

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

Ferramenta Gráfica de Criação de Unidades de Resposta Audível para o Servidor VoIP Asterisk

Guilherme Vieira

Trabalho de conclusão de curso submetido à
Universidade Federal de Santa Catarina
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de
Informação.

Orientador: Prof. Roberto Willrich, Dr

**Florianópolis - SC
2007 /2**

Guilherme Vieira

**Ferramenta Gráfica de Criação de Unidades de Resposta
Audível para o Servidor VoIP Asterisk**

Trabalho de conclusão de curso submetido à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Roberto Willrich, Dr.

Universidade Federal de Santa Catarina
willrich@inf.ufsc.br

Banca examinadora

Prof. Vitório Bruno Mazzola, Dr.

Universidade Federal de Santa Catarina
mazzola@inf.ufsc.br

Prof. Mário Antônio Ribeiro Dantas, Dr.

Universidade Federal de Santa Catarina
mario@inf.ufsc.br

Prof. Gianfranco Muncinelli, M. Sc.

Fundação Universidade do Contestado
gianfranco@ieee.org

AGRADECIMENTOS

Agradeço primeiramente a esta pessoa especial, Karla, que Deus colocou em meu caminho. Karlinha, seu apoio durante toda esta caminhada foi simplesmente fundamental.

A minha família, minha mãe Tânia, meu pai Elias, foram eles que me ensinaram os caminhos a seguir, e sempre cooperaram para que os mais diversos desafios fossem superados. Extendo este agradecimento a minha irmã, Priscila, que sempre torceu para o meu sucesso.

Obrigado ao meu Orientador, Roberto Willrich, pela grande oportunidade de aprendizado, a disposição para auxiliar nos momentos de dúvida, e principalmente a paciência durante toda esta empreitada.

Aos membros da banca, os professores, Gianfranco Muncinelli, Mário Antônio Ribeiro Dantas e Vitório Bruno Mazzola, que aceitaram fazer parte deste trabalho, ajudando a lapidá-lo com críticas e sugestões.

A turma SIN032, em especial ao Jackson Luiz Silva e Rafael Besen, que compartilharam boa parte das dificuldades, anseios, tristezas e alegrias no decorrer deste curso. Pessoal a caminhada está apenas começando.

Aos colegas de trabalho da BrasilTelecom, pela compreensão nos momentos de ausência, necessário para a conclusão de mais esta etapa.

A todos que contribuíram, direta ou indiretamente, um muito obrigado de coração !

RESUMO

O conceito de convergência, como integração de telefonia (comutada) com rede de dados, esta propiciando uma revolução em ambos os setores. Dentro desta revolução, a aplicação que possui maior destaque é o VoIP, que além da possibilidade de comunicação, alia a diminuição significativa dos custos com ligações de longa distância.

Um dos agentes que possui maior identidade com o conceito acima exposto é o Asterisk, um sistema de Ramais Privados (PABX), *open source*, de grande aceitação no mercado, que possibilita uma vasta gama de aplicações, sem a necessidade do usuário pagar as enormes quantias que seriam necessárias caso a solução PABX do mesmo fosse proprietária.

Neste contexto, este trabalho abrange a criação de uma ferramenta para desenvolvimento ágil de Unidade de Resposta Audível (URA) no sistema Asterisk, possibilitando que usuários leigos possam construir suas URA's (umas das aplicações de maior custo em soluções proprietárias), de modo gráfico, sem a necessidade de conhecer a fundo as aplicações do Asterisk.

Palavras-chave: URA, Asterisk, VoIP, telefonia, XML

ABSTRACT

The concept of convergence, as an integration of telephony (commutated) with data network, is providing a revolution in both sectors. Within this revolution, the most highlight application is VoIP, which allies the possibility of communication with a significative decreasing of costs on long distance calls.

One of the agents which most identifies with the concept above is the Asterisk, a PABX, open source, with large acceptance in the market, which makes possible a great range of applications without demanding huge amounts of money, that would be necessary if the PABX was a private one.

In this context, the scope of this work includes the creation of a tool for the quick development of an IVR in the Asterisk system, allowing "rooky" users to build their own IVR's (One of the most expensive private solutions), graphical, requiring no large know-how on Asterisk applications.

Key Words: IVR, Asterisk, VoIP, Telephony, XML

SUMÁRIO

1	<i>Introdução</i>	11
1.1	Objetivos.....	12
2	<i>VoIP</i>	14
2.1	Tipos de Implantação de VoIP.....	15
2.2	Protocolos de Sinalização	16
2.2.1	SIP	16
2.2.2	H.323	16
2.3	CODEC de Voz	17
3	<i>Unidade de Resposta Audível</i>	18
3.1	Aplicações.....	18
4	<i>Asterisk</i>	20
4.1	Arquitetura.....	20
4.2	Funções	21
4.3	Hardware.....	23
4.4	Protocolos suportados	23
5	<i>Unidade de Resposta Audível no Asterisk</i>	25
5.1	Exemplos de URA's.....	26
5.1.1	URA sem suporte a banco de dados no Asterisk.....	26
5.1.2	URA com suporte a banco de dados no Asterisk.....	28
6	<i>Ferramentas utilizadas para criar URA no Asterisk</i>	34
6.1	Visual Dialplan Beta.....	34
6.2	Dialplanner.....	36
6.3	ACTOS - Asterisk Configuration Tool Open Source.....	38
7	<i>Ferramenta para desenvolvimento de URA no Asterisk</i>	41
7.1	Metodologia.....	41
7.2	Modelo Gráfico para Especificação de URAs	42
7.3	Modelo de URA em XML	44
7.4	Traduzindo um fluxograma de URA para XML	46
7.5	Convertendo um arquivo XML para o formato padrão do Asterisk.....	47
7.6	Desenvolvimento da Ferramenta	50
8	<i>Conclusão e Trabalhos Futuros</i>	51
8.1	Conclusão	51
8.2	Trabalhos Futuros.....	51
	ANEXOS	53

<i>ANEXO 1 – Ferramenta Gráfica de Criação de Unidades de Resposta Audível para o Servidor VoIP Asterisk</i>	53
<i>Referências Bibliográficas</i>	72

LISTA DE FIGURAS

Figura 4.1- Arquitetura do Asterisk.....	21
Figura 6.1 - Visual Dialplan / Menu Principal	34
Figura 6.2 - Visual Dialplan / Menu para edição do arquivo extension.conf.....	35
Figura 6.3 - Visual Dialplan / Menu de configuração da Aplicação Read.....	35
Figura 6.4 - Dialplanner / Menu Principal.....	37
Figura 6.5 - ACTOS / Menu Principal	38
Figura 6.6 - ACTOS / Menu de localização dos arquivos de configuração	38
Figura 6.7 - ACTOS / Menu para edição do arquivo extension.conf.....	39
Figura 6.8 - ACTOS / Menu do arquivo extension.conf (seqüência de execução)	40
Figura 7.1 - Fluxograma URA - Previsão Meteorológica	43
Figura 7.2 - Fluxograma URA - Tele Voto	43
Figura 7.3 - URA Tele Voto (utilizando o modelo proposto)	49

LISTA DE TABELAS

Tabela 2-1 Principais CODEC's suportados pelo Asterisk.....	17
Tabela 7-1 Principais símbolos utilizados em um fluxograma.....	42
Tabela 7-2 Especificação das traduções (Fluxograma - XML - Asterisk)	48

ACRÔNIMOS

ATA	Adaptador de Telefone Analógico
CODEC.....	Codificador/Decodificador
CPA	Controle por Programa Armazenado
CPF.....	Cadastro de Pessoa Física
DAC	Distribuição Automática de Chamadas
DDR.....	Discagem Direta ao Ramal
DSP.....	Processador de Sinais Digitais
IAX.....	Inter-Asterisk
IP	Internet Protocol
ITU-T	International Telecommunications Union
MCU	Multiport Control Unit
PABX	Private Automatic Branch eXchange
PCM	Pulse Code Modulation
PSTN	Public Switched Telephone Network
RFC	Request for Comment
RTP.....	Real Time Protocol
SIP	Session Initiation Protocol
URA	Unidades de Resposta Audível
VoIP	Voice over Internet Protocol
XML.....	eXtensible Markup Language

1 Introdução

As redes de dados e telefonia sempre trilharam caminhos distintos, usando suas próprias infra-estruturas. Isto impacta no aumento dos custos, devido à necessidade de instalação de dois serviços, além de haver necessidade de monitorar a funcionalidade de ambos (COSTA, 2003). Este cenário está em processo de mudança desde o surgimento do conceito de convergência, permitindo integrar a telefonia e a rede de dados em grandes empresas do ramo das comunicações, gerando volumosos investimentos em pesquisas.

Segundo Ferreira et al (2006), com a evolução das tecnologias de comunicação e a melhoria do desempenho nas transmissões de dados através das redes IP (*Internet Protocol*), várias aplicações que permitem a comunicação e a interação entre pessoas geograficamente distantes estão sendo desenvolvidas.

Dentre as aplicações que permitem a comunicação e pertencem ao conceito de convergência, o VoIP (*Voice over Internet Protocol*) surge como destaque, pois o mesmo propicia além da simples comunicação, grande interatividade, além da diminuição significativa nos custos com ligações interurbanas (FERNANDES, 2000).

O telefone (fixo ou celular), ainda representa um número maior de usuários, frente ao número de computadores no Brasil. Devido a este fato, várias aplicações são ainda desenvolvidas utilizando URA (Unidade de Resposta Audível) visando interagir com usuário, repassando informações ora cadastradas em um banco de dados, como por exemplo, as consultas de saldo em bancos ou a declaração de imposto de renda via serviço ReceitaFone (RECEITAFONE, 2007).

Baseado no contexto de convergência, em 1999 surgiu um Sistema de Ramais Privados (PABX) de fonte aberto, denominado Asterisk (Asterisk, 2006), que segundo seu criador Mark Spencer, é capaz de fazer qualquer coisa com uma ligação ao colocá-la dentro do computador, portanto capaz de criar soluções altamente personalizáveis, sem a necessidade de arcar com os altos custos provenientes desta, uma vez que é de conhecimento que a indústria das telecomunicações agrega valores considerados altos para suas soluções, o que

inviabiliza os usuários com recursos financeiros reduzidos de possuir um PABX de pequeno porte adequado as suas necessidades.

Unidade de Resposta Audível (URA) é utilizada para o atendimento eletrônico ou encaminhamento de ligações. De acordo com a sua estrutura, é capaz de disponibilizar menus de navegação, tais como músicas ou mensagens pré-gravadas. A possibilidade de explorar a tecnologia VoIP com URA, possibilita a criação de várias aplicações, tais como: recepcionista digital, tele avisos (com mensagens pré-programadas), etc. Todavia a integração de Unidade de Resposta Audível com banco de dados expande exponencialmente a possibilidade de criar aplicações de maior complexidade e conseqüente utilidade, com a vantagem de não necessitar de alto investimento financeiro. Aplicações, como consulta de saldo de bancos, serviço de meteorologia, telemensagens, ou qualquer outro tipo de aplicação em que a informação de saída possa ser transformada em som, podem ser baseadas em Asterisk, com URA's configuradas para acessar bases de dados.

Atualmente o Asterisk não possui uma ferramenta livre (código e distribuição) que possibilite a construção de URA, obrigando o usuário que necessite desta facilidade estudar o funcionamento da plataforma, buscando entender os conceitos e configurações que estão por trás de uma Unidade de Resposta Audível, ou adquirir soluções proprietárias.

O presente trabalho propõe uma solução para desenvolvimento ágil de criação de URA's utilizando o Asterisk, baseado na extensibilidade deste sistema, além da integração deste com um servidor WEB.

1.1 Objetivos

Este trabalho visa efetuar o estudo e desenvolvimento de um *framework* para desenvolvimento de Unidade de Resposta Audível integrada com banco de dados e servidores Web. Mais especificamente os objetivos são:

- Estudar o funcionamento e os padrões de configurações do Asterisk.

- Modelar e implementar um *framework* para desenvolvimento de Unidade de Resposta Audível.
- Modelar a base de dados que será utilizada para armazenar as informações.
- Desenvolver aplicações que utilizam URA com consultas a base de dados.
- Validar a implementação, configurando uma URA via Internet.

O framework será desenvolvido de modo a compreender as seguintes características:

- Sistema deve propiciar agilidade na configuração da URA, sem a necessidade de conhecimentos do usuário em relação ao funcionamento do Asterisk.
- Possibilitar a implementação da URA, independente de localização geográfica, bastando ao usuário ter um computador com acesso a Internet.

2 VoIP

Um dos ramos de maior evolução atual é os das telecomunicações. Dentro deste, destaca-se a Telefonia, esta iniciou em 1876, com a invenção do telefone por Alexander Graham Bell. Nesta época, para que um telefone pudesse comunicar-se com outro, era necessário um par de fios dedicados, o que onerava consideravelmente a instalação e o crescimento deste tipo de comunicação.

Para solucionar este entrave, foram inseridas no contexto da comunicação as centrais de comutação. Elas operavam conectadas a todos os usuários via par metálico (conceito ainda utilizado atualmente), interconectando fisicamente os pares envolvidos em uma chamada. Com isto, o custo com cabos caiu drasticamente, porém a comutação ainda era lenta e totalmente mecânica.

Seguindo a cronologia dos fatos, a digitalização da telefonia foi o novo passo desta cadeia. Conforme Pinheiro (2006), a passagem para a concepção digital iniciou nos sistemas de transmissão, utilizando a técnica de digitalização PCM (*Pulse Code Modulation*), paulatinamente foi inserida nas centrais telefônicas, por meio de equipamentos com tecnologia CPA (Controle por Programa Armazenado). Nesta etapa, com os sinais já digitalizados, as portas para o mundo da convergência estavam abertas, bastando neste momento a vontade política das empresas que dominavam o mercado das telecomunicações aderir a esta tecnologia.

É nesta fase que o VoIP inicia a atual revolução. Com a evolução das redes (dados e voz), uma evolução natural seria a convergência. Esta, devido a sua expansão, consolida-se como o próximo passo na evolução das telecomunicações. Um dos expoentes na era da convergência são as aplicações de Voz que utilizam a rede IP, denominadas aplicações VoIP.

Um dos pontos fundamentais para a adoção das tecnologias VoIP é a possibilidade de redução significativa nos gastos com ligações (via PSTN) e a facilidade na comunicação entre os usuários do sistema (COSTA, 2003).

Assim como a telefonia celular, o VoIP possibilita a transparência na localização do usuário, uma vez que o mesmo acessando a Internet, pode

registrar-se em um servidor VoIP, sendo este utilizado para localizar o usuário no caso de uma chamada.

