UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UMA ABORDAGEM BASEADA EM CONHECIMENTO PARA APOIO AO COMBATE ÀS PERDAS COMERCIAIS NA DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

HELDER BERGAMIN PIMENTEL DIAS

DISSERTAÇÃO DE MESTRADO EM INFORMÁTICA

Vitória
2006

BANCA EXAMINADORA:

Prof[a.] Ana Cristina Bicharra Garcia, PhD

Prof. Berilhes Borges Garcia, DSc

Prof. Flávio Miguel Varejão, DSc

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Dias, Helder Bergamin Pimentel, 1979-

541a

Uma abordagem baseada em conhecimento para apoio ao combate às perdas comerciais na distribuição de energia elétrica / Helder Bergamin Pimentel Dias. – 2006.

96 f. : il.

Orientador: Flávio Miguel Varejão.

Dissertação (mestrado) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Inteligência Artificial. 2. Representação do conhecimento (Sistemas especialistas). 3. Aquisição de conhecimento (Sistemas especialistas). 4. XML (Linguagem de marcação de documento). 5. Sistemas especialistas (Computação). 6. Energia elétrica - Distribuição. I. Varejão, Flávio Miguel. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 004

DEDICATÓRIA

Dedico esse trabalho à minha família. Ao meu pai, Vicente Pimentel Dias, que sempre me incentivou e sempre foi fonte de orgulho para mim, a minha mãe, Helena Maria Bergamin, que foi minha primeira e principal educadora, ao meu irmão, Hednei Vicente Bergamin Pimentel Dias, que me apoiou e sempre esteve ao meu lado, a minha esposa, Edina Machado da Silva Dias, que cumpriu comigo essa trajetória oferecendo-me apoio para prosseguir, a meu filho, Hugo Machado Bergamin Dias, que tanto amo e ao(s) filho(s) ou filha(s) que virá(ão) complementar essa linda família. Com muita felicidade, divido com estes as honras de concluir este trabalho, assim como espero compartilhar o que aprendi em prol de muitos que não tiveram a mesma oportunidade que eu tive.

AGRADECIMENTOS

Agradeço a Deus por tantas portas ter aberto em minha vida, e ter permitido fechar cada uma delas com sucesso. Agradeço a minha família pelo incentivo e apoio dedicado a mim a todo momento.

Agradeço a equipe do projeto que tanto contribuiu para meu crescimento e para a concretização desta dissertação, sejam eles alunos da graduação, colegas do mestrado, professores ou mesmo clientes – nossos queridos especialistas. Agradeço a Gabriela e ao Idílio que tanto colaboraram para a concretização deste, assim como também aos líderes do projeto - Flávio, Berilhes e Rodrigo Queiroga - que souberam guiar-nos a um bom final.

Agradeço aos professores e pesquisadores da UFES, em especial aos que lutam pela manutenção da qualidade do ensino público na instituição. Agradeço em especial ao meu orientador, Flávio Miguel Varejão, que compreendeu-me muitas vezes e apoiou-me para que este trabalho fosse concluído com sucesso, sendo não apenas um mestre, mas também um amigo.

SUMÁRIO

Lista de Figuras e Ilustrações

Resumo

As fraudes e irregularidades existentes na distribuição de energia elétrica geram grandes prejuízos para as concessionárias de energia elétrica. Uma forma tradicional de combate às fraudes é a realização de inspeções aos consumidores. Entretando, selecionar quais consumidores devem ser inspecionados é uma tarefa árdua para os especialistas no assunto. Esse trabalho visa analisar o conhecimento dos especialistas e propor um sistema baseado em conhecimento capaz de registrar e aplicar as heurísticas dos especialistas sobre o processo de seleção de suspeitos de fraude. O sistema é validado através de bases de exemplos reais de inspeções realizadas em clientes da concessionária local de energia elétrica e em ambiente real através de inspeções realizadas nos consumidores indicados pelo sistema baseado em conhecimento.

Abstract

Frauds and measure failures in Power Distribution cause great losses to Power Companies. A traditional way to combat these problems is performing field inspections on customers. However, selecting customers to be inspected is a difficult task for experts on such domain. The goal of the present work is to analyze such knowledge and propose a knowledge based system able to acquire, modify and apply heuristics obtained from the experts on the customer selection process. The system used a real customer database of the local power distribution company for selecting customers for inspection. The evaluation of the results was carried out through two different strategies: executing real field inspections on the selected customers and comparing the results with previously performed field inspections.

1 Introdução

O atual mercado de distribuição de energia elétrica tem sofrido sérios prejuízos devido a realização de fraudes e outras irregularidades que fogem ao controle das empresas. As fraudes, de modo geral, são um dos elementos de perda de receita destas empresas. Esse fator faz com que prevenir, detectar e combater as fraudes sejam elementos de grande importância para as corporações.

No setor de distribuição de energia elétrica, duas componentes formam as perdas globais das empresas: perdas técnicas e perdas comerciais - também denominas perdas não-técnicas. As perdas técnicas são relativas basicamente à própria natureza do serviço realizado. Perdas técnicas são perdas devido aos fenômenos físicos naturais de perdas de energia no decorrer de sua condução, transformação e distribuição. Por outro lado, perdas comerciais são em geral decorrentes de fraudes e irregularidades encontradas na rede de distribuição.

Ao se tratar de combate a perdas comerciais, o principal método utilizado pelas distribuidoras é a realização de inspeções nos consumidores. Estas inspeções têm a finalidade de detectar fraudes ou outras irregularidades como equipamentos manipulados ou defeituosos. Entretanto, o número de clientes das distribuidoras de energia elétrica é relativamente grande. A principal distribuidora de energia elétrica do Espírito Santo, por exemplo, ultrapassou 1 milhão de clientes no ano de 2005. Esse número elevado de clientes dificulta a fiscalização realizada através das inspeções. Outro fator negativo para a realização de inspeções em massa é o custo envolvido nestas operações. Quando as inspeções detectam fraudes e irregularidades, o custo envolvido na inspeção é coberto através da receita que se recupera, entretanto, quando se realizam muitas inspeções que não detectam fraudes, estas podem agravar os prejuízos das distribuidoras.

Por estes motivos, as distribuidoras empregam métodos para melhor direcionar as inspeções a fim de encontrar um maior número de fraudes com o menor esforço possível. Um destes métodos utiliza o conhecimento dos especialistas nestes assuntos agregado ao uso das informações contidas nas bases de dados das empresas.

As bases de dados das médias e grandes empresas atuais contêm uma enorme quantidade de dados disponíveis. Esses dados, quando bem utilizados por especialistas do domínio específico, podem revelar informações não triviais de grande importância para as

corporações. Porém, a utilização dessa grande massa de dados brutos é inviável através de análises manuais ou processos realizados sem ajuda substancial de máquinas.

No caso de identificação de fraudes em energia elétrica, não utilizar o potencial das heurísticas/regras conhecidas pelos especialistas sobre as bases de dados de consumidores seria um desperdício. Entretanto, aplicar esse conhecimento especialista sobre estas bases de dados é um trabalho um tanto custoso, pois exige tempo e esforço por parte do especialista para se aplicar uma determinada regra que pode demandar cálculos e utilização de dados de diferentes origens.

Outro fator importante é que gerenciar e acompanhar as evoluções nas heurísticas é algo complicado de ser feito e mantido de forma compartilhada em uma corporação, tanto que este conhecimento acaba ficando restrito a um ou outro especialista.

Assim, o objetivo deste trabalho é realizar um estudo sobre esse conhecimento especialista, propondo uma arquitetura de sistema que apóie o processo de seleção de suspeitos de fraude. Essa solução deve suportar o registro, a manipulação e a aplicação das heurísticas dos especialistas sobre os dados dos consumidores, indicando quais devem ser inspecionados. A partir deste sistema, será realizada uma pesquisa para realização do refinamento dos parâmetros que melhor se aplicam às heurísticas dos especialistas, contribuindo assim para que seja feito o uso mais eficiente destas heurísticas. Além disso, tem-se também o objetivo de registrar o conhecimento dos especialistas em uma base de conhecimento, de modo que este possa ser reutilizado quer seja por meio do sistema ou por outros usuários que venham a utilizar o ambiente.

Esta pesquisa, após realizado o estudo do problema e a identificação de suas características, resultou na proposta de um sistema baseado em conhecimento. Esse sistema possui uma base de conhecimento que foi formalizada em uma linguagem de representação proposta por este trabalho, seguindo a mesma linha de recentes iniciativas de representação de regras de produção que utilizam XML como estrutura.

A ferramenta proposta e desenvolvida permite que se escolham quais regras devem ser aplicadas e de que forma seus resultados devem ser apurados. Essas regras são aplicadas sobre uma massa de dados de investigação, de modo a extrair daí os resultados desejados: selecionar os consumidores que possuem indícios de serem fraudadores ou de possuírem irregularidades em suas instalações.

A distribuidora local de energia elétrica disponibilizou bases de dados com casos reais para a realização do refinamento das regras (espécie de treinamento onde se ajustou os parâmetros das regras). Após o refinamento, foram utilizadas outras bases de casos reais para a realização de testes. O sistema foi aplicado também em ambiente real. Utilizando a ferramenta sobre uma base de casos que a distribuidora desejava investigar, selecionou-se um grupo de consumidores suspeitos de fraudes ou irregularidades e foram geradas inspeções para averiguação. Os testes do sistema – por base de exemplos e por experimentação em campo – atingiram bons resultados em comparação a outras técnicas de seleção de suspeitos utilizadas pela distribuidora.

1.1 Estrutura da dissertação

O primeiro capítulo desta dissertação apresenta a introdução e a estrutura deste trabalho. O capítulo que se segue apresenta a revisão bibliográfica sobre aspectos que envolvem este trabalho como: sistemas baseados em conhecimento e particularidades em seu processo de construção e aquisição de conhecimento, detecção de fraudes e métodos para avaliação de resultados.

O terceiro capítulo abordará o problema tratado por este trabalho, comentando sobre as formas atuais de combate às perdas, os dados disponíveis para a análise e realização deste trabalho e o procedimento atual de seleção de suspeitos de irregularidades segundo o conhecimento especialista sobre o domínio do problema.

O capítulo 4 apresenta a solução proposta - citando a arquitetura planejada e as características especiais do projeto - assim como a linguagem definida e suas fórmulas (funções, expressões e operadores), finalizando com uma breve apresentação do que foi realizado na implementação.

O capítulo 5 descreve a experimentação do sistema, explica como ela foi realizada, demonstra e analisa os resultados em treinamento, teste e em campo (resultados reais). Encerrando esta dissertação, o capítulo 6 apresenta as conclusões e os trabalhos futuros que podem ser derivados deste.

2 Sistemas baseados em Conhecimento e Detecção de Fraudes

Este capítulo apresentará o contexto da tecnologia computacional no qual este projeto está envolvido. A primeira seção apresentará os Sistemas Baseados em Conhecimento (SBC), definindo-os e apresentando sua arquitetura e principais características, assim como as linhas gerais dos processos de aquisição do conhecimento e construção de SBC. Na primeira seção será apresentado também uma breve descrição sobre representação do conhecimento e sobre ferramentas de apoio a construção de SBCs. A seção seguinte tratará de aspectos de tratamento de dados para fins de melhoria da qualidade destes dados, explicando a importância desta etapa para o sucesso de SBCs. A última seção discorrerá sobre detecção de fraudes de modo geral e sobre detecção de fraudes na área de energia elétrica, citando exemplos e características desta área de estudo. Nesta última seção há uma subseção que se preocupará com a apresentação de um método para avaliação de resultados.

2.1 Sistemas baseados em conhecimento

Seja na área acadêmica, seja na área comercial, Sistemas Baseados em Conhecimento (SBC) são elementos de pesquisas e projetos de grande importância. Rezende et al. (2003) afirma que "esses sistemas devem ser usados quando a formulação genérica do problema a ser resolvido computacionalmente é complexa (tipicamente combinatória) e quando existe uma grande quantidade de conhecimento específico do domínio sobre como resolvê-lo".

Stefik (1995) define Sistema Baseado em Conhecimento como sendo um sistema computacional que representa e utiliza conhecimento para resolver problemas. Rezende et al. (2003) define SBC de modo similar a Stefik (1995), mas enfatiza o fato de que nesta classe de sistemas o conhecimento para resolver problemas é representado explicitamente. Essa representação explícita permite evoluções no conhecimento sem que seja necessário alterar o código do núcleo de processamento do sistema.

Rezende et al. (2003) afirma também que uma boa indicação a respeito do uso desta tecnologia é a existência de um especialista humano capaz de solucionar o problema.

A figura 2.1 apresenta a arquitetura básica de um SBC. Esta arquitetura está apresentada em Stefik (1995), e é composta essencialmente por quatro componentes.

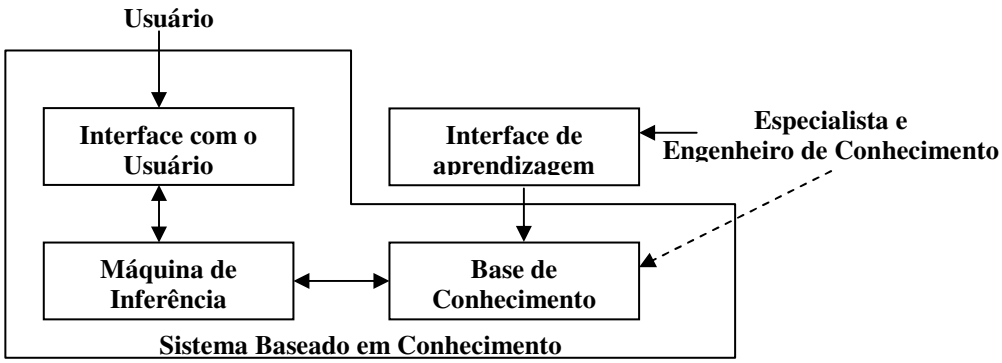

Figura 2.1 - Arquitetura básica de um SBC

A primeira componente é a interface com o usuário. Esta componente é parte do sistema que interage com os usuários do sistema. Em SBC que seja responsável por resolver problemas, por exemplo, é através desta interface que o usuário interage com o sistema a fim de solucionar um caso específico.

A segunda componente é a máquina de inferência, também denominada por alguns autores como núcleo do SBC. Esta componente é responsável pela interpretação e uso do conhecimento contido na base de conhecimento. A máquina de inferência possui um mecanismo de raciocínio capaz de realizar inferências sobre esta base e obter conclusões a partir deste conhecimento.

Uma outra componente de um SBC é a base de conhecimento. A base de conhecimento é o repositório onde está armazenado o conhecimento específico que o SBC se propõe a tratar. Nesta base de conhecimento podem-se construir sentenças em uma linguagem de representação, modelando o problema que se deseja resolver.

O segundo módulo de interface do sistema é a interface de aprendizagem. A interface de aprendizagem é essencialmente utilizada nas etapas de aquisição de conhecimento e em aprimoramentos do sistema. Ela permite que o conhecimento seja registrado, alterado e excluído da base de conhecimento. O número de usuários que utilizam essa interface é restrito. De modo geral, apenas os especialistas e engenheiros do conhecimento fazem uso desta interface. Em alguns sistemas, a representação do conhecimento pode ser realizada diretamente na base de conhecimento, dispensando a necessidade desta interface.

2.1.1 Construção de sistemas baseados em conhecimento

Como apresentado em Stefik (1995), e representado na Figura 2.2, o desenvolvimento de um SBC pode ser representado basicamente pelos seguintes estágios:

- Identificação: nesta etapa ocorre a primeira interação entre os participantes do projeto. Este estágio inclui a identificação de usuários e especialistas no domínio assim como a identificação geral do problema. Durante essa etapa se avalia se um sistema baseado em conhecimento é a solução adequada para o problema;
- conceitualização: durante este estágio os participantes do projeto desenvolvem um modelo externo da tarefa ou problema que está sendo analisado, assim como os processos de resolução deste problema ou de execução desta tarefa. Este modelo externo pode ser representado informalmente (em linguagem natural por exemplo);
- formalização: é a etapa onde ocorre a representação formal do conhecimento envolvido. Nesta etapa o modelo deve ser analisado ao nível de detalhes e representado formalmente.
- implementação: este estágio corresponde ao desenvolvimento de um protótipo operacional do SBC.
- testes e aplicação: nesta etapa o SBC deve ser testado e aplicado. Em um primeiro momento os testes podem ser executados com dados específicos para este fim. Uma vez validado, o sistema pode ser usado em ambiente real. Entretanto mesmo depois de validado e implantado, comumente há necessidade de re-executar partes do ciclo devido a naturais evoluções. Em especial, em se tratando de uso de conhecimento sobre um domínio específico, é natural que esse conhecimento precise ser atualizado ou aprimorado com o passar do tempo.

O diagrama da Figura 2.2 representa um processo de desenvolvimento que pode ser cíclico, onde a cada etapa ou ao final das etapas, permite-se uma avaliação do resultado atingido visando aprimorar o sistema. Esse refinamento pode exigir re-trabalho, retornando o processo para uma etapa anterior. Entretanto, como apresentado em Stefik (1995), o sucesso de um sistema baseado em conhecimento pode depender exatamente desta maior interação com o especialista ou usuário do sistema.

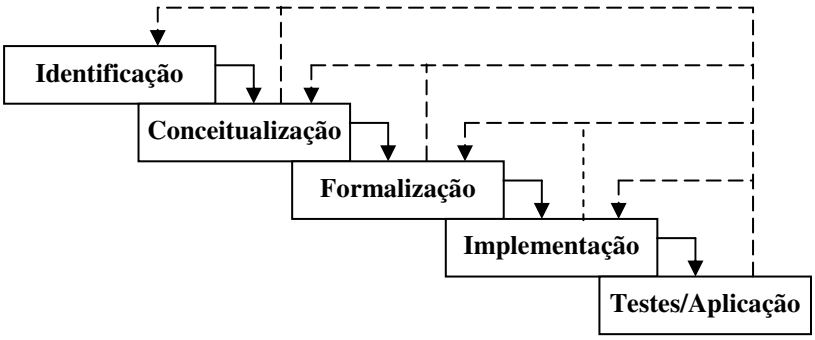

Figura 2.2 - Etapas do desenvolvimento de um SBC

O uso de processo de software assim como uso de outros requisitos fortemente defendidos pela área de Engenharia de Software (Pressman, 2001), deve também ser considerado quando se trata de desenvolvimento de SBCs (Rezende et al., 2003). Stefik (1995) discute que tais requisitos da Engenharia de Software podem não ser necessários em projetos acadêmicos ou mesmo em pequenos projetos, mas que estes requisitos são cruciais para o desenvolvimento de sistemas mais robustos e de sistemas aplicáveis a corporações.

Considera-se em geral, que a principal etapa da construção de um sistema baseado em conhecimento é a aquisição do conhecimento. Em se tratando das etapas definidas na Figura 2.2, pode-se considerar que as etapas de identificação, conceitualização e formalização compõem a etapa macro de aquisição do conhecimento. A próxima seção apresentará de forma mais detalhada o processo macro de aquisição do conhecimento.

2.1.2 Aquisição do Conhecimento

Durante a construção de um SBC, o conhecimento dos membros da organização necessita ser capturado, organizado e disponibilizado na Base de Conhecimento. Uma vez construída a base, esse conhecimento torna-se permanentemente acessível, mais facilmente recuperável e pode ser mais bem utilizado por todos, independentemente de sua capacitação. Contudo, adquirir esse conhecimento do especialista é extremamente trabalhoso e difícil. É por isso que a fase de aquisição de conhecimento é considerada o gargalo no desenvolvimento de tais sistemas.

Há duas abordagens principais para realização da aquisição de conhecimento: manual e automática. A aquisição manual é realizada por um Engenheiro de Conhecimento ou por uma equipe destinada a este fim. O Engenheiro de Conhecimento é o profissional responsável por conduzir a construção do sistema baseado em conhecimento, em especial no que diz respeito à formalização do conhecimento e a especificação do SBC. O trabalho deste profissional ou equipe é basicamente interagir com o(s) especialista(s) fazendo uso de

entrevistas, estudos de casos, estudo de problemas similares, acompanhamento das atividades dos especialistas e outros recursos para entender e posteriormente modelar esse conhecimento em um SBC.

A Figura 2.3 representa o processo manual de aquisição do conhecimento. Neste processo, o Engenheiro de Conhecimento interage diretamente com o especialista capturando e formalizando o conhecimento em uma base de conhecimento.

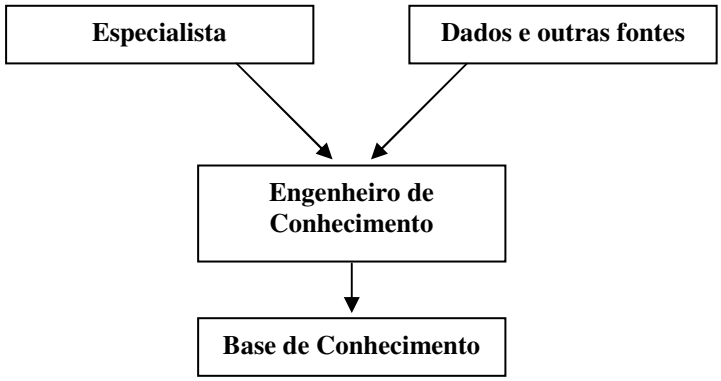

Figura 2.3 - Aquisição de conhecimento através do Eng. De Conhecimento

Outras técnicas de aquisição do conhecimento são consideradas automáticas ou ainda semi-automáticas uma vez que em muitos casos é necessária a intervenção do engenheiro de conhecimento ou do especialista mesmo com o uso de ferramentas computacionais.

Entre essas técnicas, está a tecnologia de aprendizado de máquina (Monard e Baranauskas, 2003) – na qual uma estratégia comum é o uso de exemplos para induzir regras – e a tecnologia de Mineração de Dados (Fayyad et al., 1996), técnica na qual analisam-se grandes bases de dados buscando extrair o conhecimento implícito contido nestas.

Na Figura 2.4 observa-se um diagrama que representa um exemplo desse modelo (automático). Neste exemplo, uma ferramenta computacional é aplicada a uma base de exemplos gerando uma base de conhecimento.

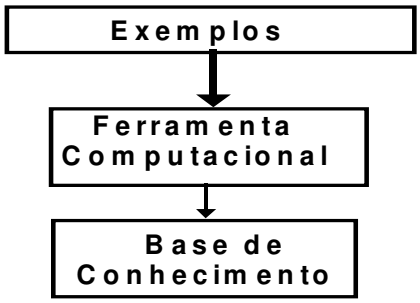

Figura 2.4 - Aquisição de conhecimento "automático"

Apesar de se considerar que o aprendizado é automático, de modo geral, há necessidade de intervenção do Engenheiro de Conhecimento ou mesmo do especialista validando ou refinando o conhecimento adquirido pelas ferramentas computacionais.

