W. (2011) “VoIP System Based on Asterisk for Enterprise Network”, 13th International Conference on Advanced Communication Technology, pp. 1284-1288, February 2011.
- Qin, D. (2011) “Research on the Performance of Asterisk-Based Media Gateway”, Fourth International Symposium on Knowledge Acquisition and Modeling, pp. 347-349, October 2011.
- Kamm, C. A., Walker, M. A. (1997) “Design and Evaluation of Spoken Dialog Systems”, Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 11-18, December 1997.
- Goel, S., Bhattacharya, M. (2010) “Speech Based Dialog Query System over Asterisk PBX Server”, 2nd International Conference on Signal Processing Systems, vol. 3, pp. 752-756, July 2010.
- Min-Jen Tsai. (2005) “The VoiceXML Dialog System for the e-Commerce Ordering Service”, Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design, vol. 1, pp. 95-100, May 2005.
- Raux, A., Eskenazi, M. (2007) “A Multi-layer Architecture for Semi-synchronous Event-driven Dialogue Management”, Proceedings of the 2007 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 514-519, December 2007.
- Grupo FalaBrasil. <http://www.laps.ufpa.br/falabrasil>. Acesso em Julho, 2013.
- Neto, N., Silva C., Batista P., Klautau, A. (2010) “Um Reconhecedor de Voz Livre para Português Brasileiro com Interface de Programação”, Agosto 2010.
- Google Glass. <http://www.google.com/glass/start>. Acesso em Julho, 2013.
- Visual Diaplan for Asterisk. <http://apstel.com>. Acesso em Julho, 2013.