Com o crescente desenvolvimento das redes sem fio, é cada vez mais aparente a possibilidade de crescimento de terminais móveis VoIP, o que causaria um impacto também no mercado de telefonia celular, até então um dos menos afetados com o crescimento das redes convergentes.

No presente capítulo, serão apresentados os tipos de implantação de VoIP, os principais protocolos utilizados nas comunicações VoIP e uma pequena explanação sobre Codec.

2.1 Tipos de Implantação de VoIP

Segundo Costa (2003), existem duas possibilidades de implantação da tecnologia VoIP:

- **Serviço puramente VoIP** – Todos os usuários do sistema estão conectados a Internet (ou Intranet), neste modelo os PABX's (*Private Automatic Branch eXchange*) são substituídos, sendo inseridos novos elementos que possibilitam a comunicação (servidor VoIP, *softphones*, ATA – Adaptador de Telefones Analógicos, etc). A vantagem está relacionada com a eliminação dos gastos com a infra-estrutura de uma rede telefônica, tais como: PABX, aparelhos telefônicos e cabeamento;
- **Gateways de voz** – Equipamentos são utilizados para interligar a rede IP a PSTN, podendo também interligar PABX tradicionais geograficamente distantes. Neste caso, a vantagem está na economia com ligações que normalmente passariam pela PSTN, sejam locais ou longas distâncias.

O uso da tecnologia VoIP implica na utilização de alguns protocolos, necessários para estabelecer o tráfego de Voz na rede IP.

2.2 Protocolos de Sinalização

Necessários para controlar os fluxos persistentes de mídia (bidirecionais) que transportam a informação, são vários os protocolos que viabilizam a comunicação VoIP.

2.2.1 SIP

O protocolo SIP (*Session Initiation Protocol*), inicialmente proposto pela RFC (*Request for Comment*) 2543 de 1999, sofreu várias atualizações, que levaram a uma nova publicação chamada RFC 3261 de 2002. Este protocolo atua na camada de aplicação, sendo sua função iniciar, estabelecer, modificar e encerrar as sessões da rede IP.

Atualmente, a principal utilização do protocolo são as chamadas VoIP, todavia teleconferências são suportadas pelo SIP (ROSENBERG, 2001).

De acordo com Bortoluzzi (2005), SIP permite ao participante encontrar e negociar parâmetros de uma sessão multimídia com outro participante. Trabalhando em conjunto com todos os protocolos desenvolvidos para enviar dados em sessões multimídia em tempo real, o SIP faz parte de uma arquitetura multimídia completa. Todavia, a função básica do protocolo independe da utilização dos demais, devido o mesmo não oferecer serviços, apenas primitivas que diferentes serviços podem utilizar.

2.2.2 H.323

O padrão H.323 é integrante da família de recomendações do ITU-T (*International Telecommunications Union*) denominadas H.32x, cuja função principal é prover serviços de multimídias sobre diversos tipos de redes. Este padrão descreve detalhadamente como a comunicação multimídia transcorre entre terminais do usuário, equipamentos de rede e vários serviços nas redes IP. É

tratado como uma especificação de sistema, pois sua documentação faz menção a outras recomendações (CORIOLANO, 2003).

Uma comunicação H.323 sempre ocorre entre de os seguintes elementos: Terminal, *Multiport Control Unit* (MCU), *Gateway* e *Gatekeeper*, sendo que estes podem estar fisicamente em um mesmo dispositivo (*hardware*), pois podem ser apenas componentes de software.

2.3 CODEC de Voz

Inicialmente, o termo CODEC fazia referência ao par COdificador/DECodificador, sendo a função do mesmo efetuar conversões entre sinais analógicos e digitais. Entretanto, no contexto atual está mais relacionado com COmpressão/DEsCompressão (SMITH, MEGGELEN e MADSEN, 2005).

A Tabela 2.1 lista com os principais CODEC's suportados pelo Asterisk:

CODEC	Taxa de transferência (Kbps)	Licença
G.711	64	Não
G.7216	16,24 ou 32	Sim
G.723.1	5,3 ou 6,3	Sim
G.729A	8	Não
GSM	13,3 ou 15,2	Não
iLBC	Entre 2,15 e 22,4	Não
Speex	Entre 2,15 e 22,4	Não

Tabela 2-1 Principais CODEC's suportados pelo Asterisk
Fonte: Adaptado de SMITH, MEGGELEN e MADSEN, 2005

3 Unidade de Resposta Audível

Segundo Barreiros (2004), uma Unidade de Resposta Audível é utilizada para o atendimento eletrônico das ligações. É capaz de disponibilizar menus de navegação, tais como músicas ou mensagens pré-gravadas.

As solicitações feitas a esta, podem direcionar o usuário ao destino final ou especificar (triar) a chamada, via níveis de questionamento da mesma, com isto o usuário final pode ser direcionado ao destino desejado sem interferência humana direta.

O direcionamento automático acaba por gerar um alto ganho, uma vez que não é necessário alocar usuários para filtrar as chamadas, e estes podem ser utilizados para outras atividades. Além disto, evita o transtorno de transferir uma chamada várias vezes, o que acaba gerando um desconforto ao originador da chamada.

Outro ponto que merece destaque, é que ao contrário de países desenvolvidos, o número de computadores frente aos de telefones é relativamente pequeno no Brasil, portanto o acesso a sistemas via telefone ainda é um mercado em expansão (GONÇALVES, 2005).

3.1 Aplicações

Para usuários que não possuem uma faixa de número externo da PSTN, DDR – discagem direta ao ramal, a URA pode viabilizar o acesso indireto ao destino, sem que a ligação passe pela entidade telefonista (humana). A chamada pode ser atendida pela URA e questionar qual o ramal que o usuário deseja chamar, após receber as informações digitadas, a chamada é encaminhada para o destino.

Aplicações complexas, com vários níveis de questionamentos, são normalmente adotadas por *callcenter* de grande porte, onde a moeda corrente é o tempo de ocupação do usuário, sendo que neste caso, é possível diminuir os tempos de atendimento, pois as chamadas já foram triadas, possibilitando um

maior tempo livre dos atendentes (agentes) e um aumento significativo de capacidade de atendimento do callcenter.

Em situações onde a presença humana é necessária apenas para digitar números de acesso, ou rotear as chamadas para uma aplicação que vocaliza a solicitação de um cliente, a URA pode ser utilizada para solicitar as cifras de acesso e via integração com a aplicação de vocalização (normalmente um banco de dados com mensagens pré-configuradas), repassar a informação solicitada pelo cliente. Neste contexto está enquadrado as URA's de bancos que disponibilizam o serviço de consulta de saldo, ou lojas que consultam se o CPF (Cadastro de Pessoa Física) do cliente está incluso nos organismo de proteção ao crédito.

4 Asterisk

O Asterisk (Asterisk, 2006) é um PABX completo em software, baseado no paradigma *Open Source*. Criado em 1999 por Mark Spencer, é apontado como um dos principais agentes da atual revolução nas telecomunicações. Originalmente desenvolvido no sistema operacional Linux, seu nome é uma alusão ao caractere * (asterisco), que é máscara que representa qualquer nome de arquivo no referido sistema.

Atualmente é suportado pelos seguintes sistemas operacionais: Linux, FreeBSD, OS X e Windows.

O objetivo deste capítulo é apresentar o Asterisk, demonstrando sua arquitetura modular, uma explanação sobre as principais funcionalidades, além de destacar os protocolos suportados pelo sistema, com especial destaque ao IAX.

4.1 Arquitetura

Segundo Gonçalves (2005), a arquitetura do Asterisk pode ser subdividida, conforme Figura 3.1, sendo três os principais conceitos incorporados a esta arquitetura: canais, codecs e aplicações. Como canal, é entendido toda a entrada (ou saída) de um sinal (digital ou analógico) de qualquer interface do Asterisk. O Codec, trata-se do elemento que codifica e decodifica as informações das comunicações VoIP, responsável por adaptar a quantidade de bits que serão transmitidos para a rede, sempre com a meta de deixar a qualidade de voz o mais próximo possível do natural (ou pelo menos o padrão já alcançado pela telefonia tradicional). Já as aplicações, são as funcionalidades acopladas ao sistema, tais como: correio de voz, conferência, etc.

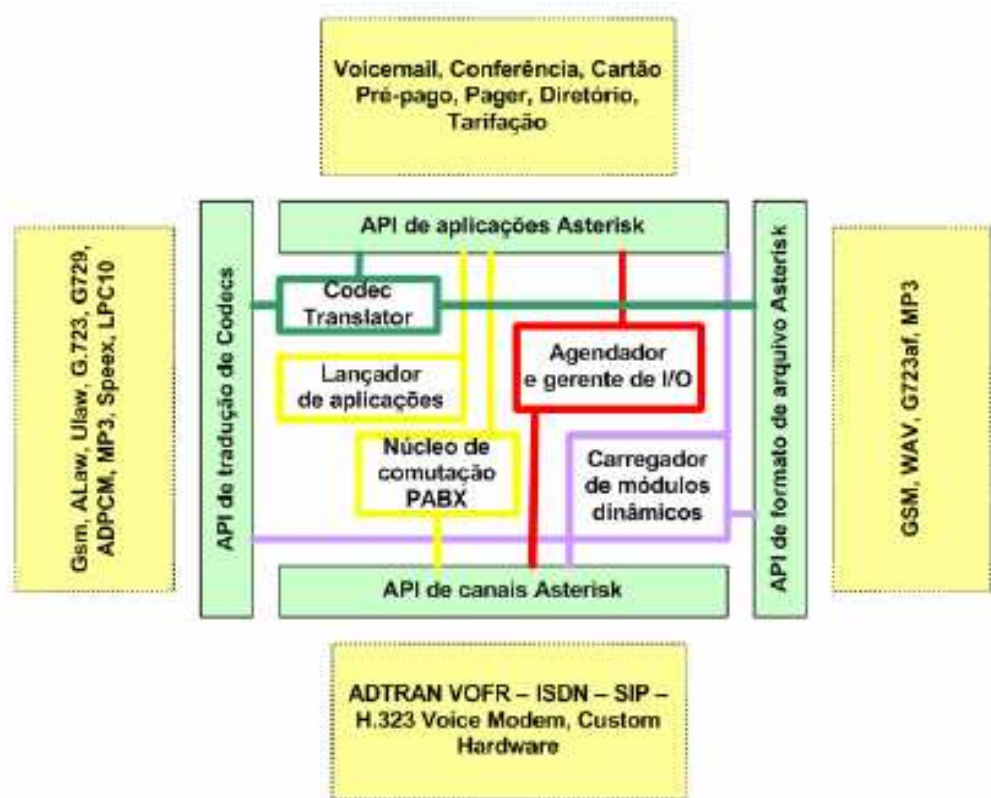


Figura 4.1- Arquitetura do Asterisk

Fonte: GONÇALVES, 2005

4.2 Funções

As principais funcionalidades incorporadas ao Asterisk, que fazem-no despontar como um equipamento de ponta, são:

- **Correio de voz:** semelhante a uma caixa postal tradicional, porém ao invés cartas, armazenas mensagens de voz. Esta mensagem pode ser enviada ao usuário via e-mail;
- **Unidade de Resposta Audível (URA):** atendimento eletrônico das ligações. Possibilidade de integração com outras aplicações e bancos

de dados, além de prover a criação de menus de navegação, e reconhecimento de voz;

- **Distribuidor Automático de Chamadas (DAC):** Distribui as chamadas entrantes no dispositivo (grupo ou serviço), utilizando algoritmos de distribuição – maior tempo livre ou menor tempo acumulado – aos agentes que estão logados no respectivo serviço. Utilizado em *Call Centers*. Possibilita o gerenciamento de filas, prioridades das chamadas e direcionamento das chamadas para agentes específicos;
- **Conferência de áudio:** Criar salas de conferência que possibilitam vários usuários conversarem simultaneamente;
- **Discador automático:** Aplicação que gera chamadas a partir de uma base de dados (contendo número de telefones), direcionando a mesma para determinado ramal ou agente;
- **Servidor de música de espera:** Vários formatos de arquivos podem ser reproduzidos pelo Asterisk, de forma síncrona ou assíncrona. Essas reproduções normalmente ocorrem ao colocar uma chamada corrente em espera, ou até mesmo antes do atendimento da chamada (quem originou a chamada pode ouvir a música enquanto aguarda o atendimento);
- **Registro detalhado das ligações:** Relatórios completos sobre as ligações (duração, origem, destino, custo, etc);
- **Ramal local e remoto:** Independência geográfica, dois ramais podem estar geograficamente distantes, porém podem fazer parte do mesmo servidor (desde que exista acesso ao servidor via rede IP);

- **Interoperabilidade com diferentes padrões de VoIP:** O Asterisk suporta praticamente todos os protocolos utilizados atualmente em telefonia e VoIP, portanto a migração e interligação com sistemas híbridos é altamente facilitada.

4.3 Hardware

Para aplicações de puramente VoIP, o Asterisk não necessita de itens de hardware especiais. Todavia é importante ressaltar que o sistema utiliza os recursos da CPU do PC onde está instalado para processar os canais de voz, ao contrario de PABX tradicionais que utilizam DSP (Processador de Sinais Digitais) dedicados em seus equipamentos, para efetuar esta tarefa. Portanto para que aplicações possam retirar o máximo do Asterisk, a necessidade de instalar os servidores Asterisk em uma rede isolada para aplicações VoIP, evitando que o servidor pabx seja alvo de vírus ou qualquer outro tipo de ataque pela rede.

Para interconexão com a PSTN, a necessidade de instalar as interfaces no sistema, sejam elas digitais ou analógicas.

4.4 Protocolos suportados

O Asterisk suporta os principais, senão todos, protocolos utilizados atualmente nas comunicações VoIP. Além dos já citados H.323 e SIP, o Asterisk implementa o IAX (*Inter-Asterisk*). Este último protocolo, segundo Smith, Meggelen e Madsen (2005), foi concebido para possibilitar a comunicação entre servidores Asterisk, sendo este o motivo do seu acrônimo. A desenvolvedora do protocolo é a Digium (DIGIUM, 2006), mesma companhia que criou o Asterisk.

O IAX um protocolo de transporte similar ao SIP, sendo uma de suas principais características utilizar apenas uma porta para comunicação, tanto para o RTP (*Real Time Protocol*), como para a sinalização do canal. Esta característica

acaba por auxiliar a configuração do firewall, pois apenas uma porta necessita ser configurada para liberar a comunicação.

5 Unidade de Resposta Audível no Asterisk

O Asterisk permite criar desde URA's simples, que apenas divulgam mensagem e encaminham para determinados ramais (recepcionistas digitais), bem como a criação de Unidades de Resposta Audíveis com alto grau de complexibilidade, envolvendo desde conexão com banco de dados e aplicações externas, até mesmo reconhecimento de voz.