Em se tratando das técnicas manuais, existem várias técnicas para auxiliar a aquisição de conhecimento, tais como, entrevistas, análise de protocolo, acompanhamento das atividades do especialista, estudos de casos, entre outras. Dentre as técnicas manuais, as técnicas baseadas em entrevista são as mais comuns (Bicharra et al., 2003).

Considerando que neste trabalho o processo principal de aquisição do conhecimento foi realizado basicamente através da técnica de entrevistas, a próxima seção apresentará mais características sobre esse método de aquisição de conhecimento.

2.1.3 Entrevistas com o especialista

As entrevistas com o(s) especialista(s) normalmente são as principais fontes de informações para a aquisição do conhecimento a ser utilizado no sistema baseado em conhecimento. A captura das informações necessárias para resolver cada caso de análise processado pelo sistema baseado em conhecimento pode ser custosa e árdua tanto para o engenheiro de conhecimento quanto para o especialista.

O engenheiro de conhecimento precisará conhecer o domínio no qual o sistema será inserido. Assim, antes ainda das entrevistas, é interessante que o engenheiro de conhecimento estude sobre o assunto abordado. Mesmo com um estudo inicial, considera-se normal à ocorrência de assuntos e termos específicos do domínio que dificultam a interação entre o especialista e o engenheiro de conhecimento. O engenheiro de conhecimento precisa ser cauteloso para conduzir as entrevistas de forma produtiva, procurando realizar reunião após reunião onde a cada uma delas, o assunto é aprofundado, e tendo assim oportunidade de entre uma reunião e outra, buscar informar-se sobre determinado assunto, sempre tendo a cautela de

documentar minuciosamente o que se passou na reunião. Uma prática para não se perder informações das reuniões, é de registrá-las em áudio e vídeo ou ao menos em áudio, uma vez que de modo geral o volume de informações é alto para que se registre apenas em anotações.

Dentre diferentes tipos de entrevistas que podem ser utilizados, Bicharra et al. (2003) destaca três deles: entrevistas não-estruturadas, entrevistas estruturadas e acompanhamento de casos. Estes diferentes tipos de entrevistas são normalmente empregados em conjunto, sendo que cada tipo é utilizado de acordo com a necessidade momentânea do processo de aquisição do conhecimento.

As entrevistas não-estruturadas são conduzidas informalmente. Normalmente tais entrevistas são utilizadas em etapas iniciais da aquisição do conhecimento, momento onde se está conhecendo o problema ou uma nova categoria de problema dentro de um domínio já em estudo. Estas entrevistas fornecem maior abertura para a exposição do problema por parte dos especialistas.

As entrevistas estruturadas seguem um questionário ou roteiro melhor estabelecido que em relação às entrevistas não-estruturadas (que também podem possuir roteiro e planejamento). Muitas vezes, em entrevistas estruturadas, usam-se perguntas diretas, direcionadas a pontos específicos que o engenheiro de conhecimento deseja detalhar. Esse tipo de entrevista é importante para o enriquecimento de detalhes do problema que se está analisando.

O acompanhamento de casos é utilizado para completar o modelo de conhecimento que se está analisando. Esta etapa visa analisar casos junto ao especialista, visando corrigir ou completar as descrições e o entendimento do conhecimento adquirido através das entrevistas estruturadas e não-estruturadas.

Após o trabalho com as entrevistas, o engenheiro de conhecimento define a estrutura da base de conhecimento, e alimenta essa base com as informações adquiridas com os especialistas. Esta base de conhecimento será em seguida validada com o especialista e usada pelo sistema. A forma de representação do conhecimento nesta base de conhecimento é elemento de fundamental atenção em SBCs. A próxima seção apresentará um pouco sobre representação de conhecimento.

2.1.4 Representação do Conhecimento com regras de produção

O conhecimento capturado na fase de aquisição deve ser registrado na base de conhecimento do sistema. Esse registro deve ser realizado segundo uma linguagem específica que suporte todo o conhecimento do domínio da aplicação.

Rezende et al. (2003) lista algumas técnicas de representação do conhecimento freqüentemente utilizadas em SBCs, a saber: regras de produção, representação lógica, redes semânticas, frames, orientação a objetos e orientação a objetos associada a regras. Apresenta-se aqui a representação através de regras de produção por ser a utilizada neste trabalho.

As regras de produção são amplamente utilizadas como técnica de representação de conhecimento em SBCs, em especial ao se tratar de sistemas de detecção de fraudes baseados em regras (Rosset et al.,1999).

Usualmente as regras de produção seguem o formato "se <condições> então <ações/conclusões>". Esse formato facilita o uso deste modelo de representação principalmente pelo fato de ser de fácil compreensão. Outra vantagem é que a edição e manutenção das regras é relativamente simples devido a modularidade - cada regra pode ser uma unidade independente. A Figura 2.5 apresenta um exemplo de regra de produção.

***Se* <"há vazamento de combustível">**
***Então* <"acione o alarme">**

Figura 2.5 - Exemplo de regra de produção

Nos pares condição-ação, a condições a serem analisadas correspondem às premissas ou antecedente e a ação – conclusão, ação ou resultado - corresponde ao conseqüente. No caso do exemplo anterior, a condição ou premissa é "há vazamento de combustível" e a ação ou conseqüente é "acione o alarme".

Esta representação pode ser vista como simulação do comportamento cognitivo de especialistas humanos. É comum que as heurísticas dos especialistas sigam naturalmente essa estrutura "se <condições> então <ações/conclusões>". Esse fator favorece o uso desta técnica em sistemas baseados em regras.

Atualmente há alguns projetos que estabelecem linguagens para representação do conhecimento que utilizam regras de produção como estrutura. Um destes projetos é o CLIPS (CLIPS, 1985). CLIPS não é apenas uma linguagem, mas trata-se de uma ferramenta para desenvolvimento de sistemas baseados em conhecimento, e será apresentada na Seção 2.1.6.

Iniciativas recentes de definição de linguagens de regra de produção utilizam XML[1] como estrutura base. Uma dessas iniciativas, apresentada por Lee & Sohn (2003) é a linguagem eXtensible Rule Markup Language (XRML). Esta linguagem foi defina no KAIST

[1] Linguagem de demarcação simples e flexível usada para descrever dados. A demarcação utilizada é também chamada de *tag*, que corresponde a um elemento delimitado por "<" e ">" ou por "</" e ">".

- Korea Advanced Institute of Science and Technology. A Figura 2.6 apresenta um exemplo simplificado de regra definida em XRML – simplificado para facilitar o entendimento do leitor. Neste exemplo, o primeiro tag, o tag ***<Rule>*** engloba a regra definida. O segundo tag, o tag ***<RuleTitle>*** define o título da regra. O restante da regra trata-se do corpo da regra. Nota-se através dos tags ***<if>*** e ***<then>*** que há uma premissa e um conseqüente, seguindo a estrutura clássica das regras de produção.

```
<Rule>
<RuleTitle> Restriction of Type-P Research Fund Expenditure </RuleTitle>
<IF>
  <AND>
     <budgetary source>type-P research fund</budgetary source>
      <OR> <spendable item>student's salary </spendable item>
            <spendable item>expense for data collection</spendable item>
      </OR>
  </AND>
</IF>
<THEN>   <expenditure>permitted</expenditure> </THEN>
</Rule>
```

Figura 2.6 - Exemplo de regra em XRML

Assim como XRML, outros trabalhos utilizam XML como estrutura para representação do conhecimento. Um destes projetos é denominado RuleML (RuleML, 2006). RuleML, abreviatura de Rule Markup Language, trata-se de um projeto que propõe uma forma canônica de representação de regras de produção em XML.

2.1.5 Vantagens, Desafios e dificuldades na elaboração e uso de um SBC

Um dos principais benefícios da elaboração de um SBC é a representação do conhecimento em um sistema computacional – conhecimento este que antes era apenas dominado pelos especialistas. Tarefas que antes um especialista realizava com certo esforço, passam a ser auxiliadas por uma ferramenta computacional que lhe proporciona maior produtividade e muitas vezes maior qualidade. O fato das regras estarem registradas em um sistema computacional permite que essas possam ser aplicadas a uma maior quantidade de casos, situação que talvez não fosse possível de ser feita manualmente pelo especialista.

A qualidade do uso das regras também pode ser maximizada, pois à medida que as regras vão sendo definidas e validadas, sabe-se que toda a aplicação destas regras será feita de modo padrão, não correndo o risco de algum passo ser esquecido - o que seria possível no caso de atuação de um especialista humano, o qual pode realizar as mesmas tarefas de formas diferentes podendo em alguns casos fazer considerações diferentes de outros. O fato de o

especialista humano poder fazer considerações diferentes em situações similares, pode ser positivo, uma vez que se acrescenta crítica sobre o caso, ou pode ser negativo caso se esteja deixando de considerar algo importante.

A aplicação das regras pode ser mais eficaz com uso de um sistema computacional, e pode permitir testes com diferentes parâmetros na busca de um melhor resultado – um especialista talvez não conseguisse variar muito os testes em uma análise de um grande conjunto de casos.

Em situação de risco ou emergência, um especialista humano pode deixar-se influenciar pela situação perigosa e não atuar adequadamente em determinada decisão, fato este que não acontece com um sistema computacional (SBC), capaz de avaliar toda a base de conhecimento com a mesma qualidade em qualquer situação.

Outra vantagem de um sistema computacional é a possibilidade de sua aplicação e execução de modo ininterrupto sem onerar em demasia os custos de uma corporação.

Todavia, não só vantagens existem quando tratamos de sistemas baseados em conhecimento. Há várias dificuldades e limitações, a começar pelo fato das limitações do mesmo a sua base de conhecimento. Um sistema baseado em conhecimento fica restrito ao conhecimento existente em sua base, e ao conhecimento que vier a ser adicionado, enquanto um especialista humano é dinâmico e passível de perceber necessidades de adaptações de sua técnica de resolver determinado problema, a cada novo caso que surge. Essas adaptações poderiam ser incorporadas a um SBC em uso, mas essa versatilidade do especialista humano é difícil de ser simulada nos sistemas computacionais.

Um dos maiores desafios na construção de um sistema baseado em conhecimento é a aquisição do conhecimento e a sua manutenção. Apesar dos especialistas conhecerem bem o que fazem, muitas vezes eles não conseguem explicar o que fazem ou porque tomam determinada decisão, o que torna complexo o processo de formalização do conhecimento. Um fator que agrava essa situação é o receio da "máquina" substituir o especialista. Esse fato não é o objetivo dos sistemas baseados em conhecimento, mas pode ser assim interpretado por alguns funcionários ou especialistas.

Um outro desafio na construção de sistemas baseados em conhecimento é a dificuldade de se avaliar/prever como será o desempenho do sistema. É difícil saber se o conhecimento adquirido do especialista é suficiente ou não para tratar os casos reais. Dependendo do caso, apenas com testes em ambiente real consegue-se avaliar o desempenho do sistema, todavia nem todo ambiente em que se constrói um SBC permite que estes testes de validação e medição de desempenho sejam executados.

Um dos desafios na construção de um SBC é a disponibilização e dedicação de tempo à elaboração do sistema. Há necessidade de uma grande interação entre especialistas e engenheiro do conhecimento, o que nem sempre é fácil de se obter. De modo geral, os especialistas possuem atividades que os sobrecarregam e dificultam a sua participação na elaboração do sistema.

2.1.6 Ferramentas de apoio à construção de Sistemas baseado em conhecimento

A construção de sistemas baseados em conhecimento é normalmente apoiada pelo uso de ferramentas que auxiliem a sua realização. Tais ferramentas podem ser desde bibliotecas que implementem determinados módulos ou algoritmos utilizados em um SBC, a frameworks que se propõem a disponibilizar toda a arquitetura necessária para a construção destes sistemas.

Um exemplo de ferramenta deste tipo é o C Language Integrated Production System - CLIPS. CLIPS é uma ferramenta para desenvolvimento de sistemas baseados em conhecimento (CLIPS, 1985). Mais especificamente, CLIPS provê um ambiente para construção de sistemas baseados em regras e/ou sistemas especialistas baseados em objetos. CLIPS permite ainda o uso de linguagem procedural.

CLIPS oferece uma máquina de inferência e um ambiente para registro da base de conhecimento, podendo ser utilizado como sistema baseado em conhecimento a partir do simples registro de fatos e regras em sua base de conhecimento.

CLIPS surgiu em 1985 em um projeto da NASA, porém sua terceira versão, datada de 1986 fora disponibilizada para grupos externos a NASA, possibilitando que a ferramenta evoluísse de modo livre de propriedade.

A Figura 2.7 apresenta um exemplo de uma regra defina em CLIPS. A regra apresentada possui como condição "animal-is duck". Caso esta condição seja satisfeita, a regra é acionada inserindo um novo fato na lista de fatos.

```
(defrule duck          ; cabeçalho da regra
   (animal-is duck) ; condição
    =>                 ; então
   (assert (sound-is quack)) ; ação
)
```

Figura 2.7 - Exemplo de CLIPS

Uma outra ferramenta para desenvolvimento de SBC é o Expert SINTA (Nogueira et al., 1996). O Expert SINTA é uma ferramenta para geração de sistemas baseados em regras.

Esta ferramenta utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidades. O objetivo da ferramenta era simplificar o trabalho de implementação de sistemas especialistas através de uma ferramenta que integra uma máquina de inferência interna com uma interface amigável de edição de regras para a formalização do conhecimento. A evolução desta ferramenta foi descontinuada alguns anos após sua criação.

Na área de detecção de fraudes, o trabalho de Burge et al. (1997) propõe uma ferramenta híbrida para detecção de fraudes, denominada BRUTUS. Esta solução híbrida é composta por um módulo que utiliza redes neurais e um módulo que utiliza sistemas baseados em regras. Segundo Burge et al. (1997), os testes iniciais da ferramenta, utilizando dados do setor de telecomunicações, e visando detectar fraudes neste domínio tiveram bons resultados apesar de serem preliminares. Entretanto em seu artigo não foram dados mais detalhes sobre a ferramenta.

Uma outra ferramenta recentemente disponibilizada e ainda em desenvolvimento é o jDREW (jDREW, 2006). Essa ferramenta trata-se de uma API para JAVA que implementa uma máquina de inferência. Sendo jDREW uma API para JAVA, essa solução se propõe a ser um facilitador para o desenvolvimento de sistemas baseados em conhecimento em JAVA. jDREW utiliza RuleML – vide Seção 2.1.4 - como linguagem de representação de conhecimento.

2.2 Tratamento dos dados – Pré-processamento de dados

O processo de tratamento dos dados é importante para diversas tecnologias, como por exemplo para uso em Data Mining e Data Warehouse, e ao se tratar de SBCs, muitas vezes este processo também pode ser de extrema importância para o sucesso do sistema (Fayyad et al., 1996). De modo geral, as corporações possuem um grande volume de dados que pode ser explorado, e o uso deste volume de informações precisa ser cauteloso para dificultar o uso do conhecimento contido nos dados. O engenheiro de conhecimento, com auxílio dos especialistas, precisa identificar os dados relevantes que podem ser usados no sistema computacional.

Além de identificar informações relevantes, há também de se considerar que os dados de aplicações reais estão sujeitos a erros gerados por coletores de dados defeituosos, falhas humanas nas digitações dos dados, erros em transmissões de dados, dentre outros. Devido a este fator, os dados disponíveis precisam ser avaliados quanto a alguns aspectos:

- Limpeza: precisa-se verificar e tratar dados incorretos, inconsistentes, incompletos ou com ruídos;
- utilidade: tais dados oferecem informações reais ou se tratam de informações redundantes em relação a outras informações encontradas na base;
- existência de histórico: em caso de uma análise que necessite de um amplo histórico de dados para se extrair uma determinada informação, há informação histórica suficiente para esta análise;
- dimensão e tipo de dados: as informações estão em dimensões e tipo de dados adequados para o uso no sistema baseado em conhecimento e
- compatibilidade entre os dados: devido aos dados serem muitas vezes provenientes de diferentes sistemas, alguns destes dados podem estar em formato incompatível entre si. Por exemplo, datas podem estar representadas em diferentes formas nos sistemas existentes.

Dependendo da situação de cada dado a ser utilizado, há necessidade de realizar tratamentos específicos sobre estes dados. Wrightt (1996) apresenta um estudo sobre as principais atividades do pré-processamento dos dados. Abaixo são citados alguns dos tratamentos empregados na preparação de dados:

- Transformação: informações que estejam em tipos inadequados de dados precisam ser modificadas (transformadas).
- tratamento de valores discrepantes ou ausentes: dados nulos (*missing values*) devem ser analisados e em alguns casos tratados, verificando o motivo de sua inexistência ou o impacto que podem gerar. Do mesmo modo, campos com valores discrepantes (*outliers*) – dados fora do domínio previsto – precisam ser analisados e tratados;
- integração de dados: dados de diferentes fontes precisam ser acoplados de modo que possam ser utilizados em conjunto. Dados que possam ser relacionados mas que estejam em formatos incompatíveis, precisam ser modificados para adquirirem compatibilidade. Por exemplo, dados que possuem diferenças de unidades de medidas podem ser compatibilizados neste passo de integração;
- derivação de novos dados: a partir dos dados existentes na base, pode haver necessidade de criação de novos dados que representem novas informações.

Esses tratamentos dos dados precisarão ser realizados cada vez que os dados forem adquiridos de suas fontes originais. A preparação dos dados é uma etapa custosa quando comparada ao processo como um todo, porém, é uma etapa que precisa ser realizada de forma

adequada para garantir uma boa utilização destes dados, possuindo grande influência no resultado final (Engels & Theusinger, 1998).

2.3 Detecção de fraudes

A detecção de fraudes é elemento de estudo no meio acadêmico e item em foco nos ambientes coorporativos. Em diferentes setores há a mesma preocupação: prevenir, detectar e combater as fraudes. Em geral a detecção de fraudes envolve recuperação de receita e esse é um dos motivos pelos quais é um elemento de grande importância para as empresas.

Ao tratar de detecção de fraudes, necessita-se em geral analisar um grande volume de dados. Brause et al. (1999), por exemplo, em seu estudo sobre Data Mining na detecção de fraudes em cartões de crédito, indicou uma proporção de cerca de 1:1000 (1 fraude em 1000 transações), o que indica que para analisar um número significativo de fraudes, faz se necessário analisar um volume considerável de transações. Outro exemplo de um estudo envolvendo grande volume de dados é o de Bolton & Hand (2002) que cita casos como o do Royal Bank da Escócia que possui cerca de 2 bilhões de transações ao ano e ainda, outro exemplo, a Barclaycard com 350 milhões de transações ao ano em um único país.

Além de grandes bases de dados, há também a limitação com o acesso a esses dados e o sigilo envolvido nesse processo. Informações pessoais, informações sobre transações financeiras, entre outras, são em muitos casos consideradas confidenciais, e o acesso a dados desta natureza por muitas vezes é bloqueado ou limitado, o que pode dificultar a elaboração de sistemas que se propõem a investigar fraudes. Esses fatores tornam o processo um tanto complexo.

Um trabalho interessante por se tratar de uma arquitetura híbrida se encontra no trabalho de Burge et al. (1997). Burge propõe uma ferramenta para detecção de fraudes que utiliza redes neurais e sistemas baseados em regras (o trabalho de Burge foi comentado na Seção 2.1.6 desta dissertação).

Os estudos e trabalhos em investigação de fraudes estão presentes em diversos ramos de atividades. Há estudos na área de cartões de crédito e instituições financeiras como citado em Brause et al. (1999) e Bolton & Hand (2002), assim como na área de sistemas computacionais (Kou et al., 2004) e na área de energia elétrica, que é o foco deste trabalho.

A atividade de fornecimento de energia elétrica é, assim como várias outras atividades, alvo comum de fraudes. Por se tratar de um serviço disponibilizado a uma grande clientela, e de difícil monitoramento e controle de fraudes, muitos são os casos de fraudes

neste setor. As fraudes podem ser simples desvios instalados diretamente na rede elétrica ou manipulações mais complexas como violações ao equipamento de medição e alterações no mesmo para que este não realize corretamente a leitura.

Em Eller (2003), apresenta-se um estudo sobre o gerenciamento de perdas comerciais de energia elétrica, no qual se propõe uma arquitetura que atua na indicação de possíveis fraudadores e ainda no gerenciamento de perdas comercias. Neste trabalho de Eller (2003), usou-se mineração de dados (Fayyad et al., 1996), em especial, destacou-se o uso de redes neurais (Haykin, 2001), e suas conclusões foram que esta é uma arquitetura com potencial, mas que ainda necessitava aprimorar os modelos para que os resultados fossem mais eficazes. Outro trabalho em detecção de fraudes de energia elétrica está apresentado em Cabral (2004), no qual se utilizou rough sets como algoritmo para seleção de características de maior potencial para utilização em ferramentas de classificação.

Um outro trabalho também envolvendo mineração de dados foi apresentado em Queiroga (2005), no qual um amplo estudo foi realizado na análise de perdas comerciais, envolvendo diversos modelos computacionais para a geração de classificadores, a citar:

- redes neurais (Haykin, 2001);
- naive bayes (Mitchel, 1997);
- bayes netk2 (Mitchel, 1997);
- bayes netb (Mitchel, 1997);
- one nearest neighboor (Cover & Hart, 1967);
- k nearest neighboor (Cover & Hart, 1967);
- indutor de regras ID-3 (Quinlan, 1986) e
- indutor de regras J4.8 (Quinlan, 1986).

No trabalho de Queiroga (2005), foram realizados testes com as diversas técnicas citadas acima, utilizando diferentes bases de dados, e os resultados atingidos em campo (resultados reais) se aproximaram dos resultados previstos no treinamento.

Este trabalho faz parte do mesmo escopo da pesquisa de Queiroga (2005). Enquanto o trabalho de Queiroga analisou o uso de classificadores a partir de algoritmos de aprendizado de máquina, este trabalho visa o uso do conhecimento dos especialistas, empregando regras para a pesquisa de suspeitos de fraude.

2.3.1 Método para avaliação de resultados

Como método para avaliar o desempenho do sistema definido neste trabalho sobre as bases de exemplos de testes, optou-se por fazer uso do mesmo método utilizado em Cometti et al. (2005), Queiroga (2005) e Drago (2005). Estes trabalhos utilizam-se de indicadores calculados a partir da matriz de confusão resultante dos experimentos que se está avaliando.

Conforme apresentado em Monard & Baranauskas (2003) a matriz de confusão oferece uma medida efetiva do modelo de classificação, ao mostrar o número de classificações corretas versus as classificações preditas. Basicamente, a matriz de confusão confronta os resultados reais dos exemplos com a classificação realizada pelo sistema. A partir desta confrontação dos dados, calculam-se algumas métricas: taxas de acerto, especificidade, entre outras, que serão definidas em seguida. Estas medidas também podem ser consultadas em Monard & Baranauskas (2003).