O desenvolvimento de aplicações de URA no Asterisk exige conhecimento da arquitetura do sistema por parte do usuário, devido à necessidade de editar arquivos de configuração manualmente, pois o sistema não dispõe de uma aplicação que facilite esta atividade (objetivo deste trabalho).

Uma possibilidade de criar URA no Asterisk (manualmente) é editando o arquivo de configuração `extension.conf`. Este arquivo é responsável pelo plano de discagem da plataforma (controla todas as chamadas).

O arquivo é estruturado de forma a conter contextos, sendo este o responsável pelo controle das chamadas entrantes nos canais.

Cada extensão é nomeada da seguinte maneira:

[ExemploNomeExtensão]

Um contexto no Asterisk segue este modelo:

<code>exten => [chave],[prioridade],[aplicação]</code>

Onde:

- **Chave:** Número digitado pelo originador da chamada. Esta é a chave que determina que a linha em questão será executada;
- **Prioridade:** Prioridade na ordem da execução das aplicações. Não necessitam ser consecutivas, porém são executadas em ordem crescente;

- **Aplicação:** Segundo Gonçalves (2005), as aplicações são fundamentais ao Asterisk, pois é esta que trata definitivamente a chamada, seja tocando música, gravando uma mensagem ou terminando uma chamada.

5.1 Exemplos de URA's

Seguem dois exemplos de URA's, mostrando os comandos que um usuário deve dominar para construir manualmente uma Unidade de Resposta Audível na plataforma Asterisk.

5.1.1 URA sem suporte a banco de dados no Asterisk.

Para construir uma URA sem suporte a banco de dados, a instalação padrão do Asterisk é o único requisito que deve ser contemplado.

Em seguida, será demonstrado como criar uma URA que disponibilize um serviço de consulta sobre meteorologia e tabua da maré, ambos limitados geograficamente ao estado de Santa Catarina (podendo esta restrição ser superada, com as devidas alterações na estrutura que será apresentada em seguida).

Implementação (via arquivo extension.conf)

A figura 7.1 Fluxograma URA – Previsão Meteorológica, apresenta o fluxograma desta implementação.

```

exten => 9000,1,Goto(ura-previsao,s,1)
[ura-previsao]
exten => s,1,PlayBack(msg-abertura)
exten => 1,1,Goto(serv-metereologia,s,1)

[serv-metereologia]
exten => s,1,Background(sel-serv-met-regiao)
    exten => 1,1,PlayBack(serv-met-litoralsul,s,1)
    exten => 1,2,Goto(serv-met-menu2,s,1)
    exten => 2,1,PlayBack(serv-met-planaltosul,s,1)
    exten => 2,2,Goto(serv-met-menu2,s,1)
    exten => 3,1,PlayBack(serv-met-grandefpolis,s,1)
    exten => 3,2,Goto(serv-met-menu2,s,1)
    exten => 4,1,PlayBack(serv-met-altovaladoitajai,s,1)
    exten => 4,2,Goto(serv-met-menu2,s,1)
    exten => 5,1,PlayBack(serv-met-meiooesteevaleriodopeixe,s,1)
    exten => 5,2,Goto(serv-met-menu2,s,1)
    exten => 6,1,PlayBack(serv-met-oeste,s,1)
    exten => 6,2,Goto(serv-met-menu2,s,1)
    exten => 7,1,PlayBack(serv-met-planaltonorte,s,1)
    exten => 7,2,Goto(serv-met-menu2,s,1)
    exten => 8,1,PlayBack(serv-met-litoralnorteemediovale,s,1)
    exten => 9,2,Goto(ura-previsao,s,1)

[serv-met-menu2,]
    exten => 1,1,Background (msg-menu2)
        exten => 1,1,Goto(ura-previsao,s,1)
        exten => 2,1,Goto(rmsg-encerramento,s,1)

[msg-encerramento]
    exten => 1,1,PlayBack(msg-encerramento,s,1)

```

Abaixo são apresentados os comandos utilizados no desenvolvimento desta ura.

PlayBack: Utilizada para tocar um arquivo de som, este comando ignora qualquer tipo de comando enviado pelo usuário afim de interagir com o sistema durante a execução do referido arquivo.

Sintaxe -

```
exten => [chave],[prioridade],Playback(arquivo_audio)
```

Exemplo -

```
exten => s, 1, PlayBack (msg-abertura)
```

Background: Esta aplicação executa um arquivo de som, porém ao receber um comando digitado pelo usuário, a execução do áudio é interrompida, e o fluxo é encaminhado para a extensão que o usuário escolheu;

Sintaxe -

```
exten => [chave],[prioridade],Background(arquivo_audio)
```

Exemplo -

```
exten => s, 1, Background (msg-abertura)
```

Goto: Esta aplicação é utilizada para efetuar a troca (saltos) de contextos, extensão ou prioridade no Asterisk.

Sintaxe -

```
exten => [chave],[prioridade], TO([[contexto_destino]][extensão]][prioridade)
```

Exemplo -

```
exten => 1,1,Goto(serv-metereologia,s,1)
```

5.1.2 URA com suporte a banco de dados no Asterisk.

Uma URA com suporte a banco de dados MYSQL (MYSQL,2007) acessando este diretamente via comandos predefinidos no arquivo extension.conf, deve possuir além da instalação padrão do Asterisk, a instalação do pacote asterisk-addons. O banco de dados pode ser acessado remotamente.

O exemplo que segue, demonstra como desenvolver uma Unidade de Resposta Audível que disponibilize um serviço de tele-voto. Neste exemplo o usuário acessa o sistema, que solicita a sua senha (cadastrada no banco), sendo

que o acesso somente é liberado com a combinação correta da identificação do usuário e sua respectiva senha. Há proteção para evitar que o usuário vote mais de uma vez, bem como limita 2 tentativas de senha por ligação.

Implementação (via arquivo extension.conf)

A figura 7.2 Fluxograma URA – Tele Voto, apresenta o fluxograma desta implementação.

```

exten => 999,1,Goto(pesquisa,s,1)

[pesquisa]
exten => s,1,PlayBack(msg-abertura)
exten => s,2,MYSQL(Connect connid localhost root root asterisk)
exten => s,3,Read(identificacao|solicita-identificacao|4)
exten => s,4,MYSQL(Query resultid ${connid} SELECT\ nome\ FROM\ eleitor\ WHERE\ id=${identificacao})
exten => s,5,MYSQL(Fetch fetchid ${resultid} resultado1)
exten => s,6,GotoIf(${resultado1} = NULL)?8:10;
exten => s,7,MYSQL(Clear ${resultid})
exten => s,8,MYSQL(Disconnect ${connid})
exten => s,9,Goto(verifica-voto,s,1)
exten => s,10,Goto(solicita-identificacao-correta,s,1)

[solicita-identificacao-correta]
exten => s,1,MYSQL(Connect connid localhost root root asterisk)
exten => s,2,Read(identificacao|solicita-identificacao|4)
exten => s,3,MYSQL(Query resultid ${connid} SELECT\ nome\ FROM\ eleitor\ WHERE\ id=${identificacao})
exten => s,4,MYSQL(Fetch fetchid ${resultid} resultado1)
exten => s,5,GotoIf(${resultado1} = NULL)?8:10)
exten => s,6,MYSQL(Clear ${resultid})
exten => s,7,MYSQL(Disconnect ${connid})
exten => s,8,Goto(verifica-voto,s,1)
exten => s,9,PlayBack(identificacao-incorreta-repetida) *****
exten => s,10,Goto(msg-encerramento)

[verifica-voto]
exten => s,1,MYSQL(Connect connid localhost root root asterisk)
exten => s,2,MYSQL(Query resultid ${connid} SELECT\ id_candidato\ FROM\ pesquisa\ WHERE\ id_eleitor=${identificacao})
exten => s,3,MYSQL(Fetch fetchid ${resultid} r_identificacao)
exten => s,4,MYSQL(Disconnect ${connid})
exten => s,5,GotoIf("${r_identificacao}" = "0")?6:7)
exten => s,6,Goto(solicita-voto,s,1)
exten => s,7,Goto(voto-repetido,s,1) ;

[solicita-voto]
exten => s,1,Read(voto|solicita-voto|1)
exten => s,2,Goto(computar-voto)s,1)

[computar-voto]
exten => s,1,MYSQL(Connect connid localhost root root asterisk)
exten => s,2,MYSQL(Query resultid ${connid} UPDATE\ pesquisa\ SET\ id_candidato=${voto}\ WHERE\ id_eleitor=${identificacao})
exten => s,3,MYSQL(Fetch fetchid ${resultid} resultado1)
exten => s,4,MYSQL(Clear ${resultid})
exten => s,5,MYSQL(Disconnect ${connid})
exten => s,6,PlayBack(voto-computado)
exten => s,7,Goto(msg-encerramento)

[voto-repetido]
exten => s,1,PlayBack(voto-repetido)
exten => s,2,Goto(msg-encerramento)

[msg-encerramento]
exten => s,1,PlayBack(msg-encerramento)

```

Abaixo são apresentados os comandos utilizados no desenvolvimento desta URA.

Read: Um dos comandos mais importantes para a criação de URA no Asterisk , pois devido aos parâmetros que o mesmo pode suportar, este é capaz de executar um arquivo de áudio, limitar a quantidade de dígitos que o usuário pode digitar, o tempo máximo de espera para receber a interação com o usuário, além de salvar em uma variável o conteúdo digitado pelo usuário.

Sintaxe -

```
Exten => [chave],[prioridade],[Read(var_digitada[arquivo_audio][max_dig][opções][tentativas][timeout])
```

Exemplo -

```
exten => s,1,Read(voto/solicita-voto|1)
```

GotoIF: Esta aplicação é utilizada para efetuar a troca (saltos) de contextos, extensão ou prioridade no Asterisk, porém diferente da aplicação Goto, este comando testa uma condição antes de efetuar o deslocamento do fluxo de execução da URA.

Sintaxe -

```
exten => [chave],[prioridade],[GotoIF([condicao],label1,label2)]
```

Obs: Sendo label a extensão e/ou contexto para qual o fluxo será desviado (sendo o label1 caso a condição proposta seja verdadeira);

Exemplo -

```
exten => s,5,GotoIf("${r_identificacao}" = "0")?6:7)
```

MYSQL: Esta aplicação adiciona ao Asterisk as funcionalidades básicas de um banco de dados MYSQL. Sua sintaxe é um pouco mais complexa que as apresentadas anteriormente, pois a necessidade de conexão e desconexão com o banco de dados para liberar as instâncias utilizadas durante a consulta , além de limpar as variáveis envolvidas.

Conexão com o banco

Sintaxe - `Exten => [chave],[prioridade],[MYSQL(Connect connid IP_BD usuário_BD senha nome_da_base)]`

Exemplo - `exten => s,1,MYSQL(Connect connid localhost root root asterisk)`

Desconexão

Sintaxe - `Exten => [chave],[prioridade],[MYSQL(Disconnect ${connid})]`

Exemplo - `exten => s,5,MYSQL(Disconnect ${connid})`

Construindo uma query

Sintaxe - `Exten => [chave],[prioridade],[MYSQL(Query var_resultado ${connid} query-string)]`

Exemplo –

`exten => s,1,MYSQL(Query resultid ${connid} SELECT id_candidato FROM pesquisa WHERE id_eleitor=${identificacao})`

Salvando o resultado de uma consulta em uma variável.

Sintaxe -

```
Exten => [chave],[prioridade],[ MYSQL(Fetch fetchid ${resultid} var1)]
```

Exemplo -

```
exten => s,4,MYSQL(Fetch fetchid ${resultid} resultado1)
```

Obs: a variável resultado1 receberá o conteúdo de resultid

Outros comandos:

Record: Esta aplicação é utilizada para gravar as chamadas ou pode ser aproveitada para gravar as mensagens que serão utilizadas pela próprio sistema Asterisk.

Sintaxe -

```
exten => [chave],[prioridade],Background(arquivo_audio)
```

Exemplo -

```
exten => s, 1, Background (msg-abertura)
```

Hangup: Ao invocar esta aplicação, o sistema Asterisk desconecta o canal em uso. Normalmente utilizada para evitar que chamadas fiquem “presas” ou em estado indefinido.

Sintaxe -

```
exten => [chave],[prioridade],Background(arquivo_audio)
```


Exemplo -

*exten => s, 1, **Background** (msg-abertura)*

6 Ferramentas utilizadas para criar URA no Asterisk

Não foram localizadas ferramentas específicas para desenvolvimento de URA. O que existem são ferramentas de configuração de Plano de Discagem (Dialplan) e ferramentas de configuração completa do servidor Asterisk.

Neste capítulo será apresentada uma breve análise de ferramentas disponíveis a usuários finais que desejam configurar uma URA no Asterisk.

6.1 Visual Dialplan Beta

Este aplicativo, contempla todo o escopo de configuração do arquivo extension.conf do sistema Asterisk, portanto seu propósito é controlar via arquivo de configuração, todos os canais configurações no sistema e suas respectivas ligações(entrantes ou saintes).

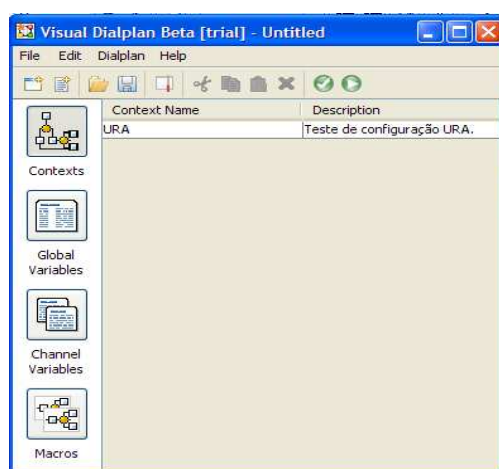


Figura 6.1 - Visual Dialplan / Menu Principal

Como a uma grande gama de configurações que podem ser efetuadas no Asterisk, e o Visual Dialplan possibilita a configuração visual (em partes) de todas estas possibilidades, a interface acaba por ficar demasiadamente carregada, fazendo com que um usuário leigo não saiba por onde iniciar a configuração.