A Figura 2.8 apresenta um modelo de matriz de confusão para facilitar a explicação e compreensão do leitor.

	Classificados como:	
Situação Real:	**Normal**	**Fraude**
Normal	**a**	**b**
Fraude	**c**	**d**

Figura 2.8 - Matriz de confusão

Em uma matriz de confusão, os dados contidos nas células indicam o número de exemplos que possui a referente classificação, sendo que a classificação indicada pela linha, indica sua classificação/situação real, enquanto a classificação indicada pela coluna indica a classificação dada pelo sistema. Por exemplo, para a matriz de confusão da Figura 2.8, 'b' representa o número de casos que na base de exemplos estão classificados como 'normal' e o sistema classificou como 'fraude'. Seguindo este raciocínio, ao se observar as linhas:

- 'a' e 'b': indicam exemplos onde a classificação real é 'normal',
- 'c' e 'd': indicam exemplos onde a classificação real é 'fraude'.

Por outro lado, as colunas da matriz de confusão indicam a classificação realizada pelo sistema:

- 'a' e 'c': indicam exemplos classificados pelo sistema como 'normal',
- 'b' e 'd': indicam exemplos classificados como 'fraude'.

Assim, sabemos que 'a' são os exemplos normais que foram classificados como normais. Enquanto 'b' são exemplos normais que foram classificados como sendo fraudadores. Da mesma forma, 'c' são fraudadores que foram classificados como normais e 'd' são fraudadores que foram classificados como fraudadores.

A primeira métrica a ser calculada trata-se da taxa de acerto do sistema. Entretanto, no domínio de fraudes em distribuição de energia elétrica, esta métrica sozinha não é muito significativa uma vez que o mais importante são as fraudes detectadas. Se considerarmos apenas a taxa de acerto, em determinadas situações, poder-se-ia ter um sistema que acerta 80% das classificações simplesmente classificando todos os casos como 'normal', mas não detecta nenhum fraudador. Por este motivo, são calculadas e avaliadas outras métricas, a fim de encontrar resultados mais satisfatórios. Abaixo são definidas as métricas obtidas a partir da matriz de confusão utilizadas por Queiroga (2005):

1. Taxa de Acertos (*T*): total de classificações corretas dividido pelo total de exemplos de teste. (Figura 2.9)

$$T=(a + c) /(a + b + c + d)$$

Figura 2.9 - Taxa de Acerto

2. Especificidade (*e*): número de consumidores classificados corretamente como fraudadores sobre o total de fraudadores. (Figura 2.10)

$$e = c / (c + d)$$

Figura 2.10 - Especificidade

3. Confiabilidade Negativa (*c*): número de fraudadores classificados como fraudadores sobre o total de exemplos classificados como fraudadores (Figura 2.11).

$$c = d / (b + d)$$

Figura 2.11 - Confiabilidade Negativa

Em outras palavras, a métrica especificidade representa o percentual dos fraudadores que estão sendo classificados como fraudadores, e a métrica confiabilidade negativa demonstra dentre os classificados como fraudador qual o percentual de acerto.

Para o escopo deste trabalho, verificou-se que os especialistas desejavam um certo equilíbrio entre essas duas métricas, pois:

- Se considerar apenas a especificidade, pode-se ter um resultado muito bom em relação aos fraudadores encontrados, mas pode-se errar muito classificando muitos exemplos normais como fraudadores;

- já considerando apenas a confiabilidade negativa, pode-se ter uma boa taxa de acerto dentre os classificados como fraude, mas pode-se estar classificando muitos fraudadores como 'normais'.

Como o mais interessante para os resultados é ter um equilíbrio entre essas duas métricas, usou-se uma métrica baseada na média harmônica que faz uma ponderação entre especificidade e confiabilidade negativa.

A média harmônica ponderada é uma variação da métrica apresentada em Rijsbergen (1979). Esta métrica valoriza mais o equilíbrio entre duas variáveis do que em situações nas quais uma variável tem um valor muito alto e a outra um valor muito pequeno (Drago, 2005):

4. Média Harmônica: média harmônica ponderada entre a confiabilidade negativa e a especificidade. A Figura 2.12 apresenta esta fórmula da média harmônica em relação a suas componentes: especificidade (e) e confiabilidade negativa (c) – onde $\beta 1$ e $\beta 2$ são constantes que visam dar um peso maior a uma ou outra medida.

$$\text{Média harmônica} = e * c / (\beta 1\ e + \beta 2\ c)$$

Figura 2.12 - Média harmônica

3 Identificação de Perdas Comerciais na Distribuição de Energia Elétrica

O problema de perdas comerciais atinge de forma preocupante as distribuidoras de energia elétrica. Perda comercial é uma parcela das perdas globais registradas pelas distribuidoras de energia elétrica. A outra parcela das perdas globais são as perdas técnicas.

Perdas técnicas são definidas pelo Comitê de Distribuição de Energia Elétrica (Codi, 1997) como: "a energia perdida no transporte, na transformação e nos equipamentos de medição da energia elétrica quando do fornecimento da mesma". E perdas comerciais como "aquelas decorrentes da energia efetivamente entregue aos consumidores finais ou a outras concessionárias, mas não computada na venda" (Codi, 1997).

A Figura 3.1 apresenta uma visão apresentada em MCPT (2004) das componentes que formam as perdas das distribuidoras de energia elétrica. A parte superior indica as componentes relacionadas às perdas técnicas. A parte inferior indica as componentes de perdas comerciais.

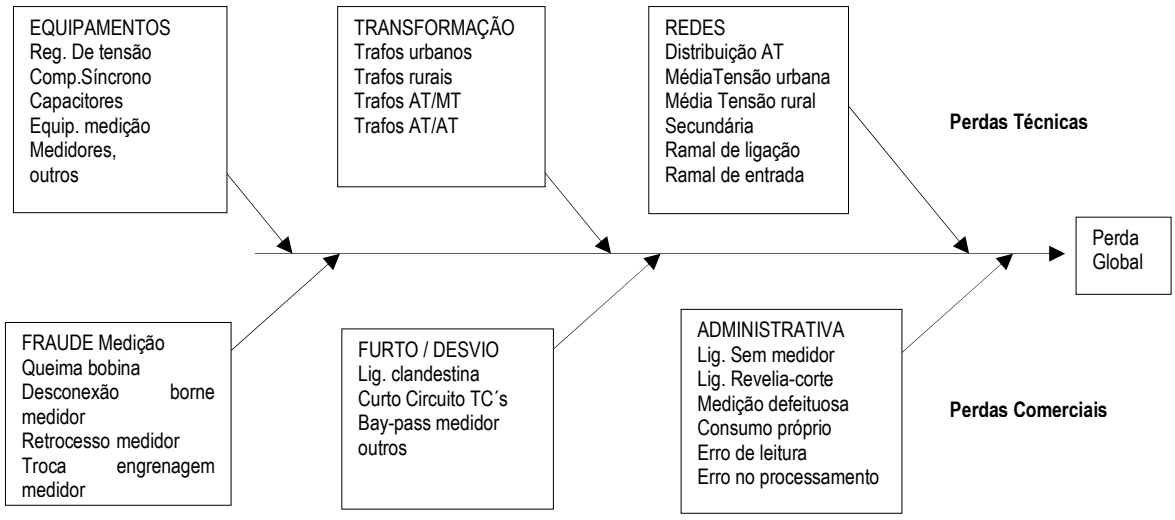

Figura 3.1 – Origens das perdas de energia (MCPT, 2004)

As perdas comerciais, objeto de estudo deste trabalho, são causadas de modo geral pelo uso irregular de energia ocasionado pela ação de terceiros ou por equipamentos defeituosos – ocorrem ainda perdas comerciais devido a erros de leitura, de processamento, consumo próprio, entre outros, mas esta parcela da perda comercial é pequena quando comparada às perdas por fraudes, desvios e irregularidades. Ações irregulares como desvios de energia realizados diretamente na rede de distribuição ou manipulação nos medidores para

que estes não funcionem corretamente são exemplos destas ações que ocasionam prejuízos às distribuidoras de energia elétrica, prejuízo esse, registrado como perda comercial.

Os desvios e fraudes nas redes elétricas ocorrem em todas as categorias de consumidores (residenciais, comerciais, industriais). Porém, segundo os especialistas, são nos clientes residenciais que se encontram o maior número destas irregularidades. A Figura 3.2 apresenta um caso de diversas ligações irregulares realizadas em um determinado poste.

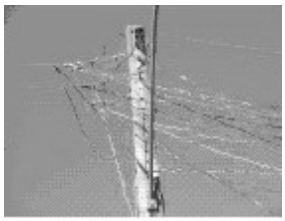

Figura 3.2 - Ligação irregular na rede de distribuição de energia.

3.1 Formas atuais de combate às perdas comerciais

As concessionárias de energia elétrica, também nomeadas distribuidoras de energia elétrica, combatem as perdas comerciais através de diferentes formas. A conscientização de que fraude nas instalações elétricas é um ato criminoso é um dos meios empregados. Utilizando-se da mídia através de propagandas em canais abertos, outdoors, folders entre outros recursos, as distribuidoras buscam realizar esta conscientização. Um outro recurso empregado é o estudo e utilização de equipamentos que cada vez mais dificultem a realização de fraudes nas redes de distribuição.

Apesar do uso destes e de outros recursos, a principal forma de combate às perdas comerciais são as inspeções aos consumidores. Essas inspeções são visitas realizadas a um determinado consumidor, onde realiza-se uma averiguação em sua instalação e investigam-se possíveis fraudes ou equipamentos defeituosos.

A ESCELSA – Espírito Santo Centrais Elétricas -, concessionária que distribui energia para a maior parte do Espírito Santo, atua com os seguintes tipos de inspeções:

- Inspeções de varredura: em regiões onde as perdas comerciais estão muito altas, realizam-se inspeções em todas as unidades consumidoras, autuando as unidades surpreendidas com fraudes no momento da inspeção e adequando equipamentos defeituosos a fim de diminuir as perdas;
- inspeções de medição indireta: em unidades consumidoras que utilizam medição indireta (consumidores geralmente industriais ou comerciais, com utilização de energia em média

tensão). A distribuidora optou por se programar para realizar ao menos uma inspeção anual em cada consumidor deste grupo;

- inspeções por denúncia: são inspeções geradas a partir de denúncias anônimas registradas pela central de atendimento da empresa;
- inspeções por indicação de leituristas: ao realizar a leitura, os leituristas observam as condições das instalações do medidor, e se suspeitarem de alguma anomalia, indicam a unidade para ser inspecionada;
- inspeções à pedido do cliente: os clientes também podem solicitar inspeções e nestes casos, apesar das fraudes ocorrerem com muita raridade, o ponto maior de investigação é a da situação do medidor;
- inspeções de consumo zero: para unidades consumidoras que permanecem um longo período com um consumo muito abaixo do mínimo, geram-se solicitações de inspeção para averiguar se há alguma irregularidade;
- inspeções de clientes especiais: são inspeções realizadas nos clientes com demanda acima de 500 KW. Estas são realizadas periodicamente, de acordo com a disponibilidade das equipes de inspeção.

Todavia, as inspeções por si só não resolvem o problema das perdas comerciais, pois de modo geral o número de consumidores é muito grande e é inviável realizar um número de inspeções suficientes para manter o percentual de perda comercial em níveis aceitáveis.

Como o número de inspeções é pequeno perante o número de clientes das distribuidoras, uma forma de melhorar os resultados é a realização de inspeções de modo mais seletivo, buscando inspecionar consumidores com maior possibilidade de estarem fraudando. Para este fim, é essencial o uso de softwares que auxiliem na escolha de quais consumidores devem ser inspecionados.

Atualmente, no caso da ESCELSA, a maior parte das inspeções realizadas são programadas a partir da escolha das regiões com maiores perdas – normalmente se escolhem os alimentadores de maiores perdas comerciais. Com isso, realiza-se um grande número de inspeções que em geral não atingem bons resultados. No ano de 2006, iniciou-se o uso do sistema MIP (Melhoramento de Identificação de Perdas), apresentado no trabalho de Queiroga (2005), no qual se utilizam técnicas de mineração de dados para a melhor seleção das unidades consumidoras suspeitas.

Diferentemente do MIP, este trabalho visa analisar o conhecimento dos especialistas no assunto e propor um sistema que utilize essencialmente o seu conhecimento para realizar a

seleção de prováveis suspeitos de fraudes ou de possuírem equipamentos defeituosos. Assim, utilizando esse conhecimento, pretende-se investigar nas bases de dados os clientes que devem ser inspecionados.

A próxima seção apresentará os dados que os especialistas indicaram à equipe do projeto para utilizarem na solução proposta.

3.2 Dados disponíveis para análise

A medida que os especialistas disponibilizaram informações de como se poderiam identificar unidades consumidoras com suspeita de anomalia, um conjunto de informações foi sendo agrupado e considerado como importante para a análise e criação de regras para o sistema baseado em conhecimento.

Basicamente, o conjunto de dados considerado inicialmente se divide em três grupos de informações: informações cadastrais sobre o consumidor, informações históricas de consumo e faturamento do consumidor e informações sobre a última inspeção do consumidor.

As seções seguintes definem melhor estes três grupos de informações.

3.2.1 Informações cadastrais

O cadastro de informações gerais sobre cada unidade consumidora é bem diversificado. Há campos como 'categoria de cliente' e 'categoria ANEEL' que aparentam ser redundantes mas que, segundo os especialistas, são igualmente importantes. Dentre os dados existentes, o conjunto de dados selecionado para uso no sistema baseado em conhecimento foram:

a) Subestação de Distribuição do Alimentador: campo numérico discreto identificando a subestação que serve à unidade consumidora;

b) Dw_uc: campo numérico discreto identificando a unidade consumidora. É a chave de identificação da base;

c) categoria de Clientes: campo numérico discreto indicando a categoria do cliente segundo a classificação disposta no artigo 20° da resolução 456 (ANEEL, 2000);

d) município: campo numérico discreto indicando o município do cliente;

e) razão: campo numérico discreto indicando o razão do cliente. Os razões são grupamentos de consumidores que são faturados no mesmo dia do mês;

f) local: campo numérico discreto indicando o local do cliente, isto é, em uma determinada localidade, contém consumidores de todos os razões;

g) livro: campo numérico discreto indicando o livro do cliente. O livro indica um conjunto de consumidores lidos num mesmo dia por um único leiturista. Um conjunto de livros forma um razão;

h) rota: campo numérico discreto indicando a rota do cliente. O fato de um grupo de clientes possuir a mesma rota significa que estes estão próximos uns dos outros.

i) Conta: Este campo é a composição de Razão, Local, Livro e um identificador único para uma unidade consumidora;

j) região: campo numérico discreto indicando a região do cliente;

k) alimentador do posto transformador: campo numérico discreto indicando o alimentador do posto transformador do cliente;

l) bloco do posto transformador: campo numérico discreto indicando o bloco do posto transformador do cliente, ou seja, a chave que desliga aquele posto transformador;

m) número do transformador: campo numérico discreto indicando o número do transformador do cliente;

n) poste: campo numérico discreto indicando o poste do cliente;

o) classe: campo numérico discreto indicando a classe do cliente de acordo com o artigo 20º da resolução 456 da ANEEL (ANEEL, 2000), porém com identificação interna da empresa;

p) subClasse: campo numérico discreto indicando a subclasse do cliente de acordo com o artigo 20º da resolução 456 da ANEEL (ANEEL, 2000), porém com identificação interna da empresa;

q) tarifa: campo numérico discreto indicando o tipo de tarifa do cliente como disposto nos artigos 40º a 56º da resolução 456 da ANEEL (ANEEL, 2000);

r) subClasse ANEEL: campo numérico discreto indicando a subclasse ANEEL do cliente de acordo com o artigo 20º da resolução da ANEEL (ANEEL, 2000), porém com identificação equivalente à disposta na resolução;

s) classe de serviço: campo numérico discreto indicando a classe de serviço do cliente, de acordo com o artigo 2º da resolução 456 da ANEEL (ANEEL, 2000), alíneas XXII e XXIII (Grupo A, Grupo B, dentre eles A1, A2, ..., B1, B2, ...);

t) área da atividade: campo numérico discreto indicando a área da atividade do cliente de acordo com classificação interna da empresa;

u) atividade: campo numérico discreto indicando a atividade do cliente, com significado similar ao campo anterior;

v) setor econômico: campo numérico discreto indicando o setor econômico de atuação do cliente;

w) atividade CNAE: campo numérico discreto indicando a atividade CNAE do cliente, com informações de mesma natureza do setor econômico;

x) categoria ANEEL: campo numérico discreto indicando a categoria ANEEL do cliente, com informações de mesma natureza do setor econômico;

y) escritório: sigla do escritório responsável pelo atendimento à UC na região;

z) nome do município: nome do município da unidade consumidora;

aa) data de ligação da UC: data de ligação da unidade consumidora no sistema elétrico;

bb) potência nominal do transformador: característica relativa ao equipamento que atende à unidade consumidora;

cc) fator de carga do transformador: razão entre a demanda média e a demanda máxima de uma UC num mesmo intervalo de tempo (mensal) e que caracteriza o transformador que serve a unidade consumidora;

dd) demanda em KiloWatt Hora (KWh): energia medida no transformador que serve a unidade consumidora;

ee) demanda em KiloWatt (KW): média das potências elétricas ativas ou reativas, solicitadas ao sistema elétrico pela parcela da carga instalada em operação na unidade consumidora;

ff) dw_motivo_ss: identificador de motivo da solicitação de inspeção.

3.2.2 Informações históricas de faturamento

Sobre faturamento a principal informação utilizada é o histórico de consumo de cada unidade consumidora. Os especialistas indicaram que os dados de consumo dos últimos 24 meses seriam suficientes para fazer as análises. Além dos consumos, foram requisitadas também o histórico de irregularidades de leitura. Essas irregularidades de leitura são registradas no momento em que o leiturista realiza o registro do consumo a partir da leitura do medidor de cada unidade consumidora. Neste instante, ele registra irregularidades de diversos tipos, por exemplo:

- Medidor sem lacre;
- padrão avariado;
- portão de acesso fechado;
- consumo fora do esperado;
- sem leitura.

Algumas dessas irregularidades possuem informações que podem invalidar o consumo do mês em observação, e, portanto são importantes para o uso no sistema baseado em conhecimento.

Outra informação similar às irregularidades de consumo são as ocorrências de faturamento. Estas são registradas automaticamente pelo sistema e também indicam que houve alguma anomalia no mês em observação. Abaixo citam-se algumas ocorrências de faturamento a título de exemplo:

- Consumo faturado pela média trimestral;
- consumo abaixo do mínimo;
- consumo faturado pela média semestral;
- consumo refaturado;
- ajuste de consumo anterior.

As informações de ocorrências de faturamento são utilizadas de modo similar às irregularidades de leitura.

3.2.3 Informações sobre últimas inspeções

Além dos dados cadastrais e das informações de consumo e faturamento, os especialistas utilizam os dados das últimas inspeções dos clientes para selecionar ou não um cliente para inspeção.

Os dados de inspeção utilizados são basicamente: data da inspeção, identificação da unidade consumidora, motivo de geração da inspeção e resultado obtido pela inspeção. Abaixo apresentam-se mais informações sobre esses dados:

a) Data da inspeção: data em que a inspeção foi realizada. Existem outras datas relacionadas à inspeção como data de geração da inspeção - data na qual a inspeção foi solicitada - e data de digitação da inspeção – data em que o usuário cadastrou o resultado da inspeção - mas essas outras datas não foram consideradas importantes;

b) identificação da unidade consumidora: identificador do cliente que foi inspecionado;

c) resultado da inspeção: informação sobre a situação encontrada no momento da inspeção na unidade consumidora. Os resultados podem ser:

- Normal: unidade consumidora sem anomalias técnicas. Podem haver irregularidades comerciais, mas estas não são consideradas de importância para investigações de perdas comerciais;
- fraude/irregular: unidade consumidora possuía fraude em algum ponto de sua instalação ou irregularidade técnica - havia alguma anomalia técnica como, por exemplo, um equipamento defeituoso ou em estado precário de conservação;
- impedimento: não foi possível realizar a inspeção (por exemplo, devido à residência estar desabitada).

d) motivo de geração da inspeção: indica qual motivo ocasionou a geração desta inspeção. Por exemplo, a inspeção pode ter sido criada devido a uma denúncia, ou por indicação do leiturista ou ainda por outros motivos diversos.

3.3 Procedimento atual de seleção segundo heurísticas

Como apresentado na Seção 3.1, a principal forma de combate às perdas comerciais atualmente empregadas pela ESCELSA são as inspeções realizadas nas instalações dos consumidores. O procedimento mais comum são as inspeções por varredura. Neste procedimento, os especialistas escolhem uma área onde as perdas comerciais estão altas, e submetem inspeções para todas as unidades consumidoras desta área. Com isso, chega-se a realizar, segundo os especialistas, cerca de 8.000 inspeções em um único mês (média realizada em 2004). Entretanto, este é um procedimento custoso e que possui uma taxa de

identificação de fraudes e irregularidades relativamente baixa, na faixa de 12% (Cometti et al., 2005).

Uma solução melhor do que inspecionar todas as unidades de uma determinada região é realizar inspeções somente em suspeitos de irregularidades. Os especialistas neste domínio de problema aplicam atualmente algumas heurísticas, mas há limitações e dificuldades para que todo o conhecimento do assunto seja utilizado nas seleções de casos. A aplicação das heurísticas envolve o uso de um volume de dados extenso, difícil de ser realizado sem ferramentas adequadas. O próximo capítulo apresentará melhor essas dificuldades. Por esta razão os especialistas ficam limitados a poucas heurísticas que conseguem aplicar através das ferramentas computacionais que dispõem. Abaixo, citam-se duas heurísticas utilizadas pelos especialistas.

3.3.1 Consumo Zero

Os especialistas consideram que qualquer residência/unidade consumidora consome, em se tratando de energia elétrica, um certo valor mínimo por mês. Clientes com um determinado período de meses com consumo abaixo do mínimo esperado são considerados suspeitos de irregularidades. Sobre esses casos suspeitos, excluem-se os casos de residências em áreas de veraneio ou em áreas com muita incidência de imóveis desocupados. Aos casos que se enquadram neste perfil, excluídos os de área de veraneio e de imóveis desocupados, geram-se inspeções para averiguação de possíveis fraudes ou irregularidades.