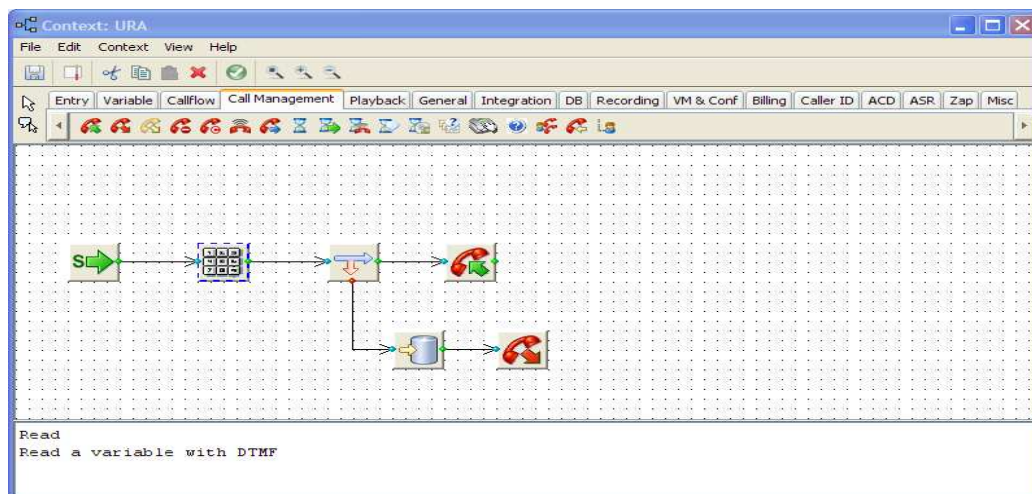


Figura 6.2 - Visual Dialplan / Menu para edição do arquivo extension.conf

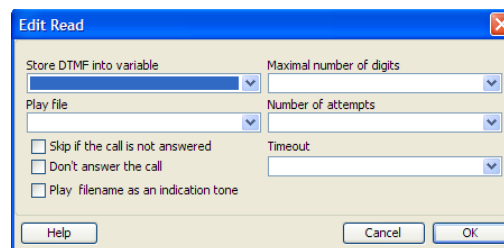


Figura 6.3 - Visual Dialplan / Menu de configuração da Aplicação Read

Todavia, o sistema disponibiliza um menu de ajuda muito vasto, que além de informações sobre as aplicações, ainda demonstra com exemplos (figuras) como é realizada cada configuração.

Ainda como diferencial, o Visual Dialplan permite que o usuário selecione a release do Asterisk para qual o arquivo de configuração esta sendo gerado, uma vez que a versão 1.4 do Asterisk criou aplicações, e estas não podem ser executadas em sistemas com release anteriores, sob pena de inviabilizar o funcionamento da nova configuração.

Devido ao propósito geral do mesmo, a confecção de uma URA é uma atividade um tanto quanto árdua, pois as configurações (módulos) necessários

estão espalhados pelas diversas abas do sistema de configuração, sendo que alguns dos ícones não são instintivos.

Porém um dos principais pontos negativos deste aplicativo é a dependência de sistema operacional, neste caso, a exclusividade da plataforma Windows. Aparentemente é uma incoerência, uma vez que o Asterisk é um dos principais sistemas Software Livre, sendo sua plataforma padrão o Unix.

Distribuição: Este software é proprietário, porém há uma distribuição beta, disponibilizada pela internet para testes.

Versão: 1.0.00-b

Desenvolvido: Apstel.

Site: <http://www.apstel.com>

6.2 Dialplanner

Com um foco em usuários mais experientes, que buscam apenas automatizar os códigos referentes a entradas, variáveis e aplicações, sem a necessidade de visualizarem a seqüência de execução da URA de forma gráfica, esta ferramenta acaba por restringir o número de usuários, uma vez que a necessidade de conhecer o funcionamento em nível de arquivos de configuração para poder efetuar a seqüência correta (não apenas válida, mas que tenha um significado ao ser executado).

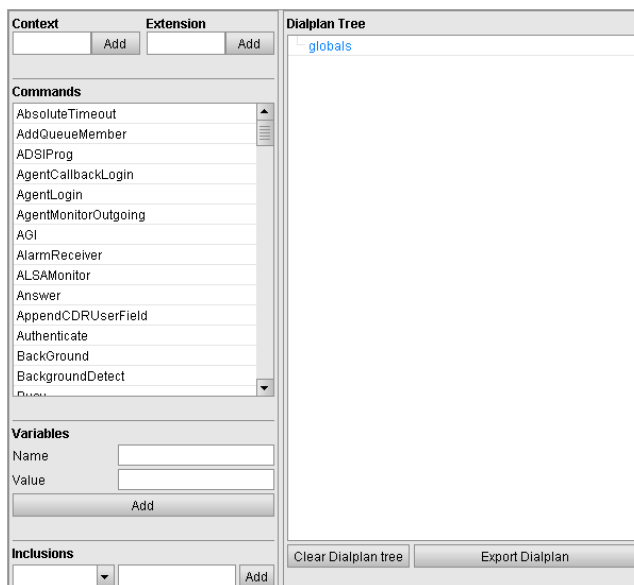


Figura 6.4 - Dialplanner / Menu Principal

Esta ferramenta possui uma grande vantagem, que é independência de plataforma, pois trata-se de um *applet* Java [JAVASE, 2007] que é executado diretamente no browser, sendo necessário apenas a instalação do Java.

A falta de um sistema de ajuda on-line também é um fato negativo. Outro ponto negativo é a pluralidade de opções de comando (todas as aplicações do Asterisk) unidas em apenas uma lista (mais de 160 opções), o que acaba causando uma grande perda de tempo para o usuário localizar a opção desejada.

A falta de validação do código gerado também é um ponto que necessita de ajustes por falta do desenvolvedor, evitando que o código seja migrado para o Asterisk e seja validada via console pelo usuário.

Distribuição: Gratuita.

Versão: não informada.

Desenvolvido: Lanvik ICU Sdn Bhd

Site: <http://www.lanvik-icu.com/asterisk/dialplanner/index.php>

6.3 ACTOS - Asterisk Configuration Tool Open Source

Esta ferramenta é utilizada para gerenciar os parâmetros de configuração mais importantes na gerência de um sistema Asterisk. Um destaque é a função que busca os arquivos de configuração, tanto na plataforma onde a ferramenta esta instalada, quanto em um servidor remoto.

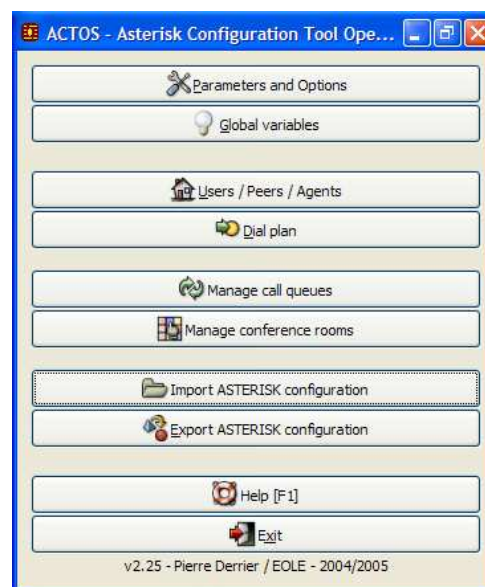


Figura 6.5 - ACTOS / Menu Principal

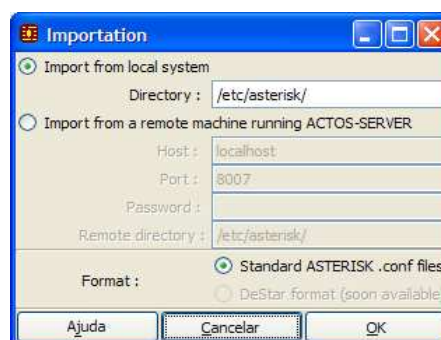


Figura 6.6 - ACTOS / Menu de localização dos arquivos de configuração

A configuração é feita de forma que apenas usuários familiarizados com os arquivos e aplicações do Asterisk possam realizar as operações de maneira correta, uma vez que não há indicação de símbolos e os arquivos de ajuda não foram implementados na versão atual.

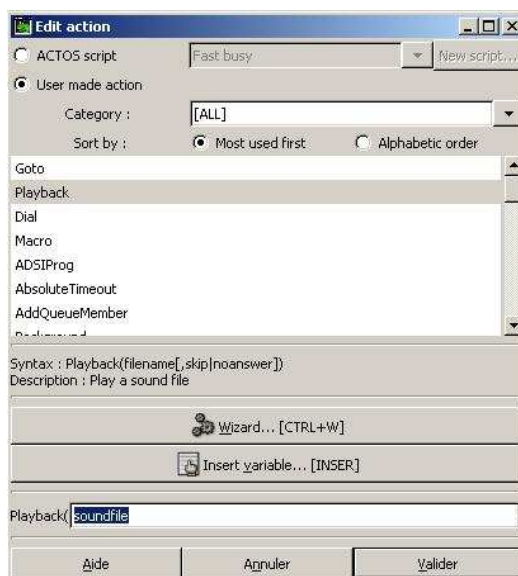


Figura 6.7 - ACTOS / Menu para edição do arquivo extension.conf

Como a aplicação é de propósito geral (todas as aplicações estão disponíveis nos menus), um usuário leigo teria dificuldade para montar uma URA com facilidade, além deste fato, não existe o recurso visual (similar a um fluxograma), para que o usuário acompanhe a evolução da implementação de sua aplicação.

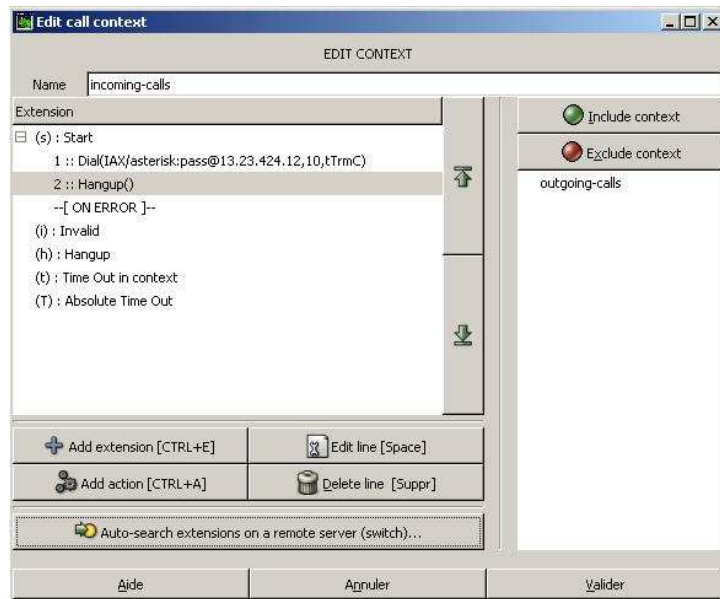


Figura 6.8 ACTOS / Menu do arquivo extension.conf (seqüência de execução)

Há versões deste software para Linux e Windows, porém a necessidade de escolher o sistema operacional onde o sistema será instalado, ou seja o conceito de multiplataforma não é totalmente respeitado.

Distribuição: Livre. Código Aberto (*Open Source*)

Versão: 2.25

Desenvolvido: Lanvik ICU Sdn Bhd

Site: <http://www.derrier.com/pierre/code/actos.html>

7 Ferramenta para desenvolvimento de URA no Asterisk

Com o intuito de propiciar ao usuário uma ferramenta que possibilite configurar uma URA no Asterisk, sem a necessidade do conhecimento em nível de arquivo de configuração (`extension.conf`), fazendo com que a arquitetura a ser criada (visualmente) seja uma seqüência de instruções com fluxos pré-determinados.

Para tanto, ao contrário das ferramentas de propósito geral (configuração completada do Asterisk), o modulo que será desenvolvido disponibilizará apenas as aplicações mais utilizadas para a construção de uma URA, facilitando a configuração para os usuários iniciantes.

7.1 Metodologia

Visando construir uma ferramenta que tornasse ágil o desenvolvimento de URA no Asterisk, subdividimos a criação desta em quatro etapas:

- A) Escolha de um modelo Gráfico para especificação da URA;
- B) Criação de um modelo de uma URA em XML (Extensible Markup Language)(URA-XML), representando textualmente o modelo gráfico definido;
- C) Desenvolvimento de um mecanismo de transformação do o arquivo URA-XML no arquivo `extension.conf` do Asterisk;
- D) Desenvolvimento de uma ferramenta gráfica para desenvolvimento de URA no Asterisk seguindo o modelo gráfico definido. Além disso, esta ferramenta incluirá o processo de geração dos arquivos URA-XML e posterior geração do arquivo `extension.conf`.

7.2 Modelo Gráfico para Especificação de URAs

A configuração de uma URA no Asterisk é definida basicamente no arquivo `extension.conf`, este possibilita a configuração de estruturas condicionais, saltos e repetições, onde podemos inferir que sua estrutura é análoga a de um algoritmo.

Segundo PETRY(2005), existem três formas mais adotadas de representar um algoritmo:

- Descrição narrativa: algoritmos são representados em linguagem natural;
- Pseudocódigo: lista as ações a serem tomadas pelo computador, sem se preocupar com as regras rígidas da sintaxe da linguagem;
- Fluxogramas: um fluxograma é uma representação gráfica de um fluxo de controle, onde setas indicam a seqüência das tarefas (representadas por retângulos) e losangos representam os pontos de tomada de decisão;

Visando facilitar e agilizar o desenvolvimento de URA, optou-se por construir uma ferramenta que utilizará como modelo gráfico fluxogramas.

Graficamente, um fluxograma é representado pelos seguintes símbolos:



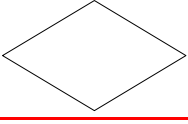
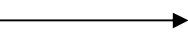
	Representa o início/fim de um fluxo.
	Indica a manipulação de dados.
	Ponto de tomada decisão.
	Indica o fluxo de controle. Interconecta os elementos do fluxograma.

Tabela 7-1 Principais símbolos utilizados em um fluxograma

Segue exemplos de fluxogramas de URAs (implementação no Asterisk abordada no capítulo 5):

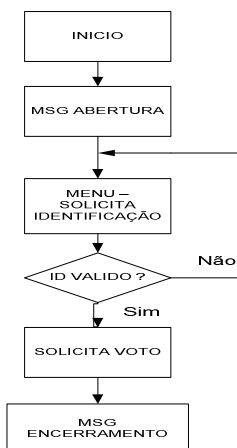


Figura 7.1 Fluxograma URA - Previsão Meteorológica

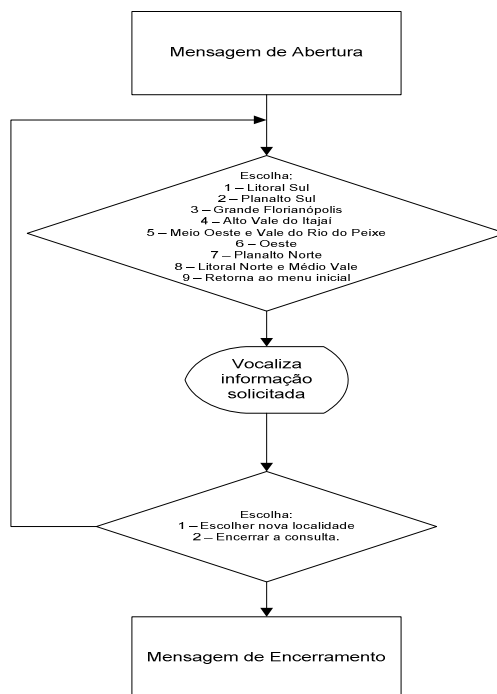


Figura 7.2 Fluxograma URA - Tele Voto

7.3 Modelo de URA em XML

Para representar internamente o fluxograma criado pelo usuário, optou-se por descrever a estrutura em XML sendo o tradução schema XML para o arquivo extension.conf efetuada automaticamente pela ferramenta.

A opção por XML é justificada por esta ser uma linguagem semântica, que possibilita a gerência, troca e apresentações dos dados.

A XML é uma linguagem de marcação que utiliza tags para delimitação das informações. Ela foi desenvolvida pelo World Wide Web Consortium - W3C (W3C, 2007).

Como principais características do XML, podemos destacar:

- Portabilidade, flexibilidade, verificação de erros, extensibilidade, representação estruturada dos dados e sem ambigüidades.

Alguns aspectos da linguagem a tornam mais eficientes, com a Interoperabilidade que permite a independência de plataforma ou de sistema operacional. Outro aspecto é a extensibilidade, que permite criar novas (próprias) regras ou elementos. Além destes, a propriedade de dados autodescritivos, facilita a reutilização em futuras aplicações.

Uma das bases da XML é o elemento. O nome do elemento deve facilitar o entendimento de seu conteúdo. A representação é feita por meio de marcações (tags), estas indicam o início e o fim dos elementos.

A sintaxe de um elemento pode ser representada da seguinte forma:

```
<nomeElemento>conteúdoElemento</nomeElemento>
```

Outro pilar da XML é o atributo, ao utilizado, deve-se declarar junto com a tag inicial do elemento, conforme o exemplo:

```
<nomeElemento nomeAtributo="texto ou valor" >
```

Atributos normalmente são utilizados para informações básicas, que adicionam dados ao conteúdo de um elemento.

Baseado nos fluxogramas apresentados no capítulo 7 (figuras 7.1 e 7.2), segue exemplos da descrição destes em XML.

A descrição XML abaixo é referente a uma URA que disponibiliza a previsão meteorológica:

```
<ura-xml runat="endereço do asterisk">
<head>
  <metadata name="title" content="URA para votação" />
  <metadata name="author" content="Guilherme Vieira e Roberto Willrich" />
  <db-name="base-ura" host="ip-db" user="root" passwd="****" ... />
  <start accessNumber="122" />
</head>
<body>
  <audio id="abertura" src="file://..." />
  <input id="solicitaIdent" name="solicita identificador" msg-src="file://..." maxDigits="4" />
  <db-access db="base-ura" id="verificaVoto" var="user_no_validate" query="commando sql" />
  <if test="user_no_validate\ = true">
    <then>
      <goto dst="solicitaIdent" />
    </then>
  </if>
  <input id="solicitaVoto" var="voto" src="file://..." maxDigits="4" />
  <db-access dB="base-ura" var="confirmaVoto" query="inserir o voto digitado na base-ura" />
  <if test="confirmaVoto\ = true">
    <then>
      <audio name="agradece voto com sucesso" src="file://..." />
    </then>
    <else>
      <audio name="voto invalido"> src="file://..." />
    </else>
  </if>
</body>
</ura-xml>
```

O exemplo que segue, é um fluxograma simplificado de uma para serviço de votação via telefone (vide fluxograma – figura 7.2).

```

<ura-xml runat="endereço do asterisk">
<head>
  <metadata name="title" content="URA metereologia" />
  <metadata name="author" content="Guilherme Vieira e Roberto Willrich" />
  <start accessNumber="123" />
</head>
<body>
  <audio id="abertura" src="file://..." />
  <input id="solicitaReg" name="solicita região" msg-src="file://..." maxDigits="1" />
    <then>
      <audio name="previsao regioao X" src="file://..." />
    </then>

  <input id="menu2" name="divulga menu" src="file://..." maxDigits="1" />
    <if test="menu" = true">
      <then>
        <audio name="mensagem de enerramento" src="file://..." />
      </then>
      <else>
        <goto dst="solicitaReg" />
      </else>
    </if>
</body>
</ura-xml>

```

7.4 Traduzindo um fluxograma de URA para XML

Após a construção da URA pelo usuário (construção do fluxograma), cabe a aplicação a ser desenvolvida neste trabalho: representar a imagem criada pelo usuário em um arquivo XML.

Esta tradução é possível devido ao mapeamento das aplicações disponibilizadas ao usuário (via figuras de um fluxograma) em tags XML.

Baseado no exemplo do item anterior, uma mensagem de abertura no fluxograma recebe a seguinte representação:

MSG ABERTURA

Sendo mapeada da seguinte forma para o arquivo XML:

```
<audio id= "abertura" src= "file://..." />
```

7.5 Convertendo um arquivo XML para o formato padrão do Asterisk

De maneira análoga com o procedimento executado no item 7.3, foi criado um mapeamento das tags XML para o comando correspondente no arquivo extension.conf do Asterisk.

Desta forma, a tag XML que representa uma mensagem de abertura (apresentada no item 7.3), ficaria da seguinte forma no arquivo extension.conf:

```
exten => [chave],[prioridade],Playback(arquivo_audio)
```

Segue abaixo a especificação completa da tradução do fluxograma gerado pelo usuário, onde a ferramenta ira transformar este em um arquivo XML (funcionando neste caso como uma linguagem intermediária).

Após esta etapa, a ferramenta converterá o arquivo XML para o padrão do Asterisk, propiciando a configuração de uma URA para o usuário.






Símbolo	Parâmetros	XML	Asterisk
	Título Autor Num. de acesso End. Asterisk	<ura-xml runat="endereço do asterisk"> <head> <metadata name="title" content="URA para votação" /> <metadata name="author" content="Guilherme Vieira e Roberto Willrich" /> <start accessNumber="122" />	; URA para votação ;Autor : Guilherme Vieira e Roberto Willrich exten => 122,1,Goto(inicio,s,1)
	URL da mensagem Max de Dígitos Var_Dígitos	<audio id="abertura" src="file://..." /> <input id="solicitaIdent" name="solicita identificador" msg-src="file://..." maxDigits="4" />	exten => s,3,Read(identificacao solicita- identificacao 4)
	End. Do BD Nome da Base Usuário do BD Senha Query	<db-name="base-ura" host="ip-db" user="root" passwd="*****" ... /> <db-access db="base-ura" id="verificaVoto" var="user_no_validate" query="commando sql" />	exten => s,2,MYSQL(Connect connid localhost root root asterisk) exten => s,3,MYSQL(Query resultid \${connid} SELECT nome FROM eleitor WHERE id=\${identificacao})
	Título URL da mensagem	<metadata name="title" content="Mensagem de Abertura" /> <audio id="abertura" src="file://..." />	; Mensagem de Abertura exten => s,1,Playback(arquivo_audio)
	Variável para teste Mensagem (condição verdadeira) Mensagem (condição falsa)	<if test="confirmaVoto = true"> <then> <audio name=" voto com sucesso" src="file://..." /> </then> <else> <audio name="voto invalido"> src="file://..." /> </else> </if>	exten => s,5,GotoIf("\${variavel}" = "0")?6:7) Obs: 6 -> linha executada caso o teste seja verdadeiro 7 -> linha executada caso o teste seja falso

Tabela 7-2 Especificação das traduções (Fluxograma - XML - Asterisk)

Conhecendo os elementos apresentados, segue um exemplo de uma URA utilizando a estrutura abordada neste trabalho:

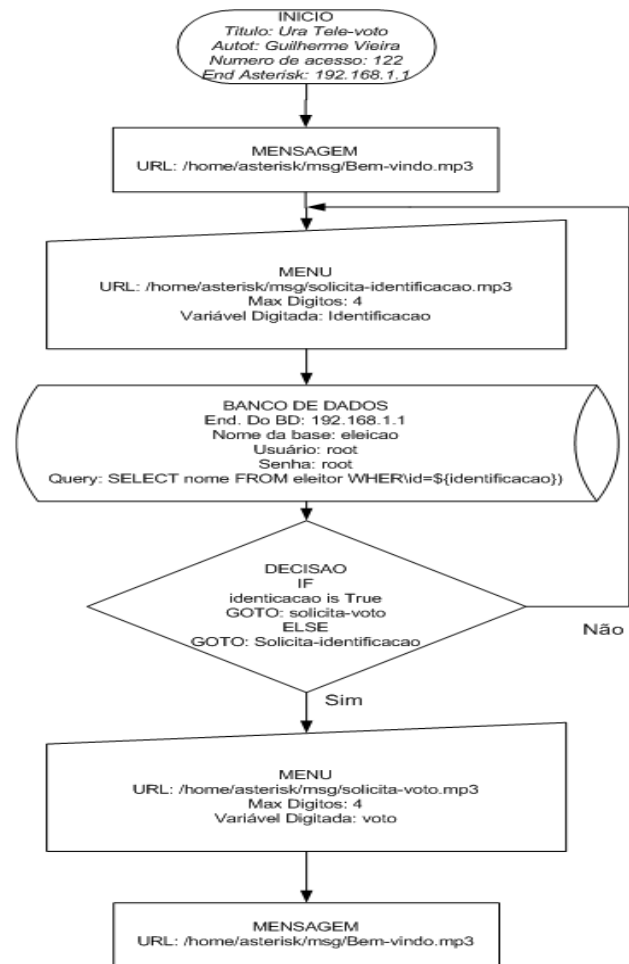


Figura 7.3 URA Tele Voto (utilizando o modelo proposto)

7.6 Desenvolvimento da Ferramenta

A ferramenta proposta, ira possibilitar ao usuário montar um fluxograma da URA, exigindo pouco conhecimento sobre o funcionamento da mesma no Asterisk.

Esta aplicação ira salvar toda a estrutura da URA em um arquivo XML, que descrevera toas as particularidades utilizadas no processo de construção da aplicação proposta pelo usuário.

8 Conclusão e Trabalhos Futuros

8.1 Conclusão

A evolução do sistema *open source* PBX Asterisk, tem quebrado vários paradigmas no mercado de Telecomunicações, um mercado até pouco tempo oligárquico, acompanha o crescimento e a rápida adoção deste sistema que é apontado por vários como o agente principal de uma nova revolução nas telecomunicações.

Várias aplicações são desenvolvidas por colaboradores (membros da comunidade *open source*) do sistema, e disponibilizadas para os demais usuários, com o intuito de facilitar e popularizar o uso do Asterisk. Este fato é um dos principais motivos para o sucesso e crescimento vertiginoso no número de novos servidores Asterisk instalados atualmente.

Entretanto, não foram localizadas ferramentas com o foco na construção de URA no Asterisk, neste momento encontramos o ponto de partida deste trabalho. O modelo de ferramenta proposto, baseia-se na criação de fluxogramas, para melhor visualizar os vários fluxos possíveis da URA, que posteriormente será codificado para o padrão do arquivo `extension.conf` do Asterisk.

Pode-se destacar também como vantagem deste modelo não requerer do usuário qualquer conhecimento prévio das aplicações do Asterisk, exige apenas lógica do mesmo, pois o intuito da ferramenta é propiciar ao usuário a criação de um fluxograma de sua URA.

8.2 Trabalhos Futuros

- Criar módulo básico e avançado para a ferramenta, para que a ferramenta seja utilizada por usuários iniciantes e experientes.

- Validar em tempo real a aplicação para o usuário, fazendo com o que o mesmo não necessite importar o arquivo gerado pelo framework para o Asterisk para efetuar testes.

- Desenvolver uma interface gráfica que permita o usuário desenvolver URA utilizando diagramas SDL e UML.

- Possibilitar a integração do framework com scripts AGI, possibilitando a construção de URA's mais complexas, controladas por aplicações externas ao Asterisk.

ANEXOS

ANEXO 1 – Ferramenta Gráfica de Criação de Unidades de Resposta Audível para o Servidor VoIP Asterisk

Ferramenta Gráfica de Criação de Unidades de Resposta Audível para o Servidor VoIP Asterisk

Guilherme Vieira¹

Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (USFC)
Florianópolis, SC – Brasil

guiv@inf.ufsc.br

Abstract. *The concept of convergence, as an integration of telephony (commutated) with data network, is providing a revolution in both sectors. Within this revolution, the most highlight application is VoIP, which allies the possibility of communication with a significative decreasing of costs on long distance calls.*

One of the agents which most identifies with the concept above is the Asterisk, a PABX, open source, with large acceptance in the market, which makes possible a great range of applications without demanding huge amounts of money, that would be necessary if the PABX was a private one.

In this context, the scope of this work includes the creation of a tool for the quick development of an IRV in the Asterisk system, allowing "rooky" users to build their own IRV's (One of the most expensive private solutions), graphical, requiring no large know how on Asterisk applications.

Resumo. *O conceito de convergência, como integração de telefonia (comutada) com rede de dados, esta propiciando uma revolução em ambos os setores. Dentro desta revolução, a aplicação que possui maior destaque é o VoIP, que além da possibilidade de comunicação, alia a diminuição significativa dos custos com ligações de longa distância.*

Um dos agentes que possui maior identidade com o conceito acima exposto é o Asterisk, um sistema de Ramais Privados (PABX), open source, de grande aceitação no mercado, que possibilita uma vasta gama de aplicações, sem a necessidade do usuário pagar as enormes quantias que seriam necessárias caso a solução PABX do mesmo fosse proprietária.

Neste contexto, este trabalho abrange a criação de uma ferramenta para desenvolvimento ágil de Unidade de Resposta Audível (URA) no sistema Asterisk, possibilitando que usuários leigos possam construir suas URA's (umas das aplicações de maior custo em soluções proprietárias), de modo gráfico, sem a necessidade de conhecer a fundo as aplicações do Asterisk.

9 Introdução

As redes de dados e telefonia sempre trilharam caminhos distintos, usando suas próprias infra-estruturas. Isto impacta no aumento dos custos, devido à necessidade de instalação de dois serviços, além de haver necessidade de monitorar a funcionalidade de ambos (COSTA, 2003). Este cenário está em processo de mudança desde o surgimento do conceito de convergência, permitindo integrar a telefonia e a rede de dados em grandes empresas do ramo das comunicações, gerando volumosos investimentos em pesquisas.

Dentre as aplicações que permitem a comunicação e pertencem ao conceito de convergência, o VoIP (Voice over Internet Protocol) surge como destaque, pois o mesmo propicia além da simples comunicação, grande interatividade, além da diminuição significativa nos custos com ligações interurbanas (FERNANDES, 2000).