3.3.2 Degrau de consumo

A regra degrau de consumo, considera que o consumo de um determinado mês não deve variar demais em relação ao consumo dos meses anteriores. Com exceção de casos nos quais ocorrem troca de inquilino ou proprietário, ou desocupação de imóvel, espera-se que o consumo não tenha uma variação brusca entre dois meses consecutivos. Assim, uma heurística simples utilizada pelos especialistas é comparar o consumo do mês com os consumos dos últimos meses anteriores, e gerar inspeções para averiguar os casos que forem suspeitos.

4 SAUIPE: Um Sistema Baseado em Conhecimento para auxiliar a identificação de Perdas Comerciais na distribuição de Energia Elétrica

Como apresentado no capítulo anterior, o combate a perdas comerciais, realizado principalmente por meio de inspeções, pode ser aprimorado através da realização de inspeções de forma criteriosa. Realizar inspeções de varredura, de modo a inspecionar todos os consumidores de uma determinada região é uma solução na qual se deixa de usar as informações contidas nas bases de dados das corporações, e deixa-se de usar o conhecimento dos especialistas em prol de uma pré-seleção dos clientes a serem inspecionados.

No caso da ESCELSA, já existe por parte dos especialistas a iniciativa de utilizar o conhecimento existente para realizar a pré-seleção dos suspeitos de irregularidades.

Uma forma de realizar esta pré-seleção é através da aplicação de heurísticas que identifiquem suspeitos através da análise dos dados dos clientes. Os especialistas, analisando os históricos de consumo e outros atributos do cliente, conseguem detectar suspeitos de irregularidades.

Entretanto, a aplicação destas heurísticas é iniciativa modesta devido às dificuldades de se realizar esta tarefa de modo manual e à ausência de ferramentas que facilitem o uso deste conhecimento. Dentre essas dificuldades, destacam-se:

- A aplicação das heurísticas envolve análises sobre um conjunto de dados extenso e exige um esforço considerável dos especialistas para a análise de cada caso. Em se tratar de uma base com centenas ou milhares de casos para se analisar, o trabalho pode tornar-se impróprio para a realização manual devido ao tempo que demandaria;
- os dados a serem avaliados muitas vezes precisam ser tratados, modificados e/ou agrupados, o que aumenta as tarefas necessárias para a realização da análise de cada caso;
- com o passar do tempo, os tipos de fraudes, ou padrões que indicam determinada fraude, variam, e há necessidade de reavaliar a heurística, alterando parâmetros, ou aprimorando as regras;

- o desenvolvimento destas regras em ferramentas computacionais não é simples, em especial devido a natureza de constante necessidade de alteração que dificulta a manutenção de uma solução computacional tradicional.

Assim, visando utilizar de forma eficiente o conhecimento dos especialistas, e mediante às características citadas acima, propõe-se como solução o desenvolvimento de um sistema baseado em conhecimento capaz de registrar e aplicar as heurísticas definidas pelos especialistas sobre as bases de clientes.

O SBC proposto e desenvolvido, denominado SAUIPE – Sistema de Auxílio à Investigação de Perdas Elétricas – foi projetado com o objetivo de possibilitar de modo simples o mapeamento do conhecimento especialista e sua aplicação de modo configurável e flexível sobre bases de casos de investigação.

Ser configurável e de fácil expansão de regras foi um dos requisitos básicos. As regras analisadas junto aos especialistas mostraram-se um tanto flexíveis e parametrizáveis, o que exige uma ferramenta dinâmica e configurável no que diz respeito a cada regra que esteja formalizada em sua base de conhecimento. Isso motivou o desenvolvimento de uma solução que não demandasse dificuldades em possíveis alterações na base de conhecimento, e que permitisse a parametrização das regras de modo fácil e prático.

Outro requisito da ferramenta é a possibilidade de aplicar essas regras sobre uma massa de dados de investigação de modo agrupado, combinando os resultados de cada regra segundo um critério definido pelo especialista. O especialista deve realizar a seleção dos consumidores segundo uma, duas ou várias regras, sendo que o resultado final da seleção possa ser a união, interseção ou outra combinação dos resultados individuais de cada regra, segundo a determinação do especialista.

O desenvolvimento deste sistema iniciou-se pela aquisição do conhecimento e estudo do domínio e das características deste problema específico. Uma vez realizado o levantamento do conhecimento especialista e os requisitos desejados para o sistema, projetou-se o sistema conforme o apresentado nas próximas seções.

4.1 Aquisição do conhecimento

Uma das etapas mais importantes da construção de um sistema baseado em conhecimento é a aquisição do conhecimento. No caso deste trabalho, esta tarefa foi realizada com uma seqüência de entrevistas junto aos especialistas da distribuidora de energia.

As entrevistas foram realizadas no ambiente de trabalho dos especialistas. Algumas foram gravadas – com a devida autorização dos envolvidos - para facilitar a posterior transcrição e análise das informações. Foram entrevistados dois especialistas no assunto. Em uma primeira etapa as entrevistas tinham um roteiro, mas de modo geral eram não-estruturadas. A medida em que se precisava atingir um número maior de detalhes, ou esclarecer dúvidas, utilizava-se entrevistas estruturadas. Além disso, utilizou-se a técnica de acompanhamento de casos para analisar casos específicos e capturar melhor o conhecimento dos especialistas.

Inicialmente as entrevistas foram realizadas com os dois especialistas juntos. Notando-se a diferença entre os perfis dos especialistas, passou-se a dar preferência para entrevistas individuais, visando facilitar a aquisição do conhecimento de cada especialista.

O primeiro especialista, mais cauteloso em suas afirmações e comentários, buscava dizer apenas o que tinha comprovação, e encontramos dificuldades de extrair o conhecimento heurístico, visto que o mesmo tinha receio em dizer algo que não fosse concretizado posteriormente em campo. Já o segundo mostrou-se mais aberto para as tentativas e apresentou suas idéias com maior abertura, possibilitando um melhor aproveitamento das entrevistas em relação a aquisição do conhecimento.

Com base nas informações dos especialistas, e com os dados disponíveis para análise, foi elaborado um conjunto de regras para identificar unidades consumidoras com suspeita de anomalia em seu comportamento – comportamento esse baseado nos dados que a distribuidora possui sobre a unidade consumidora: dados cadastrais, dados de histórico de faturamento e dados de inspeções realizadas.

Inicialmente as regras foram documentadas em texto corrido, mas logo passou-se a utilizar o formato de fluxograma pois este facilitava a validação junto ao especialista.

A partir das entrevistas e das regras documentadas, analisou-se qual seria a estrutura necessária para armazenar e utilizar este conhecimento. As próximas seções apresentarão a arquitetura e estrutura definida para atender à necessidade deste sistema.

4.2 Arquitetura do sistema

A arquitetura estabelecida para o sistema segue as linhas gerais das arquiteturas de Sistemas Baseados em Conhecimento conforme apresentado em Stefik (1995). A Figura 4.1 apresenta a arquitetura geral do SAUIPE, na qual o sistema está dividido em três camadas.

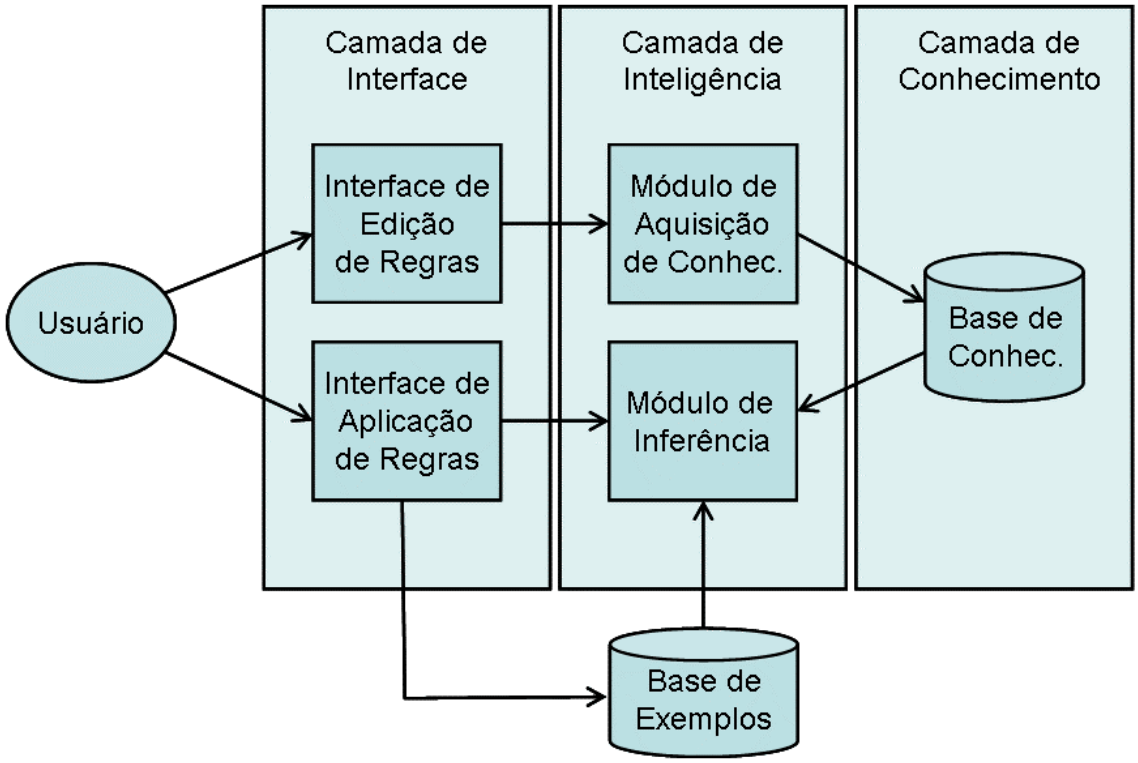

Figura 4.1 - Arquitetura do SAUIPE

A camada de conhecimento é explicitamente separada das outras camadas - característica típica dos SBCs. Na camada de conhecimento está a base de conhecimento que por se tratar de um elemento especial para o sistema baseado em conhecimento será apresentada em detalhe na próxima seção.

Além da base de conhecimento, outro importante componente do SAUIPE é a sua máquina de inferência – aqui denominada módulo de inferência. O fato de o problema envolver um domínio específico (fraudes em distribuição de energia elétrica) e o desejo de atender determinados requisitos – que serão apresentados no decorrer deste - foi um importante delimitador para direcionar o desenvolvimento de uma máquina de inferência específica. O motivo maior para a proposta de uma máquina de inferência própria é devido a representação do conhecimento adotada. A Seção 4.3 apresentará a justificativa para a escolha da representação do conhecimento adotada.

O módulo de inferência encontra-se na camada de Inteligência. A Camada de Inteligência possui dois componentes do SAUIPE: o módulo de aquisição de conhecimento e o módulo de inferência. O módulo de inferência foi projetado de modo a realizar a interpretação das regras contidas na base de conhecimento aplicando-as de modo isolado ou agrupado, conforme os parâmetros utilizados em seu acionamento. O módulo de interface de aplicação de regras também deve permitir, de forma dinâmica, que o usuário informe os valores que deseja utilizar para os parâmetros das regras. Basicamente este requisito é atendido da seguinte forma: ao ler a estrutura da regra, o módulo de interface de aplicação de regras permite que o usuário altere os valores dos parâmetros existentes nas regras. Estes valores serão utilizados na aplicação das regras sobre a base de casos em análise.

O módulo de interface de aplicação das regras também permite flexibilidade quanto a escolha e formato das bases de dados utilizadas na aplicação das regras. O módulo de inferência recebe, ao ser acionado pela interface de aplicação de regras, a informação do formato da base de dados. Basicamente, este formato refere-se às colunas de dados existentes na base de dados. Estes formatos podem ser alterados através da configuração de arquivos XML (Seção 4.4 deste trabalho).

Em resumo, antes da máquina de inferência ser acionada, o usuário pode selecionar quais regras da base de conhecimento deseja utilizar, quais parâmetros estas devem utilizar e qual o modo de junção dos resultados em um resultado final deseja utilizar (vide Seção 4.5) . Além disso o usuário seleciona uma base de casos de análise e indica o seu formato. Quando o módulo de inferência é acionado, este realiza basicamente as seguintes tarefas:

- para cada caso c' da base de casos/exemplos:
    - para cada regra x' da base de conhecimento selecionada para ser utilizada:
        - avalie o caso c' com a regra x' e adicione o resultado y' obtido à lista de resultados.
    - avalie a lista de resultados segundo o modo de junção escolhido, e defina o resultado final para c' a partir de todos os y' deste caso.
    - registre o resultado final para c' .

O outro componente do Módulo de Inteligência é o módulo de aquisição de conhecimento. Esse módulo trabalha em conjunto com a Interface de Edição de Regras. Esses dois módulos juntos possibilitam ao usuário especificar novas regras assim como alterar as regras existentes.

4.2.1 A base de conhecimento

As regras estabelecidas a partir do conhecimento dos especialistas foram analisadas de modo a buscar uma melhor forma de representação, buscando atender a alguns requisitos:

- Uso de uma forma de representação simples e de fácil compreensão;
- uso de uma representação que permitisse alteração com facilidade;
- disponibilização do conhecimento em formato aberto, público e facilmente acessível.

Além destes requisitos, observou-se que as regras especificadas pelos especialistas seguiam em geral a estrutura "se ... então ... senão", ou, quando não estavam neste formato explicitamente, poderiam ser facilmente transformadas para este formato.

Observando esses requisitos básicos, e analisando as regras especificadas pelos especialistas (conhecimento específico para análise dos casos em investigação), optou-se por utilizar regras de produção como metodologia de representação do conhecimento.

As regras foram representadas através do encadeamento de condições no formato "se <...> então <...> senão <...>". Essa escolha permitiu gerar uma base de conhecimento de fácil leitura e manipulação. Para representação deste conhecimento, foi definida uma linguagem específica denominada XmlRuleLang que será apresentada na Seção 4.3. As regras capturadas na fase de aquisição do conhecimento, representadas na linguagem XmlRuleLang formam a base de conhecimento do sistema. Estas regras serão apresentadas, de forma simplificada, nas próximas sub-seções.

4.2.1.1 Consumo Zero

Como apresentado anteriormente, os especialistas consideram que qualquer residência/unidade consumidora consome, em se tratando de energia elétrica, um certo valor mínimo por mês. A partir deste fato, clientes com um longo período de meses com consumo abaixo do mínimo esperado são considerados suspeitos de irregularidades. Sobre esses casos suspeitos, excluem-se os casos de residências em áreas de veraneio ou em áreas com muita incidência de imóveis desocupados. Exclui-se também consumidores que foram inspecionados recentemente e obtiveram resultado 'normal'. Aos casos que se enquadram neste perfil, excluídos os de área de veraneio, de imóveis desocupados e os recentemente inspecionados com resultado normal, geram-se inspeções para averiguação de possíveis fraudes ou irregularidades.

4.2.1.2 Degrau de consumo

Como apresentado anteriormente, a regra degrau de consumo, considera que o consumo de um determinado mês não deve variar demais em relação aos meses anteriores. Com exceção de casos nos quais ocorrem troca de inquilino ou proprietário, ou desocupação de imóvel, espera-se que o consumo não tenha uma variação brusca entre um intervalo de apenas dois meses consecutivos. Assim, essa heurística compara o consumo do mês com os consumos dos meses anteriores, e gera inspeções para averiguar os casos que forem suspeitos. Essa comparação é feita utilizando o desvio padrão. Os especialistas definiram que caso o consumo diminua em relação a média dos últimos 24 meses o valor referente a duas vezes o desvio deste mesmo período, então, o cliente é suspeito de fraude.

4.2.1.3 Média Por Rota

A regra "Média por rota" trata-se de uma heurística que assume que os consumidores que residem em uma mesma rota, ou seja, consumidores que residem em uma certa proximidade, devem possuir um consumo um tanto similar.

O fato de um grupo de clientes possuir a mesma rota significa que estes estão localizados fisicamente próximos uns dos outros. De modo genérico, considera-se que os consumos em uma pequena área de vizinhança sejam similares.

Assim, esta regra busca por consumidores cujo consumo está um certo percentual abaixo da média de consumo dos outros consumidores de sua rota. Além disso, a regra verifica se este consumo baixo não é devido a uma irregularidade de leitura ou irregularidade de faturamento, o que justificaria um consumo fora do esperado.

Naturalmente sabe-se que há um número grande de exceções a esta regra. Essas exceções devem-se ao fato que uma determinada área de vizinhança pode conter consumidores de diferentes portes. Mesmo com essas exceções, considera-se interessante realizar inspeções a estes casos, pois dentre eles podem haver fraudes e irregularidades.

4.2.1.4 Média dos últimos n meses

Esta regra baseia-se no fato de que os consumos de um determinado consumidor não oscilam de forma brusca quando comparados em um curto período de meses. Assim, esta regra avalia se o consumo do último mês reduziu um certo percentual em relação ao consumo dos últimos três meses. Caso tenha reduzido, verificam-se irregularidades de leitura ou ocorrências de faturamento no período. Verifica-se também se foi realizada alguma inspeção a este consumidor. Caso tenha tido inspeção com resultado normal, ou caso tenha ocorrido

alguma irregularidade de leitura ou ocorrência de faturamento, então o consumidor é desconsiderado, do contrário ele é indicado para inspeção.

4.2.1.5 Média por atividade

Esta heurística assume que os consumidores que possuem a mesma atividade econômica, devem possuir um consumo um tanto similar. Por exemplo, espera-se que os consumidores que estiverem cadastrados como "posto de gasolina" devem possuir consumos relativamente similares.

Assim, esta regra busca por consumidores cujo consumo está um certo percentual abaixo da média de consumo de sua atividade. Além disso, a regra verifica se este consumo baixo não é devido a uma irregularidade de leitura ou irregularidade de faturamento, o que justificaria um consumo fora do esperado.

4.2.1.6 Irregularidade de leitura

Essa regra se propõe a investigar se o consumidor possui determinadas irregularidades de leituras que são indícios de possíveis fraudes. Além disso, verifica-se se houveram inspeções recentes para o consumidor. Outra condição é a de o consumidor fazer parte de determinados grupos ou subgrupos de clientes. Por exemplo, pode-se com essa regra determinar que ela considere suspeito todos os consumidores do grupo de "consumidores da área rural", que possuem irregularidade de leitura indicando "lacre violado" e não foram inspecionados nos últimos 9 meses.

4.2.1.7 Análise do fator de carga por atividade

Esta heurística baseia-se no fato que de regra geral, o fator de carga dos consumidores de uma determinada atividade não devem ser muito diferentes uns dos outros. Assim, consideram-se suspeitos de fraudes os consumidores que possuem o fator de carga muito abaixo ou muito acima da média dos fatores de carga de sua atividade. Além disso, verifica-se se o mês atual é mês de férias. Nos meses de férias o fator de carga pode variar muito. Caso seja mês de férias, nenhum consumidor é considerado suspeito nesta regra (há um parâmetro na regra que indica quais meses são considerados meses de férias).

4.3 A linguagem proposta para representar o conhecimento

A base de conhecimento do sistema proposto, constitui-se de uma base de regras definidas na linguagem XmlRuleLang. Essa linguagem, que utiliza XML[2] (eXtensible Markup Language) como estrutura, foi definida para atender à necessidade de registro das regras de uma forma simples e alterável com facilidade pelo usuário final, como explicado na Seção 4.2.1.

A utilização de XML como estrutura da linguagem foi importante, pois:

- XML é uma linguagem amplamente utilizada e de fácil entendimento de sua estrutura e conteúdo - isso facilita o entendimento das regras caso seja necessário alterar ou analisar uma regra diretamente em sua definição;
- XML é uma linguagem flexível, que permite definir tags específicas para as funcionalidades desejadas; sua flexibilidade permite atribuir a esses tags propriedades e valores de acordo com a necessidade específica;
- usar XML facilita integração com outras ferramentas que possam vir a utilizar esta base de conhecimento. O uso, por exemplo, de uma ferramenta que realize a edição de XML pode ser feito para a atualização de alguma regra, sem necessidade de um software específico. Caso se deseje apresentar o conteúdo das regras em uma interface web, por exemplo, poder-se-ia extrair as regras diretamente do XML, garantindo-se assim que o conhecimento publicado não estaria desatualizado pois poderia refletir a situação das regras que estão em uso no momento;
- usar XML para definição das regras permite adicionar regras novas sem necessidade de recompilar código algum, alterando apenas os documentos que definem as regras;
- pode-se usar o DTD (Document Type Definition) para validar a regra escrita, sem haver necessidade de um software específico para este fim.

O uso de XML deixa a arquitetura aberta, facilitando o reaproveitamento das regras ou o entendimento das mesmas por outros profissionais que venham a utilizar essa base de conhecimento.

Como apresentado na Seção 2.1.4, há alternativas para a representação do conhecimento similares a solução adotada. Entre elas cita-se a apresentada por Lee & Sohn (2003) –a linguagem XRML – e a linguagem RuleML defendida pelo projeto RuleML (2006)

[2] Cf. nota 1 contida no Capítulo 2 deste trabalho

– ambas apresentadas na Seção 2.1.4 deste trabalho. Entretanto, alguns fatores influenciaram a decisão de definir uma linguagem específica para este trabalho:

- a estrutura e os tags propostos por estas linguagem (XRML, RuleML, entre outras) visam um escopo mais abrangente que o deste trabalho. Elas permitem, por exemplo, não apenas representar o conhecimento em regras de produção, mas também em representação lógica – formatada em estrutura XML. Esse fator não impediria seu uso, mas implica em duas conseqüências:
  - A estrutura da linguagem e seus tags são mais complexos de serem entendidos – isso dificultaria a manipulação da base de dados (quer seja pelo especialista, quer seja por uma interface de edição de regras ou mesmo pela máquina de inferência);
  - A máquina de inferência para estas linguagens precisa ser mais robusta em relação a necessidade deste domínio em estudo, o que demandaria um esforço computacional desnecessário.
- os tags definidos nestas linguagens estão em língua estrangeira, dificultando a manipulação e entendimento destes por parte de especialistas, assim como o compartilhamento desta base de conhecimento com outros sistemas corporativos;
- não depender de linguagens em desenvolvimento;
- ter controle sobre a linguagem, possibilitando facilmente realizar extensões, se necessário;
- estes trabalhos são recentes, ainda em evolução e definição, implicando em:
  - existência de poucas alternativas de máquina de inferência implementadas, sendo que estas também encontram-se em aprimoramento;
  - risco de elaborar uma base de conhecimento em uma linguagem de representação ainda não estável, que rapidamente pode vir a ser considerada ultrapassada - dificultando possíveis evoluções (isso poderia, por exemplo, dificultar a atualização futura da máquina de inferência).