O telefone (fixo ou celular), ainda representa um número maior de usuários, frente ao número de computadores no Brasil. Devido a este fato, várias aplicações são ainda desenvolvidas utilizando URA (Unidade de Resposta Audível) visando interagir com usuário, repassando informações ora cadastradas em um banco de dados, como por exemplo, as consultas de saldo em bancos ou a declaração de imposto de renda via serviço ReceitaFone (RECEITAFONE, 2007).

Baseado no contexto de convergência, em 1999 surgiu um Sistema de Ramais Privados (PABX) de fonte aberta, denominado Asterisk (Asterisk, 2006), que segundo seu criador Mark Spencer, é capaz de fazer qualquer coisa com uma ligação ao colocá-la dentro do computador, portanto capaz de criar soluções altamente personalizáveis, sem a necessidade de arcar com os altos custos provenientes desta, uma vez que é de conhecimento que a indústria das telecomunicações agrega valores considerados altos para suas soluções, o que inviabiliza os usuários com recursos financeiros reduzidos de possuir um PABX de pequeno porte adequado as suas necessidades.

Unidade de Resposta Audível (URA) é utilizada para o atendimento eletrônico ou encaminhamento de ligações. De acordo com a sua estrutura, é capaz de disponibilizar menus de navegação, tais como músicas ou mensagens pré-gravadas. A possibilidade de explorar a tecnologia VoIP com URA, possibilita a criação de várias aplicações, tais como: recepcionista digital, tele avisos (com mensagens pré-programadas), etc.

10 VoIP

Um dos ramos de maior evolução atual é os das telecomunicações. Dentro deste, destaca-se a Telefonia, esta iniciou em 1876, com a invenção do telefone por Alexander Graham Bell. Nesta época, para que um telefone pudesse comunicar-se com outro, era necessário um par de fios dedicados, o que onerava consideravelmente a instalação e o crescimento deste tipo de comunicação.

Para solucionar este entrave, foram inseridas no contexto da comunicação as centrais de comutação. Elas operavam conectadas a todos os usuários via par metálico (conceito ainda utilizado atualmente), interconectando fisicamente os

pares envolvidos em uma chamada. Com isto, o custo com cabos caiu drasticamente, porém a comutação ainda era lenta e totalmente mecânica.

Seguindo a cronologia dos fatos, a digitalização da telefonia foi o novo passo desta cadeia. Conforme Pinheiro (2006), a passagem para a concepção digital iniciou nos sistemas de transmissão, utilizando a técnica de digitalização PCM (Pulse Code Modulation), paulatinamente foi inserida nas centrais telefônicas, por meio de equipamentos com tecnologia CPA (Controle por Programa Armazenado). Nesta etapa, com os sinais já digitalizados, as portas para o mundo da convergência estavam abertas, bastando neste momento a vontade política das empresas que dominavam o mercado das telecomunicações aderir a esta tecnologia.

É nesta fase que o VoIP inicia a atual revolução. Com a evolução das redes (dados e voz), uma evolução natural seria a convergência. Esta, devido a sua expansão, consolida-se como o próximo passo na evolução das telecomunicações. Um dos expoentes na era da convergência são as aplicações de Voz que utilizam a rede IP, denominadas aplicações VoIP.

Um dos pontos fundamentais para a adoção das tecnologias VoIP é a possibilidade de redução significativa nos gastos com ligações (via PSTN) e a facilidade na comunicação entre os usuários do sistema (COSTA, 2003).

O uso da tecnologia VoIP implica na utilização de alguns protocolos, necessários para estabelecer o tráfego de Voz na rede IP.

10.1 Protocolos de Sinalização

Necessários para controlar os fluxos persistentes de mídia (bidirecionais) que transportam a informação, são vários os protocolos que viabilizam a comunicação VoIP.

SIP - O protocolo SIP (Session Initiation Protocol), inicialmente proposto pela RFC (Request for Comment) 2543 de 1999, sofreu várias atualizações, que levaram a uma nova publicação chamada RFC 3261 de 2002. Este protocolo atua na camada de aplicação, sendo sua função iniciar, estabelecer, modificar e encerrar as sessões da rede IP.

H.323 - O padrão H.323 é integrante da família de recomendações do ITU-T (International Telecommunications Union) denominadas H.32x, cuja função principal é prover serviços de multimídias sobre diversos tipos de redes. Este padrão descreve detalhadamente como a comunicação multimídia transcorre entre terminais do usuário, equipamentos de rede e vários serviços nas redes IP. É tratado como uma especificação de sistema, pois sua documentação faz menção a outras recomendações (CORIOLANO, 2003).

10.2 CODEC de Voz

Inicialmente, o termo CODEC fazia referência ao par COdificador/DECodificador, sendo a função do mesmo efetuar conversões entre

sinais analógicos e digitais. Entretanto, no contexto atual está mais relacionado com COrpressão/DEsCOrpressão (SMITH, MEGGELEN e MADSEN, 2005).

11 Unidade de Resposta Audível

Segundo Barreiros (2004), uma Unidade de Resposta Audível é utilizada para o atendimento eletrônico das ligações. É capaz de disponibilizar menus de navegação, tais como músicas ou mensagens pré-gravadas.

Um ponto que merece destaque, é que ao contrário de países desenvolvidos, o número de computadores frente aos de telefones é relativamente pequeno no Brasil, portanto o acesso a sistemas via telefone ainda é um mercado em expansão (GONÇALVES, 2005).

11.1 Aplicações

Para usuários que não possuem uma faixa de número externo da PSTN, DDR – discagem direta ao ramal, a URA pode viabilizar o acesso indireto ao destino, sem que a ligação passe pela entidade telefonista (humana

Aplicações complexas, com vários níveis de questionamentos, são normalmente adotadas por callcenter de grande porte, onde a moeda corrente é o tempo de ocupação do usuário, sendo que neste caso, é possível diminuir os tempos de atendimento, pois as chamadas já foram triadas, possibilitando um maior tempo livre dos atendentes (agentes) e um aumento significativo de capacidade de atendimento do callcenter.

Em situações onde a presença humana é necessária apenas para digitar números de acesso, ou rotear as chamadas para uma aplicação que vocaliza a solicitação, a URA pode ser utilizada para solicitar as cifras de acesso e via integração com a aplicação de vocalização (normalmente um banco de dados com mensagens pré-configuradas), repassar a informação solicitada pelo cliente. Neste contexto está enquadrado as URA's de bancos que disponibilizam o serviço de consulta de saldo, ou lojas que consultam se o CPF (Cadastro de Pessoa Física) do cliente está incluso nos organismo de proteção ao crédito.

12 Asterisk

O Asterisk (Asterisk, 2006) é um PABX completo em software, baseado no paradigma Open Source, sendo atualmente multiplataforma. Criado em 1999 por Mark Spencer, é apontado como um dos principais agentes da atual revolução nas telecomunicações.

12.1 Arquitetura

Segundo Gonçalves (2005), a arquitetura do Asterisk pode ser subdividida, sendo três os principais conceitos incorporados a esta arquitetura: canais, codecs e aplicações. Como canal, é entendido toda a entrada (ou saída) de um sinal (digital ou analógico) de qualquer interface do Asterisk. O Codec, trata-se do

elemento que codifica e decodifica as informações das comunicações VoIP, responsável por adaptar a quantidade de bits que serão transmitidos para a rede, sempre com a meta de deixar a qualidade de voz o mais próximo possível do natural (ou pelo menos o padrão já alcançado pela telefonia tradicional). Já as aplicações, são as funcionalidades acopladas ao sistema, tais como: correio de voz, conferência, etc.

12.2 Funções

As principais funcionalidades incorporadas ao Asterisk, que fazem-no despontar como um equipamento de ponta, são: correio de voz, unidade de resposta audível (URA), distribuidor automático de chamadas (DAC), conferência de áudio, discador automático, servidor de música de espera, registro detalhado das ligações e interoperabilidade com diferentes padrões de VoIP.

12.3 Hardware

Para aplicações de puramente VoIP, o Asterisk não necessita de itens de hardware especiais. Todavia é importante ressaltar que o sistema utiliza os recursos da CPU do PC onde está instalado para processar os canais de voz, ao contrario de PABX tradicionais que utilizam DSP (Processador de Sinais Digitais) dedicados em seus equipamentos, para efetuar esta tarefa. Portanto para que aplicações possam retirar o máximo do Asterisk, a necessidade de instalar os servidores Asterisk em uma rede isolada para aplicações VoIP, evitando que o servidor pabx seja alvo de vírus ou qualquer outro tipo de ataque pela rede.

Para interconexão com a PSTN, a necessidade de instalar as interfaces no sistema, sejam elas digitais ou analógicas.

12.4 Protocolos suportados

O Asterisk suporta os principais, senão todos, protocolos utilizados atualmente nas comunicações VoIP. Além dos já citados H.323 e SIP, o Asterisk implementa o IAX (Inter-Asterisk). Este último protocolo, segundo Smith, Meggelen e Madsen (2005), foi concebido para possibilitar a comunicação entre servidores Asterisk, sendo este o motivo do seu acrônimo.

O IAX um protocolo de transporte similar ao SIP, sendo uma de suas principais características utilizar apenas uma porta para comunicação, tanto para o RTP (Real Time Protocol), como para a sinalização do canal.

13 Unidade de Resposta Audível no Asterisk

O Asterisk permite criar desde URA's simples, que apenas divulgam mensagem e encaminham para determinados ramais (recepcionistas digitais), bem como a criação de Unidades de Resposta Audíveis com alto grau de complexibilidade, envolvendo desde conexão com banco de dados e aplicações externas, até mesmo reconhecimento de voz.

O desenvolvimento de aplicações de URA no Asterisk exige conhecimento da arquitetura do sistema por parte do usuário, devido à necessidade de editar arquivos de configuração manualmente, pois o sistema não dispõe de uma aplicação que facilite esta atividade (objetivo deste trabalho).

Uma possibilidade de criar URA no Asterisk (manualmente) é editando o arquivo de configuração `extension.conf`. Este arquivo é responsável pelo plano de discagem da plataforma (controla todas as chamadas).

O arquivo é estruturado de forma a conter contextos, sendo este o responsável pelo controle das chamadas entrantes nos canais.

Cada extensão é nomeada da seguinte maneira:

[ExemploNomeExtensão]

Um contexto no Asterisk segue este modelo:

exten => [chave],[prioridade],[aplicação]

Onde:

Chave: Número digitado pelo originador da chamada. Esta é a chave que determina que a linha em questão será executada;

Prioridade: Prioridade na ordem da execução das aplicações. Não necessitam ser consecutivas, porém são executadas em ordem crescente;

Aplicação: Segundo Gonçalves (2005), as aplicações são fundamentais ao Asterisk, pois é esta que trata definitivamente a chamada, seja tocando música, gravando uma mensagem ou terminando uma chamada.

13.1 Exemplos de URA's

Seguem dois exemplos de URA's, mostrando os comandos que um usuário deve dominar para construir manualmente uma Unidade de Resposta Audível na plataforma Asterisk.

13.1.1 URA sem suporte a banco de dados no Asterisk.

Para construir uma URA sem suporte a banco de dados, a instalação padrão do Asterisk é o único requisito que deve ser contemplado.

Em seguida, será demonstrado como criar uma URA que disponibilize um serviço de consulta sobre meteorologia e tabua da maré, ambos limitados geograficamente ao estado de Santa Catarina.

Implementação (via arquivo `extension.conf`)

```

exten => 9000,1,Goto(ura-previsao,s,1)
[ura-previsao]
exten => s,1,PlayBack(msg-abertura)
exten => 1,1,Goto(serv-metereologia,s,1)
[serv-metereologia]
exten => s,1,Background(sel-serv-met-regiao)
    exten => 1,1,PlayBack(serv-met-litoralsul,s,1)
    exten => 1,2,Goto(serv-met-menu2,s,1)
    exten => 2,1,PlayBack(serv-met-planaltosul,s,1)
    exten => 2,2,Goto(serv-met-menu2,s,1)
    exten => 3,1,PlayBack(serv-met-grandefpolis,s,1)
    exten => 3,2,Goto(serv-met-menu2,s,1)
    exten => 4,1,PlayBack(serv-met-altovaladoitajai,s,1)
    exten => 4,2,Goto(serv-met-menu2,s,1)
    exten => 5,1,PlayBack(serv-met-meiooestevaleriodopeixe,s,1)
    exten => 5,2,Goto(serv-met-menu2,s,1)
    exten => 6,1,PlayBack(serv-met-oeste,s,1)
    exten => 6,2,Goto(serv-met-menu2,s,1)
    exten => 7,1,PlayBack(serv-met-planaltonorte,s,1)
    exten => 7,2,Goto(serv-met-menu2,s,1)
    exten => 8,1,PlayBack(serv-met-litoralnorteemediovale,s,1)
    exten => 9,2,Goto(ura-previsao,s,1)
[serv-met-menu2,]
    exten => 1,1,Background (msg-menu2)
        exten => 1,1,Goto(ura-previsao,s,1)
        exten => 2,1,Goto(rmsg-encerramento,s,1)
[msg-encerramento]
    exten => 1,1,PlayBack(msg-encerramento,s,1)

```

Abaixo são apresentados os comandos utilizados no desenvolvimento desta ura.

PlayBack: Utilizada para tocar um arquivo de som, este comando ignora qualquer tipo de comando enviado pelo usuário afim de interagir com o sistema durante a execução do referido arquivo.

Sintaxe – `exten => [chave],[prioridade],Playback(arquivo_audio)`

Exemplo - `exten => s, 1, PlayBack (msg-abertura)`

Background: Este aplicação executa um arquivo de som, porém ao receber um comando digitado pelo usuário, a execução do áudio é interrompida, e o fluxo é encaminhado para a extensão que o usuário escolheu;

Sintaxe - `exten => [chave],[prioridade],Background(arquivo_audio)`

Exemplo - `exten => s, 1, Background (msg-abertura)`

Goto: Esta aplicação é utilizada para efetuar a troca (saltos) de contextos, extensão ou prioridade no Asterisk.

Sintaxe -

```
exten => [chave],[prioridade], GOTO([[contexto_destino]extensão][prioridade])
```

Exemplo -

```
exten => 1, 1, Goto(serv-metereologia,s, 1)
```

13.1.2 URA com suporte a banco de dados no Asterisk.

Uma URA com suporte a banco de dados MYSQL (MYSQL,2007) acessando este diretamente via comandos predefinidos no arquivo extension.conf, deve possuir além da instalação padrão do Asterisk, a instalação do pacote asterisk-addons. O banco de dados pode ser acessado remotamente.