Uma outra alternativa para a representação do conhecimento seria o uso da ferramenta CLIPS (ou outra similar), que possui uma linguagem de representação própria e uma máquina de inferência interna à ferramenta. Entretanto, o fato de ter-se o objetivo de uma solução simples, flexível e de fácil manipulação da base de conhecimento, influenciou o uso de uma linguagem própria, assim como o uso de um mecanismo de inferência próprio. Isso permite que todo o sistema seja moldado segundo as necessidades específicas requeridas pelo usuário, além de permitir que a solução possua plena interoperabilidade em diferentes

plataformas. O uso de uma solução como CLIPS dificultaria o compartilhamento da base de conhecimento e dificultaria também a edição das regras devido a maior complexidade da linguagem de representação quando comparado a representação proposta.

Por estes motivos, foi definida a linguagem XmlRuleLang. Para apresentar a linguagem XmlRuleLang, inicialmente será apresentada uma regra simplificada. Isso objetiva a compreensão da estrutura da linguagem. Em seguida será apresentada uma regra real definida pelos especialistas.

4.3.1 Conhecendo a linguagem por uma regra simplificada

Essa seção apresenta e explica a regra "Avaliação por média" – uma simplificação da regra "média dos últimos 3 meses"- com o propósito de apresentar a linguagem definida.

A Figura 4.2 apresenta o fluxograma da regra a ser utilizada para facilitar a compreensão do leitor. Esta regra verifica se o último consumo é inferior a uma certa proporção da média dos últimos 3 consumos. Caso seja maior ou igual, o consumidor é considerado um caso normal. Caso seja menor, a regra verifica se houve irregularidade de leitura no mês do último consumo (pode ter ocorrido algum problema com a leitura deste consumidor, e neste caso, se houver, desconsidera-se o consumidor). Se houve irregularidade de leitura, o consumidor é classificado como normal, do contrário é classificado como suspeito.

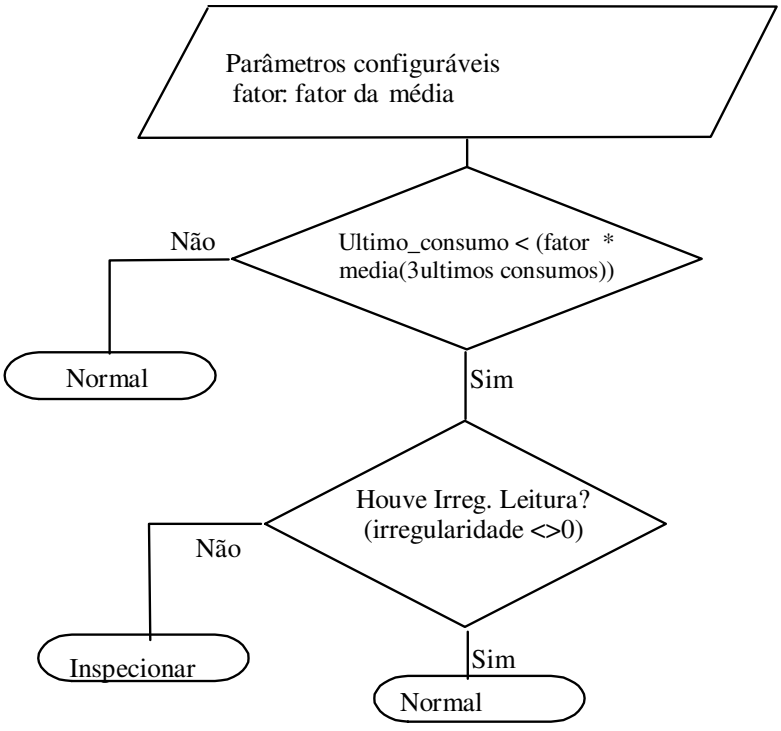

Figura 4.2 - Fluxograma da regra avaliação por média

Abaixo segue esta regra, representada na linguagem XmlRuleLang, sendo que a explicação de cada trecho será em seguida apresentada:

```
        <regra nome="Avaliação por média" inicio="Primeira">
        <descricao> Busca por clientes consumo muito abaixo da média dos
últimos meses</descricao>
        <propriedades>
        <propriedade nome="fator" descricao="Fator da média"
               tipo="numerico" visualizavel="true" default="0.6">
0.6</propriedade>
        <propriedade nome="x" descricao="Exemplo X"
               tipo="numerico" visualizavel="true" default="6">
6</propriedade>
        </propriedades>
        <cond nome="Primeira">
           <se>
             <formula nome="menor">
               <base atributo="consumo24"/>
               <formula nome="multiplicacao">
                 <propriedade nome="fator"/>
                 <formula nome="media">
                   <lstAtributos atributos="consumo21,consumo22,consumo23"/>
                 </formula>
               </formula>
             </formula>
           </se>
           <entao>
              <testacond nome=" SegundaCondição "/>   </entao>
           <senao>
              <resultado valor="0">normal</resultado> </senao>
         </cond>
        <cond nome="SegundaCondição">
          <se>
             <formula nome="igual">
               <base atributo="irreg24"/>
               <const valor="0"/>  </formula>  </se>
          <entao>
               <resultado valor="1">Inspecionar</resultado> </entao>
          <senao>
               <resultado valor="0">normal</resultado> </senao>
          </cond>

        </regra>
```

A primeira seção da regra, apresentada abaixo, possui dois tags XML. O primeiro é o tag "regra". Este tag determina o nome da regra e possui uma propriedade que determina o nome da primeira condição que será avaliada no corpo da regra.

```
        <regra nome="Avaliação por média" inicio="Primeira">
        <descricao>Busca por clientes com consumo muito abaixo da média dos
últimos meses</descricao>
```

Além do tag "regra", há o tag "descrição" que apresenta uma descrição mais detalhada da regra.

O próximo tag definido na linguagem é um tag opcional, denominado "propriedades". Este tag permite definir quais os parâmetros e quais variáveis existirão na regra. Os parâmetros são variáveis que podem ser alteradas antes da execução das regras pelo usuário do sistema. As variáveis são propriedades da regra que não são alteradas pelo usuário do sistema, mas que são utilizadas nas condicionais do corpo da regra com o intuito de fazer algum tipo de controle, como, por exemplo, controlar a avaliação ou não de uma condição.

```
        <propriedades>
           <propriedade nome="fator" descricao="Fator da média"
               tipo="numerico" visualizavel="true" default="0.6">
0.6</propriedade>
        <propriedade nome="x" descricao="Exemplo X"
               tipo="numerico" visualizavel="true" default="5">
5</propriedade>
        </propriedades>
```

No bloco de propriedades podem ser definidas diversas propriedades. Cada propriedade estará definida em uma tag </propriedade>. Cada propriedade possui um nome, uma descrição, um tipo, um indicador de visualizável ou não e um valor default.

O "nome" é a referencia para uso dessa propriedade no corpo da regra. As propriedades possuem também um tipo que pode ser:

- 'numerico': quando a propriedade representa um número ou uma lista de números;
- 'flag': quando a propriedade representa uma opção configurável;
- 'nominal': quando a propriedade representa uma palavra ou uma lista de palavras.

O indicador de visualizável ou não indica se a propriedade será ou não exibida nos diálogos de configuração da regra. Propriedades que servem como variáveis para a regra, e que não devem ser alteradas pelo usuário do sistema devem possuir esse indicador ajustado para "false".

O valor default é o valor padrão para a propriedade. Após realizadas alterações no valor da propriedade, este valor serve de referência para caso o usuário deseje retornar o valor para o valor inicialmente padronizado para a propriedade.

O restante do XML – listado abaixo – é o corpo da regra, trecho no qual podem existir um ou mais condicionais. Cada condicional é composto por um bloco "se"..."então"..."senão". Esses blocos também são tags no documento XML. A tag "se" estabelece a condição que será testada. Esta condição pode ser uma expressão lógica, um teste de variáveis ou propriedades, ou uma fórmula/função que retorne um valor 'booleano'. Caso a

condição seja atendida, o bloco do tag "então" será executado. Do contrário o bloco "senão" será executado. Abaixo está a primeira condicional da regra em análise.

```
<cond nome="Primeira">
   <se>
     <formula nome="menor">
       <base atributo="consumo24"/>
       <formula nome="multiplicacao">
         <propriedade nome="fator"/>
         <formula nome="media">
           <lstAtributos atributos="consumo21,consumo22,consumo23"/>
         </formula>
       </formula>
     </formula>
   </se>
   <entao>
      <testacond nome=" SegundaCondição "/>   </entao>
   <senao>
      <resultado valor="0">normal</resultado> </senao>
```

As fórmulas – denominação dada às funções e operadores da linguagem - serão apresentadas na Seção 4.3.3. No caso da regra em análise, observa-se que foram utilizadas diversas fórmulas/funções: "menor", "multiplicacao" e "média". O tag "então" pode definir o resultado da avaliação da regra, usando para isso o tag "resultado", ou pode direcionar o processo de avaliação da regra para o próximo condicional a ser avaliado (através do tag "testacond"). Desta forma, consegue-se encadear uma seqüência de condicionais.

Ao se tratar o primeiro condicional apresentado acima, caso a avaliação do condicional retorne o valor "verdadeiro", o tag "então" será avaliado, direcionando deste modo para o segundo condicional. Caso contrário, o tag "senão" é executado determinando o resultado "normal" para esta regra.

O segundo condicional, apresentado abaixo, avalia através de um atributo da base de entrada se houve irregularidade de leitura na unidade consumidora no mês do último consumo registrado.

```
</cond>
<cond nome="SegundaCondição">
  <se>
     <formula nome="igual">
       <base atributo="irreg24"/>
       <const valor="0"/>  </formula>  </se>
  <entao>
       <resultado valor="1">Inspecionar</resultado> </entao>
  <senao>
       <resultado valor="0">normal</resultado> </senao>
  </cond>
```

Caso não tenha havido irregularidade de leitura, o atributo estará preenchido com "0" e, portanto, a fórmula "igual" retornará verdadeiro. Com este retorno, o resultado será estabelecido pelo tag "então" – sendo assim esse caso seria indicado para inspeção. Caso

tenha havido irregularidade de leitura, o bloco acionado será o do tag "senao" e assim o resultado seria "normal".

Essa regra foi apenas uma simplificação de uma regra verdadeira para facilitar o entendimento da representação do conhecimento realizada através da linguagem. A próxima seção apresenta uma regra real e sua representação completa na linguagem XmlRuleLang.

4.3.2 Conhecendo uma regra real na linguagem definida

Para apresentar um exemplo real da utilização do XML na linguagem proposta, utilizaremos a regra "média por rota". Esta é uma das regras formalizadas a partir das entrevistas com os especialistas.

Como apresentado anteriormente, a regra “Média por rota” trata-se de uma heurística que considera os consumidores de uma determinada região como possuidores de um consumo similar e portanto, aqueles que estiverem muito abaixo da média da rota – média de sua vizinhança - devem ser inspecionados. Assim, esta regra busca por consumidores cujo consumo está um certo percentual abaixo da média de consumo dos outros consumidores de sua rota. Além disso, a regra verifica se esse consumo baixo não é devido a uma irregularidade de leitura ou ocorrência de faturamento.

Para facilitar a compreensão e validação da regra, após a descrição textual obtida a partir das entrevistas, optou-se por fazer uma representação da regra através de fluxograma para facilitar a validação e compreensão da regra. A Figura 4.3 apresenta o fluxograma da regra "Média por rota".

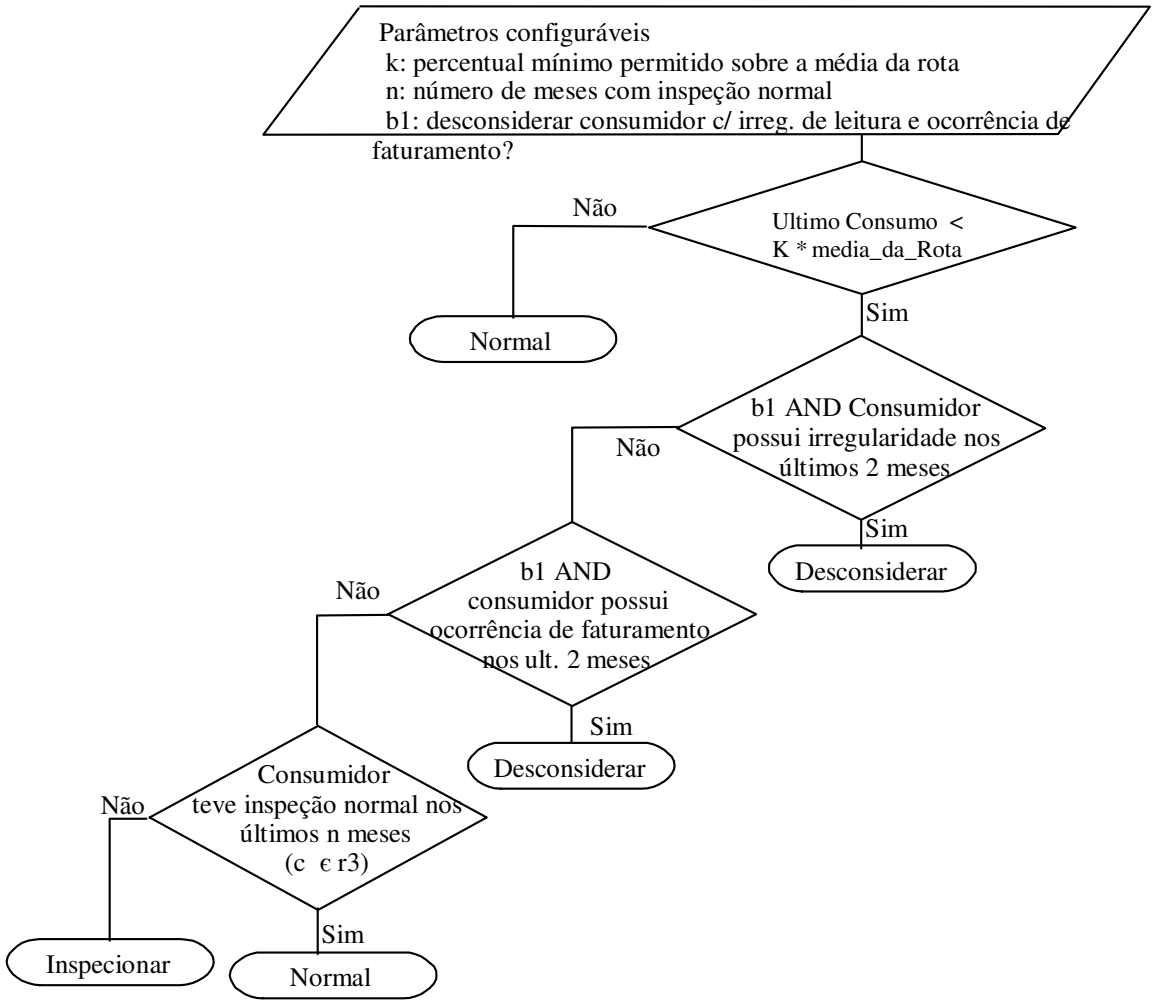

Figura 4.3 - Fluxograma da regra média por rota

O sistema proposto possui arquitetura que incorpora as novas regras ou modificações nas regras existentes sem necessidade de recompilação. A base de conhecimento é realmente independente, e as parametrizações que o sistema permite deixam-no flexível até mesmo em relação ao formato da base de casos que será usada na aplicação das regras – essa flexibilidade quanto ao formato da base de casos será apresentado na Seção 4.4.

A seguir segue a representação desta regra “Media por Rota” na linguagem definida:

```
        <regra nome="Media por rota" inicio="Primeira" sem_resultado="2">
         <descricao>Busca por clientes com ultimo consumo inferior a um
percentual da media dos consumos por rota</descricao>
```

```
         <propriedades>
            <propriedade nome="k" descricao="Percentual minimo"
tipo="numerico" visualizavel="true" default="0.75">0.75</propriedade>
            <propriedade nome="n" descricao="Meses com insp normal"
tipo="numerico" visualizavel="true" default="12">12</propriedade>
            <propriedade nome="b1" descricao="Descartar consum. com irreg e
ocor" tipo="flag" visualizavel="true" default="1">1</propriedade>
         </propriedades>
         <cond nome="Primeira">
            <se>
               <formula nome="menor">
                  <base atributo="consumo24"/>
                  <formula nome="multiplicacao">
                     <propriedade nome="k"/>
                     <base atributo="media_rota"/>
                  </formula>
               </formula>
            </se>
            <entao>
               <testacond nome="Segunda"/>
            </entao>
            <senao>
               <resultado valor="0">normal</resultado>
            </senao>
         </cond>
         <cond nome="Segunda">
            <se>
               <expressao nome="and">
                  <propriedade nome="b1"/>
                  <expressao nome="or">
                     <formula nome="diferente">
                        <base atributo="irreg23"/>
                        <const valor="0"/>
                     </formula>
                     <formula nome="diferente">
                        <base atributo="irreg24"/>
                        <const valor="0"/>
                     </formula>
                  </expressao>
               </expressao>
            </se>
            <entao>
               <resultado valor="2">sem resultado</resultado>
            </entao>
            <senao>
               <testacond nome="Terceira" />
            </senao>
         </cond>
         <cond nome="Terceira">
            <se>
               <expressao nome="and">
                  <propriedade nome="b1" />
                  <expressao nome="or">
                     <formula nome="diferente">
                        <base atributo="ocor_fat23" />
                        <const valor="0" />
                     </formula>
                     <formula nome="diferente">
                        <base atributo="ocor_fat24" />
                        <const valor="0" />
                     </formula>
```

```
            </expressao>
         </expressao>
      </se>
      <entao>
         <resultado valor="2">sem resultado</resultado>
      </entao>
      <senao>
         <testacond nome="Quarta" />
      </senao>
   </cond>
   <cond nome="Quarta">
      <se>
         <expressao nome="and">
            <formula nome="maior">
               <base atributo="ano_mes_inspecao" />
               <formula nome="subtraiMes">
                  <base atributo="ano_mes_ultimo_consumo" />
                  <propriedade nome="n" />
               </formula>
            </formula>
            <formula nome="menor_ou_igual">
               <base atributo="ano_mes_inspecao" />
               <base atributo="ano_mes_ultimo_consumo" />
            </formula>
         </expressao>
      </se>
      <entao>
         <testacond nome="Quinta" />
      </entao>
      <senao>
         <resultado valor="1">inspecionar</resultado>
      </senao>
   </cond>
   <cond nome="Quinta">
      <se>
         <expressao nome="or">
            <formula nome="igual">
               <base atributo="resultado" />
               <const valor="1" />
            </formula>
            <formula nome="igual">
               <base atributo="resultado" />
               <const valor="4" />
            </formula>
            <formula nome="igual">
               <base atributo="resultado" />
               <const valor="5" />
            </formula>
         </expressao>
      </se>
      <entao>
         <resultado valor="0">normal</resultado>
      </entao>
      <senao>
         <resultado valor="1">inspecionar</resultado>
      </senao>
   </cond>
</regra>
```

Uma característica importante de ser notada é que o bloco de propriedades permite que as regras fiquem flexíveis que em tempo de execução do sistema. Antes que o usuário inicie a avaliação de uma base de casos, ele mesmo pode alterar os parâmetros através de diálogos de configuração de cada regra. O usuário não necessita alterar a regra nos casos em que a parametrização seja suficiente para atender a flexibilidade e aplicação desejada.

4.3.3 Expressões, fórmulas e outros recursos da linguagem definida

A linguagem XmlRuleLang foi definida basicamente a partir de duas premissas:

- Atender às necessidades das regras estabelecidas junto aos especialistas;
- disponibilizar um conjunto de operações possíveis que venha a permitir flexibilidade para que novas regras sejam inseridas sem necessidade de alteração da linguagem.

A partir destas premissas, analisou-se o que as regras existentes demandavam de recursos e avaliou-se, além destes, quais outros recursos poderiam vir a serem interessantes de estarem disponíveis. As seções que se seguem apresentarão mais detalhes sobre os recursos definidos na linguagem para atender ao problema em questão.

4.3.4 Inicialização e atribuição de variáveis

Além das características já apresentadas, a linguagem permite realizar operações de inicialização e atribuição de variáveis. O bloco apresentado abaixo mostra como pode ser feita uma inicialização de variáveis. Esse tipo de inicialização é útil, por exemplo, quando se deseja ter uma variável inicializada com um valor de cada instância da base de casos. Nesse caso, esse recurso de inicialização permite, por exemplo, que se inicie uma variável com um valor de um atributo da base de dados.

```
<inicializa>
      <formula nome="atribui">
            <variavel nome="x"/>
            <base atributo="mesesEmAtraso"/>
      </formula>
      <formula nome="atribui">
            <variavel nome="cont"/>
            <formula nome="soma">
                  <propriedade nome="m"/>
                  <propriedade nome="n"/>
            </formula>
      </formula>
</inicializa>
```

Esse bloco de inicialização apresentado acima deve ser definido entre o bloco de propriedades e o primeiro bloco condicional – antes do corpo da regra. Nota-se ainda que o bloco utilizou uma fórmula/função denominada "atribui" que realiza a atribuição de valores às

variáveis, sendo que essas variáveis devem ter sido definidas na seção de propriedades. A atribuição realizada pode ser de uma constante, de um campo da base de casos ou ainda um resultado de uma fórmula.

4.3.5 Argumentos de fórmulas/funções e dados básicos da linguagem

O número de argumentos de uma fórmula varia de fórmula para fórmula. Os dados que são usados como argumentos das fórmulas e expressões podem basicamente ser de três tipos:

- Atributo: indica o nome de um atributo da base de dados de entrada (base de casos de análise ou de exemplos);

```
<base atributo="consumo24"/>
```

- constante: Valor constante definido na regra;

```
<const valor="30"/>
```

- propriedade: Valor de uma propriedade da regra. Esta propriedade pode ser um parâmetro configurável pelo usuário no momento em que ele vai aplicar a regra ou pode ser uma variável não alterável pelo usuário.

```
<propriedade nome="k"/>
```

Utilizando-se destes argumentos foram definidos os seguintes grupos de operadores:

- Expressões ("and", "or", "not"): operadores para realização de operações lógicas sobre expressões ou fórmulas que retornem valores booleanos;
- fórmulas relacionais ("maior", "menor", "igual", "pertence", entre outras): são em geral operadores de 'comparação' que a partir da avaliação de seus argumentos retornam um valor booleano. Em geral recebem dois argumentos, mas podem receber mais argumentos;
- fórmulas numéricas ("soma", "diferença", "div", "mod", "incrementa", "absoluto" entre outras): são fórmulas/funções que realizam operações sobre seus argumentos de modo a retornar valores numéricos. Em geral estas fórmulas executam operações matemáticas, mas podem executar operações sobre conjuntos como, por exemplo, retornar o número de elementos de um conjunto.

A relação completa destes operadores, assim como exemplos de sua aplicação, encontra-se no Anexo A. As principais fórmulas são apresentados na próxima seção.