O exemplo que segue, demonstra como desenvolver uma Unidade de Resposta Audível que disponibilize um serviço de tele-voto. Neste exemplo o usuário acessa o sistema, que solicita a sua senha (cadastrada no banco), sendo que o acesso somente é liberado com a combinação correta da identificação do usuário e sua respectiva senha. Há proteção para evitar que o usuário vote mais de uma vez, bem como limita 2 tentativas de senha por ligação.

```
exten => 999,1,Goto(pesquisa,s,1)

[pesquisa]
exten => s,1,PlayBack(msg-abertura)
exten => s,2,MYSQL(Connect connid localhost root root asterisk)
exten => s,3,Read(identificacao|solicita-identificacao|4)
exten => s,4,MYSQL(Query resultid ${connid} SELECT nome\ FROM\ eleitor\ WHERE id=${identificacao})
exten => s,5,MYSQL(Fetch fetchid ${resultid} resultado1)
exten => s,6,GotoIf($${resultado1} = NULL)?8:10;
exten => s,7,MYSQL(Clear ${resultid})
exten => s,8,MYSQL(Disconnect ${connid})
exten => s,9,Goto(verifica-voto,s,1)
exten => s,10,Goto(solicita-identificacao-correta,s,1)

[solicita-identificacao-correta]
exten => s,1,MYSQL(Connect connid localhost root root asterisk)
exten => s,2,Read(identificacao|solicita-identificacao|4)
exten => s,3,MYSQL(Query resultid ${connid} SELECT nome\ FROM\ eleitor\ WHERE id=${identificacao})
exten => s,4,MYSQL(Fetch fetchid ${resultid} resultado1)
exten => s,5,GotoIf($${resultado1} = NULL)?8:10;
exten => s,6,MYSQL(Clear ${resultid})
exten => s,7,MYSQL(Disconnect ${connid})
exten => s,8,Goto(verifica-voto,s,1)
exten => s,9,PlayBack(identificacao-incorreta-repetida) *****
exten => s,10,Goto(msg-encerramento)

[verifica-voto]
exten => s,1,MYSQL(Connect connid localhost root root asterisk)
exten => s,2,MYSQL(Query resultid ${connid} SELECT id_candidato\ FROM\ pesquisa\ WHERE id_eleitor=${identificacao})
exten => s,3,MYSQL(Fetch fetchid ${resultid} r_identificacao)
exten => s,4,MYSQL(Disconnect ${connid})
exten => s,5,GotoIf($${r_identificacao} = "0")?6:7)
exten => s,6,Goto(solicita-voto,s,1)
exten => s,7,Goto(voto-repetido,s,1) ;

[solicita-voto]
exten => s,1,Read(voto|solicita-voto|1)
exten => s,2,Goto(computar-voto)s,1)

[computar-voto]
exten => s,1,MYSQL(Connect connid localhost root root asterisk)
exten => s,2,MYSQL(Query resultid ${connid} UPDATE\ pesquisa\ SET\ id_candidato=${voto}\ WHERE\ id_eleitor=${identificacao})
exten => s,3,MYSQL(Fetch fetchid ${resultid} resultado1)
exten => s,4,MYSQL(Clear ${resultid})
exten => s,5,MYSQL(Disconnect ${connid})
exten => s,6,PlayBack(voto-computado)
exten => s,7,Goto(msg-encerramento)

[voto-repetido]
```

Abaixo são apresentados os comandos utilizados no desenvolvimento desta URA.

Read: Um dos comandos mais importantes para a criação de URA no Asterisk , pois devido aos parâmetros que o mesmo pode suportar, este é capaz de executar um arquivo de áudio, limitar a quantidade de dígitos que o usuário pode digitar, o tempo máximo de espera para receber a interação com o usuário, além de salvar em uma variável o conteúdo digitado pelo usuário.

Sintaxe -

```
exten => s,1,Read(voto/solicita-voto/1)
```

Exemplo -

```
exten = > [chave],[prioridade],[Read(var_digitada[arquivo_audio][[max_dig][[opções][[tentativas][[timeout])
```

GotoIF: Esta aplicação é utilizada para efetuar a troca (saltos) de contextos, extensão ou prioridade no Asterisk, porém diferente da aplicação Goto, este comando testa uma condição antes de efetuar o deslocamento do fluxo de execução da URA.

Sintaxe -

```
exten => [chave],[prioridade],[GotoIF([condicao],label1,label2)]
```

Obs: Sendo label a extensão e/ou contexto para qual o fluxo será desviado (sendo o label1 caso a condição proposta seja verdadeira);

Exemplo -

```
exten => s,5,GotoIf("${r_identificacao}" = "0")?6:7)
```

MYSQL: Esta aplicação adiciona ao Asterisk as funcionalidades básicas de um banco de dados MYSQL. Sua sintaxe é um pouco mais complexa que as apresentadas anteriormente, pois a necessidade de conexão e desconexão com o banco de dados para liberar as instâncias utilizadas durante a consulta , além de limpar as variáveis envolvidas.

Conexão com o banco

Sintaxe -

```
Exten = > [chave],[prioridade],[ MYSQL(Connect connid IP_BD usuário_BD senha nome_da_base)]
```

Exemplo -

```
exten => s,1,MYSQL(Connect connid localhost root root asterisk)
```

Desconexã

o

Sintaxe -

```
Exten = > [chave],[prioridade],[ MYSQL(Disconnect ${connid})
```

Exemplo -

```
exten => s,0,MYSQL(Disconnect ${connid})
```

Construindo uma query

Sintaxe -

```
Exten = > [chave],[prioridade],[ MYSQL(Query var_resultado ${connid} query-string)
```

Exemplo -

```
exten => s,1,MYSQL(Query resultid ${connid} SELECT\ id_candidato\ FROM\ pesquisa\ WHERE\ id_eleitor=${identificacao})
```

Salvando o resultado de uma consulta em uma variável.

```
Exten = > [chave],[prioridade],[ MYSQL(Fetch fetchid ${resultid} var1)]
```

Sintaxe -

Exemplo -

```
exten => s,4,MYSQL(Fetch fetchid ${resultid} resultado1)
```

Obs: a variável resultado1 receberá o conteúdo de resultid

14 Ferramentas utilizadas para criar URA no Asterisk

Não foram localizadas ferramentas específicas para desenvolvimento de URA. O que existem são ferramentas de configuração de Plano de Discagem (Dialplan) e ferramentas de configuração completa do servidor Asterisk.

Segue uma breve análise de ferramentas disponíveis a usuários finais que desejam configurar uma URA no Asterisk.

14.1 Visual Dialplan Beta

Este aplicativo, contempla todo o escopo de configuração do arquivo extension.conf do sistema Asterisk, portanto seu propósito é controlar via arquivo de configuração, todos os canais configurações no sistema e suas respectivas ligações(entrantes ou saintes).

Como a uma grande gama de configurações que podem ser efetuadas no Asterisk, e o Visual Dialplan possibilita a configuração visual (em partes) de todas estas possibilidades, a interface acaba por ficar demasiadamente carregada, fazendo com que um usuário leigo não saiba por onde iniciar a configuração.

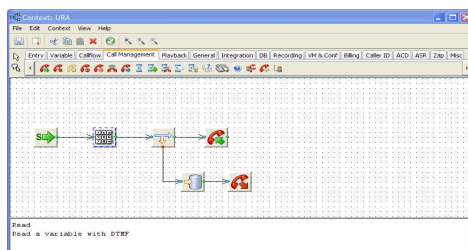


Figura 14.1 - Visual Dialplan / Menu para edição do arquivo extension.conf

Todavia, o sistema disponibiliza um menu de ajuda muito vasto, que além de informações sobre as aplicações, ainda demonstra com exemplos (figuras) como é realizada cada configuração.

Ainda como diferencial, o Visual Dialplan permite que o usuário selecione a release do Asterisk para qual o arquivo de configuração esta sendo gerado, uma vez que a versão 1.4 do Asterisk criou aplicações, e estas não podem ser executadas em sistemas com release anteriores, sob pena de inviabilizar o funcionamento da nova configuração.

Devido ao propósito geral do mesmo, a confecção de uma URA é uma atividade um tanto quanto árdua, pois as configurações (módulos) necessários estão espalhados pelas diversas abas do sistema de configuração, sendo que alguns dos ícones não são instintivos.

Porém um dos principais pontos negativos deste aplicativo é a dependência de sistema operacional, neste caso, a exclusividade da plataforma Windows. Aparentemente é uma incoerência, uma vez que o Asterisk é um dos principais sistemas Software Livre, sendo sua plataforma padrão o Unix.

Distribuição: Este software é proprietário, porém há uma distribuição beta, disponibilizada pela internet para testes.

Versão: 1.0.00-b

Desenvolvido: Apstel.

Site: <http://www.apstel.com>

14.2 Dialplanner

Com um foco em usuários mais experientes, que buscam apenas automatizar os códigos referentes a entradas, variáveis e aplicações, sem a necessidade de visualizarem a seqüência de execução da URA de forma gráfica, esta ferramenta acaba por restringir o número de usuários, uma vez que a necessidade de conhecer o funcionamento em nível de arquivos de configuração

para poder efetuar a seqüência correta (não apenas válida, mas que tenha um significado ao ser executado).

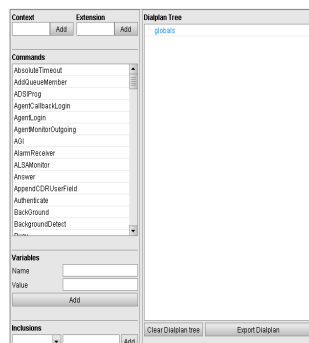


Figura 14.2 - Dialplanner / Menu Principal

Esta ferramenta possui uma grande vantagem, que é independência de plataforma, pois trata-se de um applet Java [JAVASE, 2007] que é executado diretamente no browser, sendo necessário apenas a instalação do Java.

A falta de um sistema de ajuda on-line também é um fato negativo. Outro ponto negativo é a pluralidade de opções de comando (todas as aplicações do Asterisk) unidas em apenas uma lista (mais de 160 opções), o que acaba causando uma grande perda de tempo para o usuário localizar a opção desejada.

A falta de validação do código gerado também é um ponto que necessita de ajustes por falta do desenvolvedor, evitando que o código seja migrado para o Asterisk e seja validada via console pelo usuário.

Distribuição: Gratuita.

Versão: não informada.

Desenvolvido: Lanvik ICU Sdn Bhd

Site: <http://www.lanvik-icu.com/asterisk/dialplanner/index.php>

14.3 ACTOS - Asterisk Configuration Tool Open Source

Esta ferramenta é utilizada para gerenciar os parâmetros de configuração mais importantes na gerência de um sistema Asterisk. Um destaque é a função que busca os arquivos de configuração, tanto na plataforma onde a ferramenta esta instalada, quanto em um servidor remoto.

A configuração é feita de forma que apenas usuários familiarizados com os arquivos e aplicações do Asterisk possam realizar as operações de maneira correta, uma vez que não há indicação de símbolos e os arquivos de ajuda não foram implementados na versão atual.

Como a aplicação é de propósito geral (todas as aplicações estão disponíveis nos menus), um usuário leigo teria dificuldade para montar uma URA com facilidade, além deste fato, não existe o recurso visual (similar a um

fluxograma), para que o usuário acompanhe a evolução da implementação de sua aplicação.

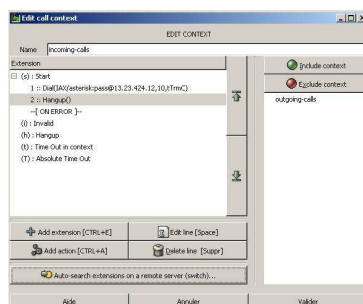


Figura 14.3 ACTOS / Menu do arquivo extension.conf (seqüência de execução)

Há versões deste software para Linux e Windows, porém a necessidade de escolher o sistema operacional onde o sistema será instalado, ou seja o conceito de multiplataforma não é totalmente respeitado.

Distribuição: Livre. Código Aberto (Open Source)

Versão: 2.25

Desenvolvido: Lanvik ICU Sdn Bhd

Site: <http://www.derrier.com/pierre/code/actos.html>

15 Ferramenta para desenvolvimento de URA no Asterisk

Com o intuito de propiciar ao usuário uma ferramenta que possibilite configurar uma URA no Asterisk, sem a necessidade do conhecimento em nível de arquivo de configuração (extension.conf), fazendo com que a arquitetura a ser criada (visualmente) seja uma seqüência de instruções com fluxos pré-determinados.

Para tanto, ao contrário das ferramentas de propósito geral (configuração completada do Asterisk), o módulo que será desenvolvido disponibilizará apenas as aplicações mais utilizadas para a construção de uma URA, facilitando a configuração para os usuários iniciantes.

15.1 Metodologia

Visando construir uma ferramenta que tornasse ágil o desenvolvimento de URA no Asterisk, subdividimos a criação desta em quatro etapas:

- A) Escolha de um modelo Gráfico para especificação da URA;
- B) Criação de um modelo de uma URA em XML (Extensible Markup Language)(URA-XML), representando textualmente o modelo gráfico definido;
- C) Desenvolvimento de um mecanismo de transformação do o arquivo URA-XML no arquivo extension.conf do Asterisk;
- D) Desenvolvimento de uma ferramenta gráfica para desenvolvimento de URA no Asterisk seguindo o modelo gráfico definido. Além disso, esta ferramenta incluirá o

processo de geração dos arquivos URA-XML e posterior geração do arquivo extension.conf.

15.2 Modelo Gráfico para Especificação de URAs

A configuração de uma URA no Asterisk é definida basicamente no arquivo extension.conf, este possibilita a configuração de estruturas condicionais, saltos e repetições, onde podemos inferir que sua estrutura é análoga a de um algoritmo.

Segundo PETRY(2005), existem três formas mais adotadas de representar um algoritmo:

- Descrição narrativa: algoritmos são representados em linguagem natural;
- Pseudocódigo: lista as ações a serem tomadas pelo computador, sem se preocupar com as regras rígidas da sintaxe da linguagem;
- Fluxogramas: um fluxograma é uma representação gráfica de um fluxo de controle, onde setas indicam a seqüência das tarefas (representadas por retângulos) e losangos representam os pontos de tomada de decisão;

Visando facilitar e agilizar o desenvolvimento de URA, optou-se por construir uma ferramenta que utilizará como modelo gráfico fluxogramas.

Graficamente, um fluxograma é representado pelos seguintes símbolos:





	Representa o início/fim de um fluxo.
	Indica a manipulação de dados.
	Ponto de tomada decisão.
	Indica o fluxo de controle. Interconecta os elementos do fluxograma.

Tabela15-1 Principais símbolos utilizados em um fluxograma

Segue exemplo de fluxograma de URA, para uma aplicação de tele-voto.

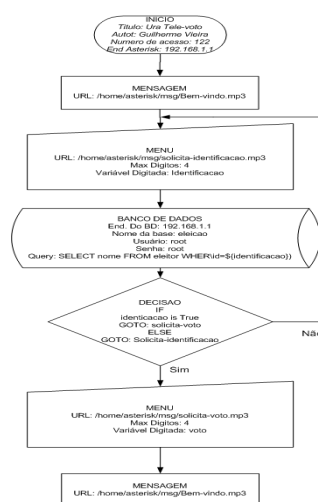


Figura 15.1 - Fluxograma URA - Tele Voto

15.3 Modelo de URA em XML

Para representar internamente o fluxograma criado pelo usuário, optou-se por descrever a estrutura em XML sendo a tradução schema XML para o arquivo `extension.conf` efetuada automaticamente pela ferramenta.

A opção por XML é justificada por esta ser uma linguagem semântica, que possibilita a gerência, troca e apresentações dos dados.

A XML é uma linguagem de marcação que utiliza tags para delimitação das informações. Ela foi desenvolvida pelo World Wide Web Consortium - W3C (W3C, 2007).

Como principais características do XML, podemos destacar: portabilidade, flexibilidade, verificação de erros, extensibilidade, representação estruturada dos dados e sem ambigüidades.

Alguns aspectos da linguagem a tornam mais eficientes, com a Interoperabilidade que permite a independência de plataforma ou de sistema operacional. Outro aspecto é a extensibilidade, que permite criar novas (próprias) regras ou elementos. Além destes, a propriedade de dados autodescritivos, facilita a reutilização em futuras aplicações.

Uma das bases da XML é o elemento. O nome do elemento deve facilitar o entendimento de seu conteúdo. A representação é feita por meio de marcações (tags), estas indicam o início e o fim dos elementos.

A sintaxe de um elemento pode ser representada da seguinte forma:

```
<nomeElemento>conteúdoElemento</nomeElemento>
```

Outro pilar da XML é o atributo, ao utilizado, deve-se declarar junto com a tag inicial do elemento, conforme o exemplo:

```
<nomeElemento nomeAtributo="texto ou valor" >
```

A descrição XML abaixo é referente a uma URA que disponibiliza a previsão meteorológica:

```

<ura-xml runat="endereço do asterisk">
<head>
  <metadata name="title" content="URA para votação" />
  <metadata name="author" content="Guilherme Vieira e Roberto Wilrich" />
  <db-name="base-ura" host="ip-db" user="root" passwd="*****" ... />
  <start accessNumber="122" />
</head>
<body>
  <audio id="abertura" src="file://..." />
  <input id="solicitaIdent" name="solicita identificador" msg-src="file://..." maxDigits="4" />
  <db-access db="base-ura" id="verificaVoto" var="user_no_validate" query="commando sql" />
  <if test="user_no_validate\ = true">
    <then>
      <goto dst="solicitaIdent" />
    </then>
  </if>
  <input id="solicitaVoto" var="voto" src="file://..." maxDigits="4" />
  <db-access db="base-ura" var="confirmaVoto" query="inserir o voto digitado na base-ura" />
  <if test="confirmaVoto\ = true">
    <then>
      <audio name="agradece voto com sucesso" src="file://..." />
    </then>
  </if>
</body>
</ura-xml>

```

erviço

```

<ura-xml runat="endereço do asterisk">
<head>
  <metadata name="title" content="URA metereologia" />
  <metadata name="author" content="Guilherme Vieira e Roberto Wilrich" />
  <start accessNumber="123" />
</head>
<body>
  <audio id="abertura" src="file://..." />
  <input id="solicitaReg" name="solicita região" msg-src="file://..." maxDigits="1" />
  <then>
    <audio name="previsao regioao X" src="file://..." />
  </then>
  <input id="menu2" name="divulga menu" src="file://..." maxDigits="1" />
  <if test="menu\ = true">
    <then>
      <audio name="mensagem de enerramento" src="file://..." />
    </then>
  </if>
  <else>
    <goto dst="solicitaReg" />
  </else>
</body>
</ura-xml>

```

Após a construção da URA pelo usuário (construção do fluxograma), cabe a aplicação a ser desenvolvida neste trabalho: representar a imagem criada pelo usuário em um arquivo XML.

Esta tradução é possível devido ao mapeamento das aplicações disponibilizadas ao usuário (via figuras de um fluxograma) em tags XML.

Baseado no exemplo do item anterior, uma mensagem de abertura no fluxograma recebe a seguinte representação:

MSG ABERTURA

Sendo mapeada da seguinte forma para o arquivo XML:

```
<audio id= "abertura" src= "file://..." />
```

15.5 Convertendo um arquivo XML para o formato padrão do Asterisk

De maneira análoga com o procedimento executado no item 7.3, foi criado um mapeamento das tags XML para o comando correspondente no arquivo extension.conf do Asterisk.

Desta forma, a tag XML que representa uma mensagem de abertura (apresentada no item 7.3), ficaria da seguinte forma no arquivo extension.conf:

```
exten => [chave],[prioridade],Playback(arquivo_audio)
```

15.6 Desenvolvimento da Ferramenta

A ferramenta proposta, ira possibilitar ao usuário montar um fluxograma da URA, exigindo pouco conhecimento sobre o funcionamento da mesma no Asterisk.

Esta aplicação ira salvar toda a estrutura da URA em um arquivo XML, que descrevera todas as particularidades utilizadas no processo de construção da aplicação proposta pelo usuário

16 Conclusão e Trabalhos Futuros

16.1 Conclusão

A evolução do sistema open source PBX Asterisk, tem quebrado vários paradigmas no mercado de Telecomunicações, um mercado até pouco tempo oligárquico, acompanha o crescimento e a rápida adoção deste sistema que é apontado por vários como o agente principal de uma nova revolução nas telecomunicações.

Várias aplicações são desenvolvidas por colaboradores (membros da comunidade open source) do sistema, e disponibilizadas para os demais usuários, com o intuito de facilitar e popularizar o uso do Asterisk. Este fato é um dos principais motivos para o sucesso e crescimento vertiginoso no número de novos servidores Asterisk instalados atualmente.

Entretanto, não foram localizadas ferramentas com o foco na construção de URA no Asterisk, neste momento encontramos o ponto de partida deste trabalho. O modelo de ferramenta proposto, baseia-se na criação de fluxogramas, para melhor visualizar os vários fluxos possíveis da URA, que posteriormente será codificado para o padrão do arquivo extension.conf do Asterisk.

Pode-se destacar também como vantagem deste modelo não requerer do usuário qualquer conhecimento prévio das aplicações do Asterisk, exige apenas lógica do mesmo, pois o intuito da ferramenta é propiciar ao usuário a criação de um fluxograma de sua URA

16.2 Trabalhos Futuros

- Criar módulo básico e avançado para a ferramenta, para que a ferramenta seja utilizada por usuários iniciantes e experientes.
- Validar em tempo real a aplicação para o usuário, fazendo com o que o mesmo não necessite importar o arquivo gerado pelo framework para o Asterisk para efetuar testes.
- Desenvolver uma interface gráfica que permita o usuário desenvolver URA utilizando diagramas SDL e UML.
- Possibilitar a integração do framework com scripts AGI, possibilitando a construção de URA's mais complexas, controladas por aplicações externas ao Asterisk.

17 Referências Bibliográficas

- [Asterisk, 2006] Asterisk :: An Open Source PBX and telephony toolkit. Disponível em: <<http://www.asterisk.org/>>. Acesso em: 26 nov. 2006.
- [BARREIROS, 2001] BARREIROS, Alessandra Natasha Alcântara. Estratégias de CRM para Empresas de Processamento de Dados Governamentais. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina, 2001.
- [CORIOLANO, 2003] CORIOLANO, Alice Cristina Campos. Voz sobre IP: Implementação e Teoria. Monografia (Graduação em Engenharia de Computação) – Universidade Federal de Goiás, 2003.
- [COSTA,2003] COSTA, João Carlos Peixoto de Almeida da. Implementação e Gerência de uma Arquitetura de Voz sobre IP. Dissertação (Mestrado em Ciências em Informática) - Universidade Federal do Rio de Janeiro, Núcleo de Computação Eletrônica, 2003.
- [Digium,2006] Digium – The Asterisk Company. Disponível em: <<http://www.digium.com/>>. Acesso em: 26 nov. 2006
- [FERNANDES, 2000] FERNANDES, Nelson Luiz Leal. Voz sobre IP: Uma Visão Geral. Rio de Janeiro, 2000. Disponível em: <http://www.ravel.ufrj.br/arquivosPublicacoes/nelson_voip.pdf>. Acesso em: 15 abr. 2007.
- [Ferreira et.al, 2006] FERREIRA, Dácio Miranda; LOPES, Paulo Roberto de Lima; SIGULEM, Daniel; PISA, Ivan Torres. Benefícios da Utilização do Session Initiation Protocol (SIP) em Aplicações de Comunicação Multimídia para a Saúde. In X Congresso Brasileiro de Informática em Saúde, 14-18 out. 2006.
- [GONÇALVES, 2005] GONÇALVES, Flávio Eduardo de Andrade. Asterisk PBX: Guia de Configuração. V. Office Networks, 2005.
- [JAVASE, 2007] Java Enterprise Edition, 2007. Disponível em: <<http://java.sun.com/javase/downloads/index.jsp>>. Acesso em: 17 mai 2007.
- [MYSQL, 2007] MySQL - The world's most popular open source database. Disponível em: <<http://www.mysql.com/>>. Acesso em: 10 abr. 2006
- [PETRY, 2005] PETRY, Patrícia Gerent. Um sistema para o ensino e aprendizagem de algoritmos utilizando um companheiro de aprendizagem colaborativo. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Santa Catarina, Departamento de PGCC, 2005. Disponível em: <<http://www.tede.ufsc.br/teses/PGCC0717.pdf>>. Acesso em: 2 out. 2007.

[PINHEIRO, 2004] PINHEIRO, Paulo Ricardo Guedes. Ciclos Evolutivos das Telecomunicações. Teleco, 2004. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialciclos/default.asp>>. Acesso em: 23 mar. 2007.

[RECEITAFONE, 2006] Ministério da Fazenda - Receita Federal - RECEITAFONE - 0300-789-0300. Disponível em: <<http://www.receita.fazenda.gov.br/pessoafisica/receitafone/>>. Acesso em: 26 nov. 2006.

[ROSENBERG, 2001] ROSENBERG, J.; PERTERSON, J.; SCHULZRINNE, H. Models for Multi-Party Conferencing in SIP. IETF Internet Draft, nov. 2001, work in progress.

[SMITH, MEGGELEN e MADSEN, 2005] SMITH, Jared; MEGGELN, Jim Van; MADSEN, Leif. Asterisk : O Futuro da Telefonia. Rio de Janeiro: Alta Books, 2005.

[W3C, 2007] World Wide Web Consortium. Disponível em <<http://www.w3.org>>. Acesso em 21 ago. 2007

Referências Bibliográficas

[Asterisk, 2006] **Asterisk :: An Open Source PBX and telephony toolkit.**

Disponível em: <<http://www.asterisk.org/>>. Acesso em: 26 nov. 2006.

[BARREIROS, 2001] BARREIROS, Alessandra Natasha Alcântara. **Estratégias de CRM para Empresas de Processamento de Dados Governamentais.**

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina, 2001.

[BORTOLUZZI, 2005] BORTOLUZZI, Dayna Maria. **Uma Extensão a Arquitetura da Internet pela Inserção de uma Camada de Sessão.** Tese (Doutorado em Engenharia de Produção Elétrica) - Universidade Federal de Santa Catarina, 2005.

[CORIOLANO, 2003] CORIOLANO, Alice Cristina Campos. **Voz sobre IP: Implementação e Teoria.** Monografia (Graduação em Engenharia de Computação) – Universidade Federal de Goiás, 2003.

[COSTA,2003] COSTA, João Carlos Peixoto de Almeida da. **Implementação e Gerência de uma Arquitetura de Voz sobre IP.** Dissertação (Mestrado em Ciências em Informática) - Universidade Federal do Rio de Janeiro, Núcleo de Computação Eletrônica, 2003.

[Digium,2006] **Digium – The Asterisk Company.** Disponível em: <<http://www.digium.com/>>. Acesso em: 26 nov. 2006

[FERNANDES, 2000] FERNANDES, Nelson Luiz Leal. **Voz sobre IP: Uma Visão Geral.** Rio de Janeiro, 2000. Disponível em:

<http://www.ravel.ufrj.br/arquivosPublicacoes/nelson_voip.pdf>. Acesso em: 15 abr. 2007.

[Ferreira et.al, 2006] FERREIRA, Dácio Miranda; LOPES, Paulo Roberto de Lima; SIGULEM, Daniel; PISA, Ivan Torres. Benefícios da Utilização do Session Initiation Protocol (SIP) em Aplicações de Comunicação Multimídia para a Saúde. In **X Congresso Brasileiro de Informática em Saúde**, 14-18 out. 2006.

[GONÇALVES, 2005] GONÇALVES, Flávio Eduardo de Andrade. **Asterisk PBX: Guia de Configuração**. V. Office Networks, 2005.

[JAVASE, 2007] Java Enterprise Edition, 2007. Disponível em: <<http://java.sun.com/javase/downloads/index.jsp>>. Acesso em: 17 mai 2007.

[MYSQL, 2007] **MySQL - The world's most popular open source database**. Disponível em: <<http://www.mysql.com/>>. Acesso em: 10 abr. 2006

[PETRY, 2005] PETRY, Patrícia Gerent. **Um sistema para o ensino e aprendizagem de algoritmos utilizando um companheiro de aprendizagem colaborativo**. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Santa Catarina, Departamento de PGCC, 2005. Disponível em: <<http://www.tede.ufsc.br/teses/PGCC0717.pdf>>. Acesso em: 2 out. 2007.

[PINHEIRO, 2004] PINHEIRO, Paulo Ricardo Guedes. **Ciclos Evolutivos das Telecomunicações**. Teleco, 2004. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialciclos/default.asp>>. Acesso em: 23 mar. 2007.

[RECEITAFONE, 2006] **Ministério da Fazenda - Receita Federal - RECEITAFONE - 0300-789-0300**. Disponível em:

<<http://www.receita.fazenda.gov.br/pessoafisica/receitafone/>>. Acesso em: 26 nov. 2006.

[ROSENBERG, 2001] ROSENBERG, J.; PERTERSON, J.; SCHULZRINNE, H. **Models for Multi-Party Conferencing in SIP**. IETF Internet Draft, nov. 2001, work in progress.

[SMITH, MEGGELEN e MADSEN, 2005] SMITH, Jared; MEGGELN, Jim Van; MADSEN, Leif. **Asterisk : O Futuro da Telefonia**. Rio de Janeiro: Alta Books, 2005.

[W3C, 2007] **World Wide Web Consortium**. Disponível em <<http://www.w3.org>>. Acesso em 21 ago. 2007