4.3.6 Principais fórmulas/funções (primitivas da linguagem)

Dentre as funções/fórmulas definidas na linguagem, cita-se aqui as principais:

- **media**: calcula a média de uma lista de valores, que pode ser uma propriedade (que contenha uma lista de valores), uma constante (que contenha uma lista de valores) ou uma lista de atributos da base. Exemplo:

```
<formula nome="media">
  <lstAtributos atributos=”consumo23,consumo22,
consumo21”/>
</formula>
```

- **desvio_padrao**: Calcula o desvio padrão de uma lista de valores, que pode ser uma propriedade (que contenha uma lista de valores), uma constante (que contenha uma lista de valores) ou uma lista de atributos da base. Exemplo:

```
<formula nome="desvio_padrao">
  <lstAtributos atributos=”consumo23,consumo22,
consumo21”/>
</formula>
```

- **intervalo**: função que verifica se um elemento está em um intervalo. Retorna verdadeiro quando o primeiro elemento é menor ou igual ao segundo elemento e o segundo elemento é menor ou igual ao terceiro. Esta fórmula deve ser aplicada a três elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="intervalo">
      <const valor=”0”/>
      <base atributo=”consumo24”/>
      <const valor=”30”/>
</formula>
```

- **pertence**: fórmula que retorna verdadeiro quando o primeiro argumento pertence ao conjunto de valores definido no segundo argumento. Esta fórmula deve ser aplicada a dois argumentos numéricos. O primeiro pode ser um elemento básico (propriedade, atributo da base e constante) ou uma fórmula numérica. O segundo elemento deve ser uma lista de valores (propriedade, lista de atributos ou lista de constantes). Exemplo:

```
<formula nome="pertence">
      <base atributo=”mes_ultimo_consumo”/>
      <propriedade nome=”MesesDeFerias”/>
</formula>
```

- **pertenceString**: fórmula semelhante à “pertence”, mas que realiza a comparação considerando os valores como seqüência de caracteres (string). Esta fórmula deve ser aplicada a dois elementos. O primeiro pode ser elemento básico (propriedade, atributo da base e constante) ou fórmula numérica. O segundo elemento deve ser

uma lista de valores (propriedade, lista de atributos ou lista de constantes). Exemplo:

```
<formula nome="pertenceString">
      <base atributo="mes_ultimo_consumo"/>
      <const valor="JAN,FEV,NULL"/>
</formula>
```

- **subtraiMes**: calcula a subtração, em meses, de um argumento, que pode ser um elemento básico (propriedade, atributo da base e constante) no formato 'YYYYMM'. Exemplo:

```
<formula nome="subtraiMes">
      <base atributo="ano_mes_ultimo_consumo"/>
      <propriedade nome="n"/>
</formula>
```

Essas são as principais fórmulas/funções definidas na linguagem a fim de atender às necessidades específicas do tipo de regra e de manipulação de dados utilizados pelos especialistas entrevistados. O conjunto completo das fórmulas definidas na linguagem encontra-se listado no Apêndice A.

4.4 Flexibilidade quanto à entrada de dados e cálculos não previstos

A entrada de dados para avaliação pelo sistema é realizada por arquivo no formato CSV[3]. A escolha deste formato deve-se ao desejo de possibilitar flexibilidade quanto aos dados a serem recebidos.

O formato do arquivo de entrada é definido através de um arquivo de configuração, possibilitando assim a flexibilidade desejada. Esse arquivo de configuração utiliza XML como estrutura, e é de fácil manipulação e atualização. Esse arquivo contem as seqüências de campos que compõem a base de casos.

No momento em que se deseja aplicar as regras, escolhe-se o formato da base. Os formatos disponíveis estão no arquivo de configuração do sistema. Novos formatos podem ser adicionados alterando esse arquivo. Apresenta-se abaixo um trecho do tag "formatosbase" que se encontra no arquivo de configuração:

```
<formatosBase>
   <formatoBase nome="Formato Base Padrao1">
      <arqxml>conf/formatoBasePadrao1.xml</arqxml>
   </formatoBase>
   <formatoBase nome="Formato Analise Atividade Especifica">
      <arqxml>formatoAnaliseAtividadeEspecifica.xml</arqxml>
   </formatoBase>
```

[3] 'CSV' é um formato público, de arquivo texto com campos separados por vírgula. Em geral possui um cabeçalho das colunas na primeira linha. Cada linha seguinte do arquivo representa um registro.

```
</formatosBase>
```

Para cada formato previsto, há um arquivo XML que determina a ordem dos atributos e quais atributos que se encontram na base de casos. O usuário pode alterar esses formatos, assim como pode incluir novos formatos. Em seguida, apresenta-se um exemplo de um arquivo de configuração de base:

```
<atributo nome="uc" tipo="numeric"/>
<atributo nome="consumo1" tipo="numeric"/>
<atributo nome="classe" tipo="numeric"/>
```

Em caso de necessidade de uso de fórmulas/funções não definidas na linguagem, a primeira solução a ser tentada é a construção de uma estrutura na própria linguagem XmlRuleLang a fim de atender a nova necessidade. O cálculo de uma média, por exemplo, poderia ser realizado através da própria linguagem sem a necessidade da existência da fórmula "média" ou de um cálculo externo.

Caso a linguagem não seja suficiente para tal construção, a arquitetura proposta utiliza a flexibilidade do formato de entrada dos dados. Essa flexibilidade permite que uma nova coluna seja incorporada à base de casos de análise. Deste modo, em caso de necessidade de cálculo não previsto na linguagem, tal cálculo pode ser realizado em tempo de preparação da base de dados, gerando uma nova coluna na base.

Essa nova coluna (novo dado) passa a ser utilizado na linguagem como um atributo da base. O usuário do sistema necessita apenas cadastrar adequadamente o formato da base informando essa nova coluna, e posteriormente pode utilizá-la nas regras desejadas, ou ainda definir novas regras que utilizem-na.

4.5 Definição do resultado final a partir dos resultados de cada regra

A proposta do Sauipe prevê a existência de várias regras na base de conhecimento. Estas regras podem ser aplicadas isoladamente ou em conjunto. Quando aplicadas em conjunto, um consumidor da base de casos avaliado pelo sistema pode ter resultados diferentes nas diversas regras aplicadas. Um mesmo consumidor pode ser classificado por uma regra como "suspeito" – classificação dada a um suspeito de ser fraudador ou suspeito de possuir anomalias em suas instalações - mas ser considerado "normal" por uma outra regra.

Para resolver este tipo de ocorrência, o SAUIPE prevê diferentes modos de avaliação deste conjunto de resultados. Cada modo de avaliação trata o conjunto de resultados de uma forma diferente, gerando um único resultado. Os modos de avaliação previstos para o Sauipe são:

- União: este modo de avaliação considera o consumidor "suspeito" se ao menos um dos resultados das regras indicar esta classificação;
- interseção: considera o resultado final "suspeito" apenas se todas as regras classificarem o consumidor desta forma;
- voto: este modo de avaliação considera o resultado como 'suspeito' se dentre os resultados de todas as regras, a maioria obtiver esta classificação. Caso o número de resultados das duas classificações seja igual, então o consumidor é classificado como "normal". Se existem por exemplo 7 regras sendo aplicadas, é necessário que pelo menos 4 regras indiquem resultado 'suspeito' para que o resultado final seja 'suspeito';
- voto 2/3: comporta-se de modo similar ao método de avaliação por voto, todavia, não basta a maioria simples. Para um consumidor ser considerado "suspeito", é necessário que 2/3 das regras indiquem esta classificação;
- mínimo: permite ao usuário do sistema informar quantas regras (no mínimo) precisam indicar que o consumidor é "suspeito" para que ele seja assim classificado.

4.6 Solução implementada

Este trabalho faz parte de um projeto maior, no qual toda uma equipe esteve envolvida nas diversas fases de desenvolvimento deste e de outros produtos. O foco deste trabalho foi a definição da arquitetura e das estruturas necessárias para o desenvolvimento do sistema baseado em conhecimento bem como a definição da linguagem de representação do conhecimento necessária. denominado SAUIPE. A linguagem definida por este trabalho foi fundamental para a elaboração deste sistema.

A implementação do sistema SAUIPE é apresentada no trabalho de Perim (2006).

As funcionalidades e interfaces do SAUIPE foram sugeridas por este trabalho e serão apresentadas nas próximas seções.

4.6.1 O sistema Sauipe

A interface principal do SAUIPE proporciona o acesso a inicialização do processo de aplicação das regras. A Figura 4.4 apresenta a tela principal do sistema. Por meio desta interface, realizam-se as seguintes operações:

- Seleciona-se a base de casos a ser avaliada, escolhendo o arquivo de dados no formato CSV;
- seleciona-se o formato da base de casos (formato que define a estrutura de colunas que existem na base de casos);
- seleciona-se o conjunto de regras que se deseja aplicar sobre a base de casos, possibilitando a configuração das propriedades de cada regra;
- seleciona-se o modo de avaliação desejado;
- inicia-se o processo de avaliação da base de casos.

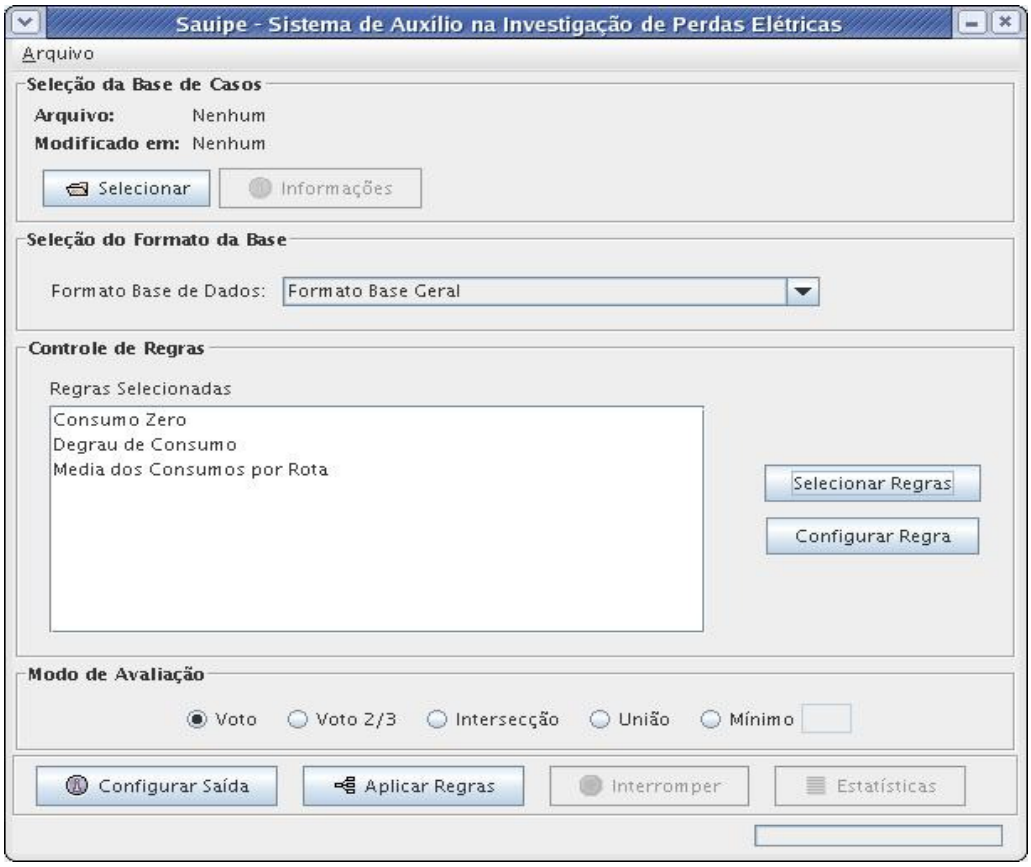

Figura 4.4 - Interface principal do Sauipe

A Figura 4.5 demonstra uma caixa de diálogo que permite ao usuário alterar os valores configuráveis das propriedades das regras. Esta interface é montada dinamicamente de acordo com cada regra e suas configurações. Na Figura 4.5, por exemplo, o sistema está apresentando os parâmetros (propriedades) da regra média dos consumos por atividade.

Figura 4.5 - Diálogo de configuração dos parâmetros de uma regra

Outra opção é selecionar como deve ser registrado o resultado da avaliação dos casos. A Figura 4.6 apresenta o diálogo de configuração de como será o formato de saída dos resultados. Uma opção é que no resultado final só liste os selecionados como suspeitos ou liste todos os consumidores indicando a classificação realizada. Outra opção refere-se ao nível de detalhe do resultado. Pode-se configurar para que o sistema exiba para cada consumidor os resultados de cada regra separadamente (resultado detalhado), ou se preferir configurar para que o sistema exiba apenas uma linha para cada consumidor e apresente os resultados das regras em colunas. O resultado final em qualquer um dos casos é sempre listado.

Figura 4.6 - Interface de configuração do formato de saída dos resultados

4.6.2 Edição de regras

A interface de edição de regras permite ao usuário cadastrar novas regras sem a necessidade de editar diretamente um arquivo XML. Esta interface está em fase de desenvolvimento. O sistema Sauipe não depende desta interface pois as regras podem ser definidas diretamente em XML, entretanto, esse módulo propiciará maior facilidade para o usuário. Através de interface apresentada nas figuras 4.7 e 4.8, o usuário poderá registrar uma nova regra: cadastrar seu nome e descrição, cadastrar suas propriedades e cadastrar o encadeamento de fórmulas/funções através de blocos "se ... então ... senão".

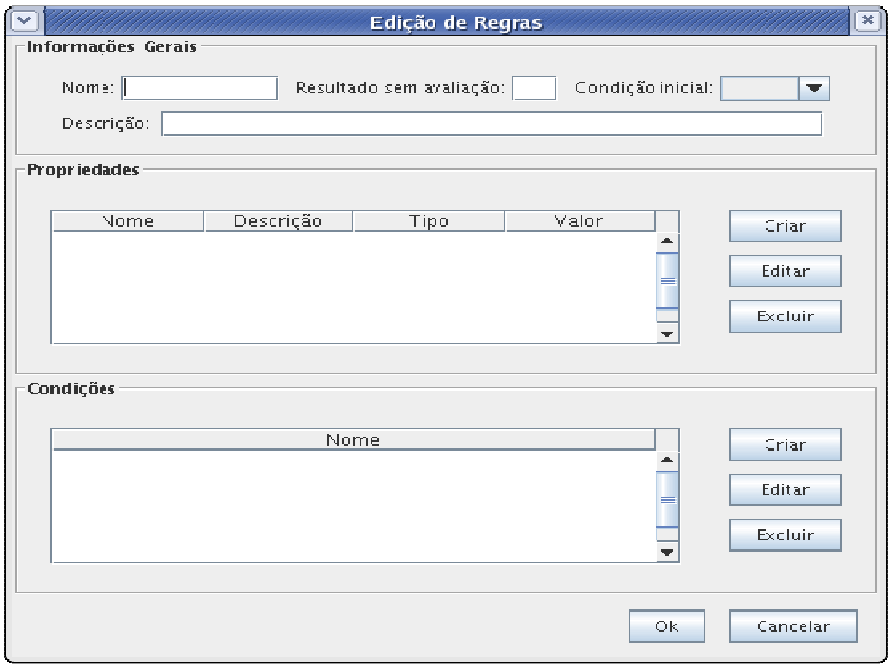

Figura 4.7 - Interface de edição de regras

A Figura 4.8 apresenta a interface da janela de edição de uma condição da regra.

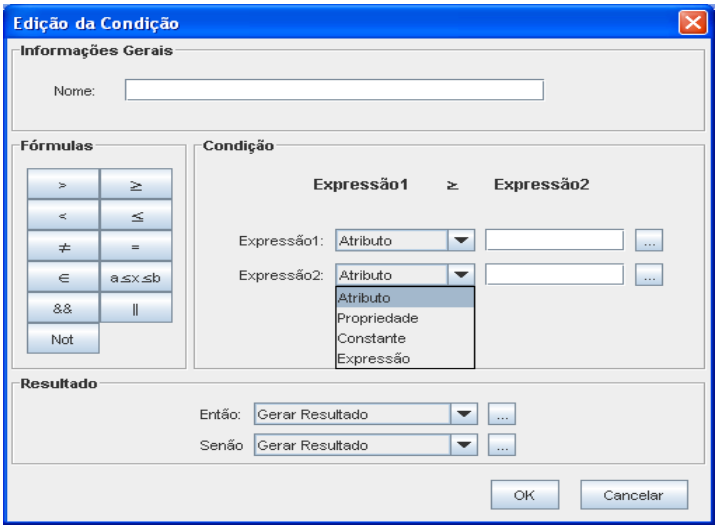

Figura 4.8 - Interface de edição de uma condição de uma regra

5 Descrição dos Experimentos

Como apresentado no capítulo anterior, as regras registradas na base de conhecimento podem possuir parâmetros diversos que ajustam a forma como a regra atuará. Por exemplo, a regra "avaliação por média" - apresentada na Seção 4.3.1 - possui um parâmetro denominado "fator da média". Esta regra verifica se o último consumo é inferior a uma determinada proporção da média dos últimos 3 consumos. O parâmetro "fator da média" define qual a proporção a ser avaliada. Por exemplo, se esse parâmetro for ajustado para 50%, a regra verificará se o último consumo é inferior a 50% da média dos últimos três meses de consumo do cliente.

De modo geral os especialistas não possuem um valor exato para definir este parâmetro – da mesma forma que não possuem valores exatos para definir os parâmetros de outras regras. Por este motivo, precisou-se realizar uma etapa de seleção de valores para os parâmetros das regras. Essa etapa inicial foi denominada de treinamento – termo utilizado na área de aprendizado de máquina para denominar o processo de preparação de classificadores mediante bases de exemplos. Pode-se considerar que esta etapa foi um refinamento da etapa de aquisição de conhecimento – prova disto é o fato desta fase de treinamento ter demandado grande interação com os especialistas.

Após o processo de treinamento, foi realizada a etapa de testes. Os testes foram realizados de duas formas: a primeira foi através de bases de exemplos que possuíam dados de consumidores que haviam sido inspecionados no passado. Desta forma poder-se-ia comparar a classificação indicada pelo SBC com a classificação real do consumidor – resultado da inspeção realizada no passado. A segunda forma de testes foi realizada através do envio de casos selecionados pelo sistema para inspeção de campo, avaliando assim o desempenho do SBC desenvolvido no ambiente real da distribuidora de energia elétrica do Espírito Santo.

5.1 Bases disponíveis para treinamento e teste

A concessionária de energia local disponibilizou uma base de dados com cerca de 40.600 clientes de diversos municípios do estado, que foram inspecionados no ano de 2005.

A base de exemplos possuía dados cadastrais dos clientes, dados de faturamento com histórico de cerca de 2 anos e os dados de resultados da última inspeção realizada em cada cliente – como descrito na Seção 3.2.

Esta base foi dividida em dois grandes conjuntos, de forma aleatória, de modo a um ser usado para treinamento e outro para testes. Uma base com 29.000 clientes (aproximadamente 70% da base completa) e outra com o restante dos clientes (11.600 clientes). A base de 29.000 clientes foi utilizada para a etapa de treinamento e a de 11.600 foi destinada aos testes.

Estas duas bases, treinamento e testes, foram re-divididas em diversas partes:

- A base de treinamento foi dividida de forma aleatória em 10 bases distintas, com o objetivo de individualmente serem utilizadas em cada procedimento de treinamento;

- a base de testes foi dividida em 4 bases distintas para que seja realizado o teste e verificação se os resultados foram compatíveis com os alcançados no treinamento.

Além da base de exemplos, a concessionária forneceu uma base de dados para consulta a campo que não possuía resultados de inspeções. Esta segunda base foi utilizada em um segundo momento para selecionar casos reais para investigação em campo e assim testar o sistema na prática.

5.2 Procedimento de Treinamento

O objetivo principal desta etapa é definir quais os melhores valores para serem utilizados nos parâmetros das regras. Vale lembrar que tais valores de parâmetros não serão necessariamente sempre os melhores valores, mas sim, um conjunto de parâmetros que atingiu em média os melhores resultados em uma amostra considerável de treinamento e que foram comprovadas em testes em bases independentes.

Em ambiente operacional (corporativo), esses valores de parâmetros devem ser reavaliados periodicamente com novos experimentos de treinamento. Em cada reavaliação desses valores, deve-se utilizar bases de exemplos recentes, visto que as características de comportamento dos consumidores podem se alterar com o passar do tempo.

Em um primeiro momento as regras foram aplicadas individualmente com o objetivo de aperfeiçoar as heurísticas. Após a aplicação das regras, alguns clientes eram analisados manualmente, averiguando se a classificação era ou não coerente. Confrontou-se também a classificação realizada com o resultado obtido na inspeção real realizada no passado. Com base nesta análise – realizada em conjunto com o especialista -, alterações foram feitas nas regras, de modo a melhorar as heurísticas empregadas.

Após este primeiro momento, foi realizado o processo referente à busca dos melhores valores para os parâmetros das regras. Nesta busca, cada regra foi trabalhada isoladamente. Basicamente, os procedimentos realizados para cada regra foram:

a) identificar os parâmetros da regra;

b) definir quais valores devem ser testados para cada parâmetro (definição dos limites da busca). Definir, por exemplo, que o parâmetro "fator da média" deve ser testado no intervalo entre 25% e 50%;

c) aplicar a regra sobre diversas bases de exemplos para cada conjunto de valores de parâmetros definido no passo anterior;

d) identificar o conjunto de parâmetros que apresentou melhor resultado médio a partir dos resultados encontrados;

e) aplicar a regra com o conjunto de parâmetros escolhido em bases de testes para verificar a eficácia destes parâmetros (etapa de testes, Sessão 5.3).

Para facilitar o entendimento de como foi realizado o treinamento, a próxima seção apresenta os dados da escolha do conjunto de parâmetros da regra "média por atividade" (treinamento) e a seção 5.3 apresenta os resultados dos testes destes parâmetros selecionados.

5.2.1 Realizando treinamento

Essa seção apresenta os dados do processamento do treinamento para a regra “média por atividade”. A apresentação desta seção segue o procedimento definido na seção anterior:

a) Identificar os parâmetros da regra

A regra média por atividade possui três parâmetros:

- Percentual mínimo ou Fator da média: Refere-se ao percentual mínimo (fator) da média que um consumidor pode possuir para ser considerado “normal”; de modo geral, um consumidor é considerado suspeito se seu último consumo for inferior a um percentual de sua média (percentual indicado por este parâmetro);
- descartar consumidor com irregularidade de leitura e ocorrência de faturamento: determina se a regra desconsiderará os consumidores com irregularidade de leitura ou ocorrência de faturamento;
- meses com inspeção normal: a heurística desta regra prevê que se o consumidor foi inspecionado em um período recente e obteve resultado "normal", então este não deve ser considerado suspeito. A proposta dos especialistas para este

parâmetro é de 6 meses. Assim: se o consumidor tiver sido inspecionado com resultado normal nos últimos 6 meses, então este não é considerado suspeito.

A realização do treinamento sobre um conjunto de bases de exemplos, determinará quais valores devem ser utilizados para cada um destes parâmetros.

b) Definir quais valores devem ser testados para cada parâmetro investigado

- Fator da média (Percentual mínimo): Para este parâmetro os especialistas determinaram que este percentual devesse estar entre 15% e 60% (0,15 e 0,60). Investigou-se portanto todo o intervalo entre 0,15 e 0,70;
- descartar consumidor com irregularidade de leitura e ocorrência de faturamento: Este parâmetro possui duas possibilidades: “sim”(1) ou “não”(0). Ambas foram avaliadas no treinamento.
- meses com inspeção normal: este parâmetro foi considerado fixo em fase de treinamento. Para efeito de treinamento este parâmetro não tem sentido pois na base de casos de exemplo só existe a informação da última inspeção que foi realizada no cliente, e esta informação é utilizada como ponto de referência para a análise de sucesso ou não da predição realizada pelo sistema. Além disso, os especialistas determinaram que este parâmetro existe apenas para dar-lhe uma margem de configuração, mas que em geral este parâmetro será fixado em 6 meses.

c) aplicar a regra sobre diversas bases de exemplos para cada conjunto de valores de parâmetros definido no passo anterior

Neste passo, utilizou-se o Sauipe para aplicar a regra sobre as bases de treinamento com os parâmetros utilizando cada um dos valores determinados no passo anterior. Os resultados destes testes estão registrados na Tabela 5.1. As métricas utilizadas foram apresentadas na Seção 2.3.1. Todas elas são baseadas na matriz de confusão. A matriz de confusão – também explicada na Seção 2.3.1 – confronta a classificação real dos exemplos com a classificação dada pela aplicação da regra através do SBC.

Nesta etapa, o sistema automaticamente aplicou a regra em questão, com cada conjunto de parâmetros em treinamento para cada uma das 10 bases de treinamento. Isso gerou uma grande massa de dados de resultados. A Tabela 5.1 apresenta uma parte destes dados a título de exemplo. Nesta tabela, cada linha equivale a uma aplicação da regra sobre uma base de treinamento. Nota-se que o mesmo conjunto de parâmetros foi aplicado sobre

cada uma das bases de treinamento de modo independente, obtendo resultados individuais para cada aplicação.

Param1 – Fator da média	Param2 (Descart. Irreg.)	Base de Treino	Acerto (%)	c (Conf. Negativa)	e (especif.)	h (média harmônica)
0,30	0 (Não)	b01	81,053	0,207	0,321	0,2317
		b02	80,296	0,205	0,338	0,2326
		b03	80,248	0,205	0,337	0,2327
		b04	79,954	0,214	0,303	0,2345
		b05	80,467	0,257	0,384	0,2849
		b06	81,527	0,222	0,331	0,2465
		b07	81,644	0,218	0,331	0,2424
		b08	80,628	0,197	0,309	0,2211
		b09	81,787	0,215	0,352	0,2436
		b10	79,827	0,180	0,287	0,2023
0,30	1 (Sim)	b01	88,723	0,240	0,134	0,1936
		b02	88,058	0,188	0,121	0,1611
		b03	88,110	0,237	0,153	0,2036
		b04	87,197	0,232	0,127	0,1857
		b05	88,010	0,281	0,168	0,2341
		b06	88,429	0,220	0,131	0,1829
		b07	88,579	0,274	0,174	0,2336
		b08	88,759	0,239	0,123	0,1858
		b09	89,030	0,228	0,152	0,1984
		b10	87,247	0,197	0,129	0,1702
0,31	0 (Não)	b01	80,695	0,202	0,321	0,2276
		b02	79,992	0,203	0,342	0,2309
		b03	79,985	0,204	0,341	0,2316
		b04	79,645	0,211	0,307	0,2328
		b05	80,133	0,253	0,387	0,2820
		b06	81,091	0,217	0,335	0,2428
		b07	81,419	0,216	0,335	0,2414
		b08	80,370	0,197	0,316	0,2219
		b09	81,523	0,213	0,356	0,2421
		b10	79,639	0,179	0,291	0,2024
0,31	1 (Sim)	b01	88,535	0,232	0,134	0,1900
		b02	88,017	0,192	0,126	0,1662
		b03	88,037	0,239	0,158	0,2072
		b04	86,921	0,220	0,127	0,1803
		b05	87,774	0,270	0,169	0,2286
		b06	88,264	0,220	0,136	0,1852
		b07	88,508	0,270	0,175	0,2320
		b08	88,726	0,248	0,132	0,1962
		b09	88,992	0,227	0,153	0,1978
		b10	87,252	0,197	0,130	0,1705

Tabela 5.1 – Resultado da aplicação da regra Média por atividade para escolha de parâmetro

A tabela 5.1 apresenta apenas uma parte dos dados de treinamento. O trecho onde os parâmetros variaram entre 0,30 e 0,31 e o segundo parâmetro entre 0(Não) e 1 (Sim). A partir destes dados de resultados individuais, calculou-se o desempenho médio de cada conjunto de

parâmetros aplicando-se a média sobre a métrica média harmônica. Obteve-se assim a tabela 5.2, que representa o resultado médio de cada conjunto de parâmetros.

Param1 – Fator da média	Param2 (Descart. Irreg.)	h (média harmônica)
0,30	0	0,2372
0,30	1	0,1949
0,31	0	0,2355
0,31	1	0,1954
0,32	0	0,2366
0,32	1	0,1980
0,33	0	0,2378
0,33	1	0,2030
0,34	0	0,2371
0,34	1	0,2052
0,35	0	0,2347
0,35	1	0,2024

Tabela 5.2 – Desempenho médio de cada conjunto de parâmetros

Para melhor apresentar os dados dos resultados (exemplificados na Tabela 5.2), estes estão apresentados no gráfico da Figura 5.1. Neste gráfico, os valores do parâmetro 1 estão mapeados pela escala no eixo horizontal e os valores no eixo vertical referem-se a média harmônica. Observando-se que os testes com parâmetro 2 igual a 1 resultaram em resultados bem diferentes dos testes nos quais este parâmetro foi ajustado para 0, optou-se pela apresentação dos dados em duas curvas. Uma curva para os casos onde o Parâmetro 2 (descartar consumidores com irregularidade de consumo) foram testados como sendo 0 e outro como sendo 1.

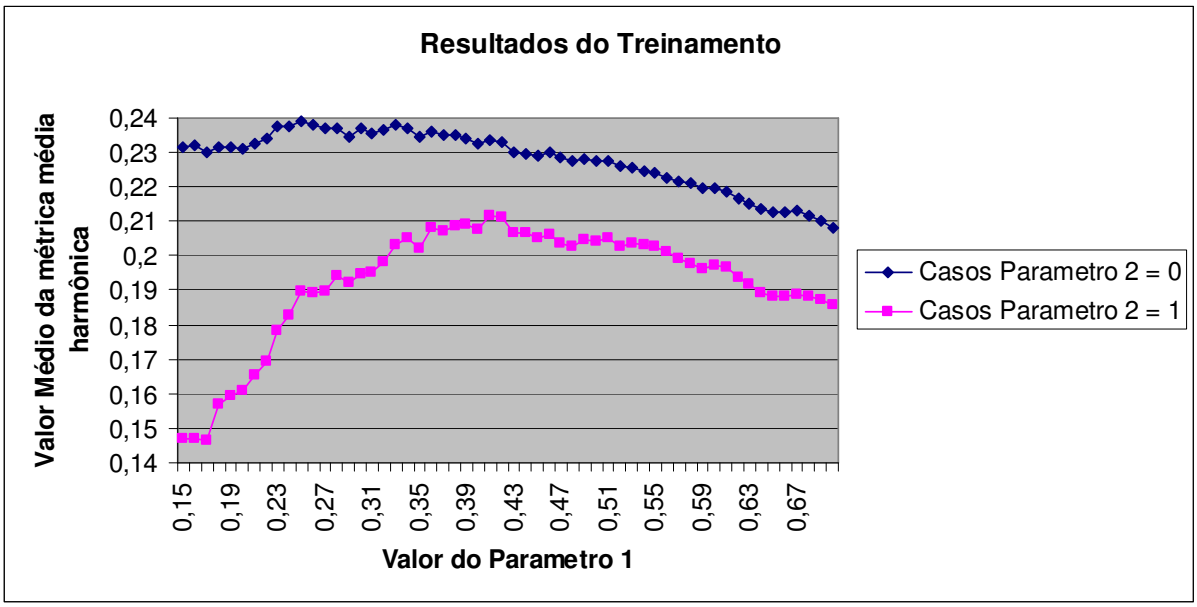

Figura 5.1 – Resultados do treinamento para a regra média por Atividade

No gráfico apresentado, cada ponto representado não se refere a uma aplicação da regra sobre uma base, mas sim à média dos resultados de várias aplicações da regra nas diversas bases de treinamento.

Como o objetivo era maximizar o resultado observando a métrica média harmônica, observou-se que o parâmetro 2 apresentava resultados melhores sempre que ajustado para 0. Assim, considerando apenas estes casos, tem-se no gráfico da Figura 5.2 uma representação mais clara dos resultados, que permite verificar a convergência dos resultados para o parâmetro 1 ajustado para 0,25. O gráfico da Figura 5.2 retrata os mesmos dados da Figura 5.1, desconsiderando, entretanto, os casos de treinamento com parâmetro 2 ajustado para 1 (segunda curva na Figura 5.1).

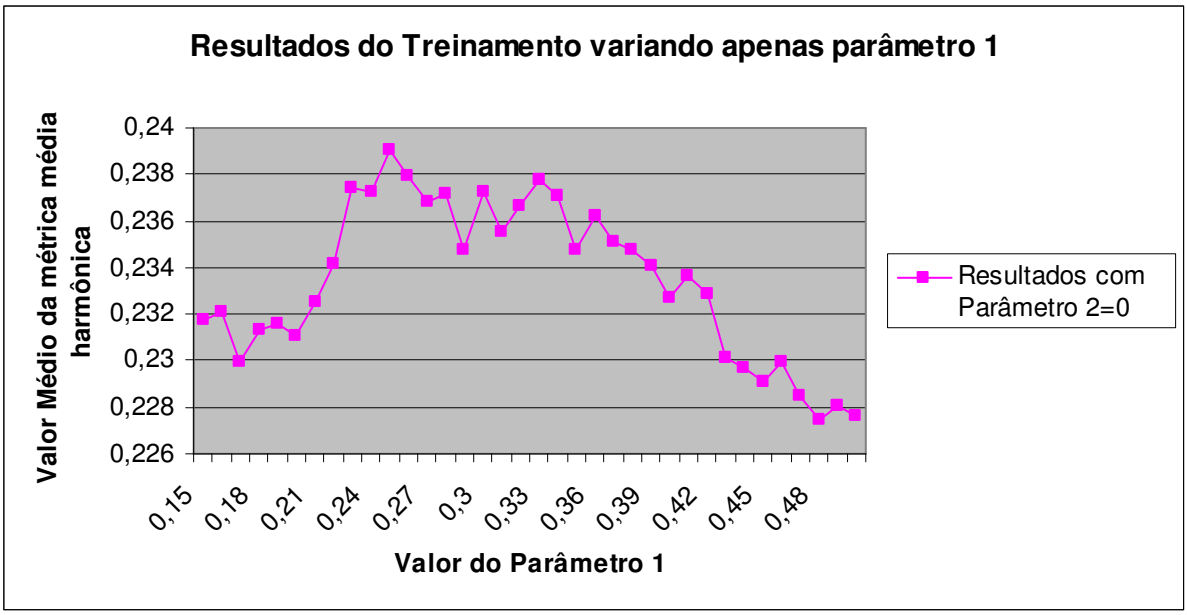

Figura 5.2 – Resultados do treinamento para a regra média por Atividade com parâmetro 2 = 0

d) identificar o conjunto de parâmetros que apresentou o melhor resultado médio a partir dos resultados encontrados

Como explicado anteriormente, para verificar qual experimento obteve melhor resultado utilizou-se a métrica média harmônica – como apresentado na Seção 2.3.1. Esta métrica foi considerada a mais adequada por buscar um ponto de equilíbrio entre as outras métricas. A partir do gráfico, verificou-se que os melhores resultados foram alcançados quando o parâmetro 1 era próximo de 0,25. A Tabela 5.3 apresenta os dados da região do gráfico que apresenta os melhores resultados.

Parâmetro 1 - Fator da Média	Parâmetro 2 (Descart. Irreg.)	h (média harmônica)
0,21	0	0,233
0,23	0	0,237
0,25	**0**	**0,239**
0,27	0	0,237
0,29	0	0,235
0,31	0	0,236

Tabela 5.3 – Melhores resultados

Com este processo completado, e a devida avaliação dos gráficos e da tabela de resultados realizada (Tabela 5.3), definiu-se que o parâmetro 1 ("fator da média") da regra analisada deve ser utilizado com o valor 0,25. Ou seja, os consumidores serão considerados suspeitos de fraude caso seu último consumo esteja mais que 25% abaixo da média da sua atividade. Quanto ao parâmetro 2 (desconsiderar consumidores com irregularidade de consumo), o treinamento indicou que este dever ser usado com valor = 0 (ou seja, não se deve desconsiderar os consumidores que possuem irregularidade de consumo). A próxima etapa constitui da validação destes parâmetros sobre bases de teste para concluir se os resultados serão estáveis e no mesmo nível dos encontrados em treinamento.

e) aplicar a regra com o conjunto de parâmetros escolhido em bases de testes para verificar a eficácia destes parâmetros

Esta etapa de verificação dos parâmetros encontrados é denominada de experimentos de teste e está apresentada na próxima sessão.

5.3 Experimentos de Teste

Após a realização do treinamento, onde se definiu quais os valores deveriam ser utilizados para os parâmetros das regras, realizaram-se os experimentos de teste que estão apresentados nesta sessão.

Os experimentos de teste servem para verificar o desempenho das regras com os parâmetros definidos e prever a eficiência do sistema em ambiente real. Não se deve considerar os resultados dos experimentos de treinamento pois estes experimentos podem estar super-ajustados (overfitting): após o treinamento, as regras podem resultar em bom resultado nas bases de treinamento devido a estarem super-ajustadas para estas bases. Por esta razão, utiliza-se uma ou mais bases de dados de teste diferentes das bases de treinamento para a validação dos resultados.

Para a regra Média por Atividade, estabeleceu-se para o procedimento de teste a averiguação dos resultados sobre quatro bases de testes distintas (geradas a partir da divisão aleatória da base destinada a teste). A Tabela 5.4 apresenta os resultados da aplicação da Regra Média por Atividade com os parâmetros definidos no treinamento.

Base	Parâmetro 1 - Fator da Média	Parâmetro 2 (Descart. Irreg.)	Média harmônica (h)	Confiabilidade Negativa
Base Teste 1	0,25	0	0,245	0,221
Base Teste 2	0,25	0	0,278	0,255
Base Teste 3	0,25	0	0,220	0,202
Base Teste 4	0,25	0	0,248	0,223
Média da média harmônica = 0,248				
Média da Confiabilidade negativa = 0,225				

Tabela 5.4 – Resultado dos testes

A Figura 5.3 representa graficamente a tabela 5.4, auxiliando a comparação dos resultados.

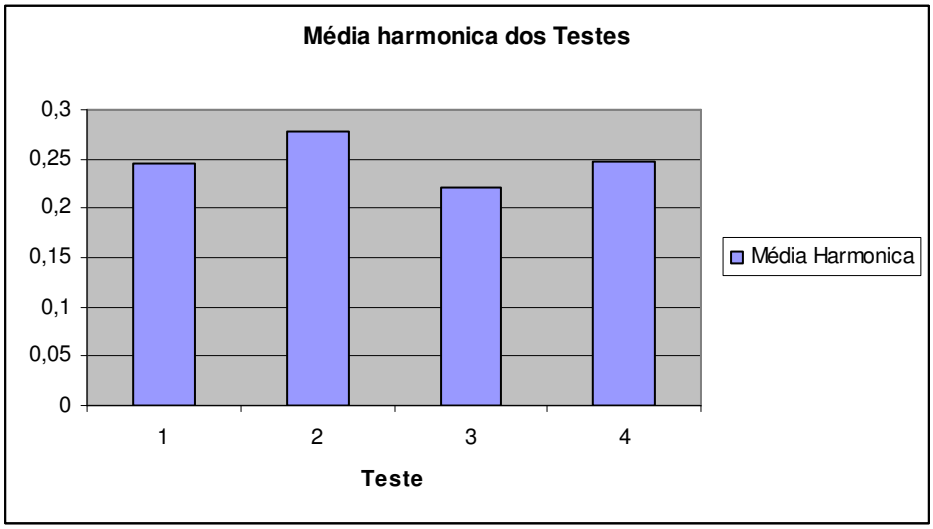

Figura 5.3 – Resultados dos testes com os parâmetros definidos no treinamento

Analisando a Tabela 5.4 e Figura 5.3, verifica-se que os experimentos de testes demonstraram que os resultados atingidos no treinamento foram mantidos. Além de obter um resultado médio acima do encontrado no treinamento (0,248 no teste e 0,239 no treinamento), todos os resultados de teste tiveram bom resultado (mesmo o pior desempenho, no teste 3, obteve um resultado próximo aos demais resultados). Isso indica que os parâmetros foram adequadamente selecionados pelo processo de treinamento.

Além de serem bons resultados quando comparados aos alcançados em treinamento, um indicativo positivo de sucesso neste processo é a comparação destes resultados com os

índices que os especialistas possuem em suas atuais formas de seleção e realização de inspeção. O índice atual dos especialistas em campo possui acerto de cerca de 12% dentre os inspecionados. Esse índice de sucesso dos especialistas pode ser comparado à métrica confiabilidade negativa, pois ela representa o número de acertos sobre o total de consumidores indicados para inspeção (classificados como suspeitos de fraude). Observando a Tabela 5.4 pode-se verificar que todos os testes tiveram a métrica confiabilidade negativa com índice superior a 20%.

Deste modo, conclui-se que os parâmetros definidos no treinamento são válidos e podem ser aplicados, uma vez que o teste confirmou-se através dos testes a eficiência dos mesmos.

Este procedimento completo de treinamento e teste foi repetido para cada uma das regras investigadas neste trabalho, gerando um conjunto de parâmetros que melhor se aplicam segundo o treinamento realizado. Ressalva-se que tais parâmetros não são os melhores parâmetros para qualquer base de casos, entretanto, são os parâmetros que apresentaram melhor desempenho médio nos treinamentos realizados. Além do treinamento, estes parâmetros foram verificados sobre as bases de testes, averiguando assim que os parâmetros mantinham os resultados alcançados no treinamento.

As seções seguintes apresentam as configurações de parâmetros encontrados nos experimentos (valores dos parâmetros encontrados na etapa de treinamento) e os resultados destes parâmetros nas bases de teste.

5.3.1 Parâmetros encontrados para cada regra

Esta seção apresenta os valores dos parâmetros encontrados em tempo de treinamento. Esses parâmetros foram utilizados posteriormente na aplicação dos testes.

a) Consumo zero: busca por consumidores com um consumo inferior a um valor mínimo nos últimos meses. Configuração dos parâmetros da regra:

- Meses a avaliar = 12;
- Meses da base = 24;
- Meses acima do consumo mínimo = 3;
- Consumo mínimo = 30;
- Imóveis desocupados (classe) = (relação das classes que representam imóveis desocupados);
- Áreas de veraneio (não definido);
- período férias = 1, 7 (janeiro e julho);

- meses com inspeção normal = 12

b) Degrau de consumo: Busca por consumidores que tiveram variação de consumo superior a 2 sigma, ou seja, consumidores que tiveram uma queda de consumo no último mês.

Parâmetros:

- grupo sazonal (não definido);
- período férias = 1, 7 (janeiro e julho);
- período de férias coletivas = 1 (janeiro);
- meses com inspeção normal = 6.

c) Média dos últimos 3 meses: Busca por consumidores com último consumo inferior a um percentual da média dos últimos 3 consumos. Configuração dos parâmetros da regra:

- Percentual mínimo = 0,60;
- Meses com inspeção normal = 12;
- Descartar consumidor com irregularidade de leitura e ocorrência de faturamento = não;

d) Média por atividade: Busca por consumidores com último consumo inferior a um percentual da média dos consumos dos consumidores com a mesma atividade do consumidor analisado. Parâmetros:

- Percentual mínimo = 0,25;
- Meses com inspeção normal = 6 (determinado pelo especialista);
- Descartar consumidor com irregularidade de leitura e ocorrência de faturamento não;

e) Média por rota: Busca por consumidores com último consumo inferior a um percentual da média dos consumos dos consumidores com a mesma rota a qual o consumidor analisado está ligado. Parâmetros:

- Percentual mínimo = 0,20;
- Meses com inspeção normal = 6 (determinado pelo especialista);
- Descartar consumidor com irregularidade de leitura ou ocorrência de faturamento = não;

5.3.2 Resultados dos experimentos de teste individuais

A Tabela 5.5 apresenta os resultados médios finais dos experimentos individuais de teste sobre as diversas regras avaliadas. Estas regras passaram por treinamento e teste como apresentado nas sessões anteriores.

Regra	Percentagem de Acerto(%)	Confiabilidade Negativa	Especificidade	Média Harmônica
Consumo Zero	85,805	0,226	0,143	0,193
Degrau de Consumo	87,866	0,149	0,031	0,069
Média dos últimos 3 meses	83,956	0,202	0,186	0,197
Média por atividade	82,267	0,225	0,324	0,248
Média por rota	81,749	0,215	0,316	0,238

Tabela 5.5 – Resultado da aplicação individual das regras usando a base de testes

Os resultados destes experimentos de testes são considerados positivos pois mantiveram-se com confiabilidade negativa superior aos índices que os especialistas possuem nas atuais formas de seleção e realização de inspeção (cerca de 12% de acerto dentre os inspecionados). Nota-se na Tabela 5.5 que todas as regras tiveram a métrica confiabilidade negativa com índice superior a 12%.

5.3.3 Experimentos em Campo

O trabalho de escolha das regras e da metodologia de agrupamento para realização dos testes em campo foi realizado junto ao especialista, utilizando-se assim do conhecimento e experiência do mesmo na busca de melhores resultados. As considerações básicas para estas escolhas foram os resultados do treinamento e teste realizados na ocasião junto ao especialista. O processo foi o mesmo apresentado nas seções anteriores. Desta forma, os especialistas escolheram a seguinte a configuração para os testes em campo:

- Regras: “média por atividade”, “média por rota”
- Modo de avaliação escolhido: Interseção

Um dos motivos que influenciou na escolha do modo de avaliação por interseção foi o de este método ser o que retorna o menor número de casos de clientes suspeitos (tinha-se um limite definido pela distribuidora no período em que os testes foram solicitados).

A base de casos utilizada foi uma base de dados de uma região da Grande Vitória (ES) que era de interesse de investigação por parte da concessionária local. Dentre os casos selecionados como suspeitos de irregularidades pelo sistema, foram escolhidos 200 casos aleatoriamente para que estes fossem enviados a campo para inspeção. Dentre as 200 inspeções, o último levantamento obtido da distribuidora indica que 184 haviam sido

realizadas até o momento da escrita deste trabalho, e o resultado encontrado em campo está apresentado na Tabela 5.6.

Classificação	Total de consumidores
Normal	134
Irregular	50

Tabela 5.6 - Resultado obtido em campo

A Tabela 5.6 demonstra que o sistema obteve um percentual de acerto de 27,17%, um número expressivo quando comparado a resultados de outras operações de investigação deste gênero. A taxa de acerto das operações de varredura estão na faixa de 12%. E a taxa de acerto das inspeções motivadas por denúncias é de 22%, sendo que estas, motivadas por denúncias, são as que de modo geral resultam nos melhores índices.

Quando comparado aos índices de acerto conseguidos com a ferramenta de mineração de dados MIP, tais números também são expressivos, uma vez que a taxa de acerto do MIP foi, nos testes divulgados, de cerca de 23,20% de sucesso (Cometti et al., 2005).

Os experimentos de campo dependem da disponibilidade da concessionária em realizar as inspeções. Esse fator limita a possibilidade de realizar um amplo teste do sistema em ambiente real. Por este motivo, o experimento realizado em campo foi relativamente pequeno e preliminar. Todavia, apesar de serem em número limitado, são resultados que enfatizam os resultados dos experimentos com bases de teste (apresentados nas seções anteriores), pois os resultados de campo não só acompanham as tendências mas ainda superam as expectativas de tais experimentos de teste.

6 Conclusões e Trabalhos Futuros

Este capítulo apresenta as conclusões desta pesquisa, revisando o que foi realizado e o que foi atingido de resultado final. A segunda seção apresentará um pouco sobre as dificuldades e os desafios enfrentados para a realização do trabalho e por fim a última seção indicará trabalhos que podem ser realizados a partir deste.

6.1 Conclusões

Selecionar quais consumidores de uma empresa de distribuição de energia elétrica devem ser inspecionados é o problema que motiva este trabalho. Uma forma amplamente utilizada por algumas distribuidoras é a execução de inspeções a todos os consumidores de determinadas regiões onde as perdas são muito elevadas. Entretanto, esta modalidade de filtro gera milhares de inspeções que muitas vezes não atingem resultados satisfatórios, desperdiçando tempo e dinheiro. Por este motivo, é de grande importância o desenvolvimento de soluções para apoiar a indicação de suspeitos de irregularidades ou fraudes para que apenas estes sejam inspecionados.

Duas abordagens são indicadas como promissoras para a realização desta tarefa de seleção de suspeitos de irregularidades:

- uma é a utilização de Data Mining e aplicação de técnicas como Redes Neurais e outros modelos computacionais para a geração de classificadores, (Queiroga, 2005), (Eller, 2003) , (Cometti et al., 2005);
- a segunda abordagem trata do uso do conhecimento dos especialistas para realização desta seleção de suspeitos.

O foco desta dissertação foi exatamente nesta segunda abordagem, uma vez que a primeira fora explorada por trabalhos complementares a este (Queiroga, 2005).

O estudo iniciou-se pela análise do conhecimento especialista a respeito de fraudes e irregularidades existentes nas redes de distribuição de energia elétrica, mais especificamente nas instalações dos consumidores deste setor.

Neste estudo, verificou-se que o conhecimento existente compõe-se de diversas heurísticas. Essas heurísticas – também denominadas regras - determinam meios de classificar um consumidor como sendo ou não suspeito de fraude ou irregularidade. Entretanto, o uso dessas heurísticas sem a ajuda de ferramentas adequadas trata-se de uma tarefa árdua ou mesmo inviável. O fato das bases de informações possuírem milhares de dados a serem

avaliados, e o fato dessas regras muitas vezes envolverem diversos cálculos e grandes volumes de dados é o principal motivo desta complexidade. Por este motivo, o conhecimento especialista acaba por ser sub-aproveitado.

A fim de colaborar com a utilização deste conhecimento, propôs-se um sistema baseado em conhecimento, denominado SAUIPE, capaz de registrar este conhecimento e utilizá-lo em grandes quantidades de dados, apoiando a realização da seleção de suspeitos de fraude. A partir deste sistema, pôde-se realizar um refinamento das heurísticas dos especialistas, determinando melhores parâmetros para sua aplicação e contribuindo assim para o melhor uso das mesmas.

O refinamento e avaliação do sistema foi realizado com bases de casos de exemplos reais de consumidores que foram inspecionados no passado. Além de utilizar bases de dados reais para refinamento e teste, o sistema foi testado em campo (em ambiente real). A distribuidora local de energia disponibilizou uma base de dados de consumidores que foi submetida à seleção de suspeitos de irregularidades pelo sistema. A esses suspeitos selecionados foram realizadas inspeções para averiguar possíveis fraudes. Os resultados dos testes de exemplo e do teste em ambiente real atingiram bons resultados, especialmente se comparados a resultados de outras técnicas de seleção de suspeitos utilizadas.

A solução proposta envolveu a definição de uma nova linguagem de representação, denominada XmlRuleLang, similar a outras em desenvolvimento recente – como apresentado em Lee & Sohn (2003) e RuleML (2006) - e, portanto, condizente com as atuais pesquisas em andamento, que indicam o uso de XML como opção de estrutura para novas alternativas de representação do conhecimento.

As contribuições deste trabalho iniciam-se pela apresentação de um caso onde uniu-se pesquisa e aplicação corporativa. O presente trabalho aplicou os conceitos de sistema baseados em conhecimento em um ambiente real, apresentando os passos básicos e os desafios a serem considerados no desenvolvimento de sistemas deste gênero.

Todavia, a contribuição principal foi o estudo realizado sobre as heurísticas dos especialistas. Além de contribuir para a melhor definição e posterior documentação destas heurísticas na base de conhecimento, esta pesquisa realizou também a definição de parâmetros que maximizam os resultados das mesmas tendo por base os dados históricos reais analisados neste trabalho.

O desenvolvimento de um sistema baseado em conhecimento foi importante pois foi utilizado para modelar o conhecimento dos especialistas em detecção de fraude e utilizar este conhecimento em prol da seleção de suspeitos.

Um dos objetivos alcançados foi a documentação e formalização do conhecimento especialista em uma base de conhecimento disponível e utilizável por outros profissionais ou sistemas. Algo comum de acontecer em diversos setores é a existência de documentos desatualizados. Isso ocorre, por exemplo, com manuais de sistemas que em muitos casos não são atualizados quando os sistemas sofrem evoluções.

Essa desatualização não ocorre com as regras registradas no SBC proposto, uma vez que a documentação da regra é a própria base de conhecimento do sistema. Basicamente essa base de conhecimento permite:

- disponibilizar o conhecimento a outros sistemas e
- garantir que a informação compartilhada entre sistemas ou profissionais estará sempre atualizada e condizente com as regras utilizadas pelos especialistas.

Uma vez que uma regra seja atualizada na base de conhecimento, qualquer aplicação ou profissional que acessar as regras terá acesso à nova regra, não correndo o risco de acessar uma documentação desatualizada.

Uma característica da proposta de grande importância para a concretização do compartilhamento da base de conhecimento é o uso de XML como estrutura para a linguagem definida. Essa característica tem motivado várias outras propostas de linguagem de representação de conhecimento baseadas em XML.

Uma outra contribuição deste trabalho refere-se a própria linguagem que foi definida. A linguagem proposta, denominada XmlRuleLang, é uma linguagem de representação de regras que não precisa estar limitada ao domínio de distribuição de energia, mas que pode ser utilizada em outros domínios de problema. Além disso, trata-se de uma representação que possui suas primitivas em língua portuguesa, fato que facilita sua utilização ao tratar de trabalhos realizados nesta língua – em especial quando envolvem especialistas que venham a manipular a base de conhecimento sem uso de ferramenta gráfica ou ainda usuários ou analistas que queiram analisar ou alterar uma regra através de sua definição em XmlRuleLang.

Outras linguagens similares em geral propõem-se a atender um amplo número de requisitos, gerando complexidade no uso de tais abordagens. Por este motivo, apesar de possuir propostas similares, a linguagem definida fica como opção para casos em que os requisitos sejam compatíveis com este trabalho.

Uma crítica que se pode fazer a este trabalho é a ausência de um módulo de explicação. O presente trabalho não possuía o objetivo de propor um sistema de explicação, entretanto este poderia ser um módulo que agregaria valor a solução definida, facilitando sua

utilização. Este módulo ajudaria os especialistas a aprimorarem as bases de conhecimento uma vez que teriam um mapeamento dos motivos pelos quais o SBC classificou ou deixou de classificar determinado caso como suspeito.

Outra crítica refere-se aos testes realizados em ambiente real. A amostragem realizada para estes testes é pequena para garantir a boa avaliação do sistema. Apesar dos testes em ambiente real representarem pouco para se avaliar o sistema, os testes realizados com as bases de exemplos demonstraram bons resultados e uma perspectiva de sucesso - aparentemente comprovada com testes preliminares em ambiente real.

6.2 Dificuldades e desafios enfrentados

O acesso a dados corporativos é algo naturalmente restrito, e o uso destes dados em pesquisas e em desenvolvimento de sistemas para as distribuidoras é em geral dificultado pelas regras de sigilo das empresas. Com isso, encontra-se uma dificuldade nata em trabalhos desta natureza. Além de o acesso ser limitado a certo volume de dados, não se tem acesso a todas as informações que poderiam ser utilizadas, e não se tem acesso há um grande histórico de informações. Essas limitações restringem o potencial que poderia ser atingido.

Uma outra dificuldade é a existência de uma grande quantidade de dados com ruídos ou mesmo incompletos – nulos – nas bases de dados fornecidas para análise. A existência de ruídos ou nulos muitas vezes inviabiliza a utilização de muitos exemplos – casos reais de clientes com inspeções anteriores – que poderiam ser usados para avaliar a precisão das regras desenvolvidas. Em parte destes casos consegue-se tratar eventuais nulos ou ruídos, entretanto vários casos são descartados por ausência de dados necessários para algumas heurísticas.

Como o acesso aos dados é de certo modo restrito e sigiloso, muitos atributos fornecidos foram codificados para não fornecerem informações claras quanto aos dados que estavam sendo utilizados. Alguns atributos vieram com os dados como sendo apenas as chaves utilizadas no data warehouse da distribuidora, fator que não gerou dificuldades em alguns casos, mas que em outros complicava o entendimento em analisar o que se estava tratando.

Outra dificuldade - comum na definição e construção de sistemas baseados em conhecimento - foi a pouca disponibilidade de tempo dos especialistas. Normalmente os especialistas estão envolvidos com diversas atividades que não os permite dedicar o tempo ideal para a construção de um SBC. As dificuldades de se reunir, as reuniões desmarcadas ou

realizadas às pressas foram situações relativamente normais e compreensíveis mediante o volume de atividades que os mesmos desempenham.

Os desafios citados, entre outros, são de certo modo característicos de sistemas baseados em conhecimento e previsíveis dentro da literatura existente. Sobretudo, estes desafios dificultam mas não invalidam e nem impedem a definição e construção de um SBC.

6.3 Trabalhos futuros

Um trabalho que influenciaria positivamente na utilização do SAUIPE seria o desenvolvimento de um sistema ou módulo de explicação. Este módulo deverá trabalhar em conjunto com o módulo de inferência, acompanhando e registrando os passos realizados e os caminhos seguidos pela execução das regras. O objetivo deste módulo seria de poder apresentar ao usuário do sistema uma explicação sobre os motivos de uma determinada classificação realizada pelo sistema.

Um trabalho mais robusto que poderia ser implementado seria a integração do SAUIPE com sistemas que utilizem Data Mining - como é o caso do MIP (Queiroga, 2005). A utilização destas ferramentas em conjunto resultaria em uma solução híbrida que poderia potencializar o poder de precisão nas classificações realizadas.

Um trabalho que é naturalmente continuação deste é a implementação da interface de edição de regras – que atualmente se encontra em estágio de desenvolvimento. A interface de edição de regras não é pré-requisito para a utilização do sistema. A manipulação da base de conhecimento pode ser realizada diretamente nos arquivos de definição de regras. Todavia, apesar da linguagem de representação utilizar XML e ser de fácil compreensão e alteração, a existência de uma interface gráfica que permita a edição de regras em uma interface amigável, potencializará o uso da ferramenta.

A ausência de um validador sintático de regras - carência do atual trabalho - poderia ser suprida em um trabalho futuro. Atualmente espera-se que o módulo de edição de regras venha a validar as regras inseridas, entretanto, regras que tenham sido inseridas ou alteradas manualmente ou através de outras ferramentas deveriam ser validadas por alguma funcionalidade do sistema.

Para implementar essa validação, uma solução simples seria utilizar um recurso do próprio XML: os DTD's. Um DTD - Document Type Definition, ou definição de tipo de documento - define como as tags são estruturadas em uma determinada categoria de documentos XML. Este DTD permite validar se um documento está ou não de acordo com as

regras de definição previstas para aquele tipo de documento. Assim, definir um DTD para a linguagem proposta, a XmlRuleLang, e implementar um acionamento da validação das regras a partir deste DTD permitiria a realização de validação das regras definidas.

A ferramenta desenvolvida, tendo apresentado bom desempenho nos testes preliminares, tende a ser melhor explorada pelos especialistas. Desta forma, as regras definidas evoluirão de forma natural, podendo demandar novas necessidades. A linguagem de representação definida visa disponibilizar ampla flexibilidade para a construção de regras, entretanto, podem surgir necessidades de novas primitivas para atender à necessidades não previstas até o presente momento.

Apêndice A: Descrição de Expressões e Fórmulas na Linguagem de Representação

As expressões e fórmulas foram definidas na linguagem de representação para realizar o conjunto de operações necessárias na análise de condições e na geração de resultados das regras.

Chamamos de expressão toda fórmula que realiza uma operação 'booleana', ou seja, que retorna sempre um valor 'verdadeiro' ou 'falso'. As expressões definidas foram:

- **and**: Expressão que só retorna verdadeiro quando todas as expressões condicionais subseqüentes são verdadeiras. As expressões subseqüentes também devem ser condicionais. Exemplo:

```
<expressao nome="and">
      <formula nome="diferente">
            <base atributo=”irreg24”/>
            <const valor=”0”/>
      </formula>
      <formula nome="diferente">
            <base atributo=”irreg23”/>
            <const valor=”0”/>
      </formula>
</expressao>
```

- **or**: Expressão que só retorna falso quando todas as expressões condicionais subseqüentes são falsas. As expressões subseqüentes também devem ser condicionais Exemplo:

```
<expressao nome="or">
      <expressao nome="not">
            <propriedade nome=”b1”/>
      </expressao>
      <formula nome="pertence">
            <base atributo=”cod_grupo”/>
            <propriedade nome=”r4”/>
      </formula>
</expressao>
```

- **not**: Expressão que retorna verdadeiro quando a expressão condicional subseqüente é falsa, e retorna falso quando a expressão condicional subseqüente é verdadeira. Exemplo:

```
<expressao nome="not">
      <formula nome="pertence">
            <base atributo=”cod_grupo”/>
            <propriedade nome=”r4”/>
      </formula>
</expressao>
```

No sistema foram definidos dois tipos de fórmulas: fórmulas condicionais e fórmulas numéricas. As fórmulas condicionais realizam operações 'booleanas'. As seguintes fórmulas condicionais foram definidas no sistema:

- **maior**: Fórmula que retorna verdadeiro quando o primeiro elemento é maior que o segundo elemento. Esta fórmula deve ser aplicada a dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="maior">
      <formula nome="soma">
            <base atributo=”consumo24”/>
            <const valor=”2”/>
      </formula>
      <propriedade nome=”n”/>
</formula>
```

- **maior_ou_igual**: Fórmula que retorna verdadeiro quando o primeiro elemento é maior ou igual ao segundo elemento. Esta fórmula deve ser aplicada a dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="maior_ou_igual">
      <formula nome="soma">
            <base atributo=”consumo24”/>
            <const valor=”2”/>
      </formula>
      <propriedade nome=”n”/>
</formula>
```

- **menor**: Fórmula que retorna verdadeiro quando o primeiro elemento é menor que o segundo elemento. Esta fórmula deve ser aplicada a dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="menor">
      <base atributo=”consumo24”/>
      <formula nome="multiplicacao">
            <propriedade nome=”k”/>
            <base atributo=”media_rota”/>
      </formula>
</formula>
```

- **menor_ou_igual**: Fórmula que retorna verdadeiro quando o primeiro elemento é menor ou igual ao segundo elemento. Esta fórmula deve ser aplicada a dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="menor_ou_igual">
      <formula nome="diferenca">
```

```
                <base atributo="consumo24"/>
                <propriedade nome="n"/>
            </formula>
            <const valor="10"/>
        </formula>
```

- **igual**: Fórmula que retorna verdadeiro quando o primeiro elemento é igual ao segundo elemento. Esta fórmula deve ser aplicada a dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="igual">
      <base atributo="resultado"/>
      <const valor="0"/>
</formula>
```

- **igualString**: Fórmula semelhante à "igual", mas que realiza a comparação considerando os valores como seqüência de caracteres (string). Esta fórmula deve ser aplicada a dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="igualString">
      <base atributo="consumo20"/>
      <const valor="NULL"/>
</formula>
```

- **diferente**: Fórmula que retorna verdadeiro quando o primeiro elemento é diferente do segundo elemento. Esta fórmula deve ser aplicada a dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="diferente">
      <base atributo="irreg24"/>
      <const valor="0"/>
</formula>
```

- **intervalo**: Fórmula que retorna verdadeiro quando o primeiro elemento é menor ou igual ao segundo elemento e o segundo elemento é menor ou igual ao terceiro. Esta fórmula deve ser aplicada a três elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="intervalo">
      <const valor="0"/>
      <base atributo="consumo24"/>
      <const valor="30"/>
</formula>
```

- **pertence**: Fórmula que retorna verdadeiro quando o primeiro elemento pertence ao conjunto de valores definido no segundo elemento. Esta fórmula deve ser aplicada a dois elementos. O primeiro pode ser elemento básico (propriedade, atributo da base e constante) ou fórmula numérica. O segundo elemento deve ser uma lista de valores (propriedade, lista de atributos ou lista de constantes). Exemplo:

```
<formula nome="pertence">
      <base atributo=”mes_ultimo_consumo”/>
      <propriedade nome=”r3”/>
</formula>
```

- **pertenceString**: Fórmula semelhante à “pertence”, mas que realiza a comparação considerando os valores como seqüência de caracteres (string). Esta fórmula deve ser aplicada a dois elementos. O primeiro pode ser elemento básico (propriedade, atributo da base e constante) ou fórmula numérica. O segundo elemento deve ser uma lista de valores (propriedade, lista de atributos ou lista de constantes). Exemplo:

```
<formula nome="pertenceString">
      <base atributo=”mes_ultimo_consumo”/>
      <const valor=”1,2,NULL”/>
</formula>
```

As fórmulas numéricas realizam operações que retornam algum valor numérico. No sistema foram definidas as seguintes fórmulas numéricas:

- **soma**: Calcula a soma entre dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome= "soma">
      <base atributo=”consumo24”/>
      <const valor=”3”/>
</formula>
```

- **Diferenca**: Calcula a subtração entre dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="diferenca">
      <base atributo=”consumo24”/>
      <formula nome="soma">
            <base atributo=”consumo24”/>
            <const valor=”3”/>
      </formula>
</formula>
```

- **multiplicacao**: Calcula a multiplicação entre dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="multiplicacao">
      <formula nome="soma">
            <base atributo=”consumo23”/>
            <const valor=”3”/>
      </formula>
      <const valor=”2”/>
</formula>
```

- **divisao**: Calcula a divisão real entre dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="divisao">
      <formula nome="soma">
            <base atributo=”consumo23”/>
            <const valor=”3”/>
      </formula>
      <formula nome="multiplicacao">
            <base atributo=”consumo23”/>
            <const valor=”3”/>
      </formula>
</formula>
```

- **div:** Calcula o quociente da divisão inteira entre dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Esta fórmula retorna o valor **q** da expressão **a = b*q + r**, onde **a** é o dividendo, **b** é o divisor, **q** é o quociente e **r** é o resto. Exemplo:

```
<formula nome="div">
      <base atributo=”consumo24”/>
      <formula nome="multiplicacao">
            <base atributo=”consumo23”/>
            <const valor=”3”/>
      </formula>
</formula>
```

- **mod**: Calcula o resto da divisão inteira entre dois elementos, que podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Esta fórmula retorna o valor **r** da expressão **a = b*q + r**, onde **a** é o dividendo, **b** é o divisor, **q** é o quociente e **r** é o resto. Exemplo:

```
<formula nome="mod">
      <base atributo=”consumo24”/>
      <const valor=”3”/>
</formula>
```

- **incrementa**: Soma o valor 1 (um) ao elemento, que pode ser um elemento básico (propriedade, atributo da base e constante) ou uma fórmula numérica. Exemplo:

```
<formula nome="incrementa">
      <base atributo=”consumo24”/>
</formula>
```

- **decrementa**: Subtrai o valor 1 (um) do elemento. Esse elemento pode ser elemento básico (propriedade, atributo da base e constante) ou uma fórmula numérica. Exemplo:

```
<formula nome="decrementa">
      <base atributo=”consumo24”/>
</formula>
```

- **absoluto**: Calcula o valor absoluto de um elemento, que pode ser um elemento básico (propriedade, atributo da base e constante) ou uma fórmula numérica. Exemplo:

```
<formula nome="absoluto">
      <formula nome="decrementa">
            <base atributo=”consumo24”/>
      </formula>
</formula>
```

- **negacao**: Calcula o valor com sinal inverso de um elemento, que pode ser um elemento básico (propriedade, atributo da base e constante) ou uma fórmula numérica. Exemplo:

```
<formula nome="negacao">
      <formula nome="absoluto">
            <formula nome="decrementa">
                  <base atributo=”consumo24”/>
            </formula>
      </formula>
</formula>
```

- **media**: Calcula a média de uma lista de valores, que pode ser uma propriedade (com uma lista de valores), uma constante (com uma lista de valores) ou uma lista de atributos da base. Exemplo:

```
<formula nome="media">
  <lstAtributos atributos=”consumo23,consumo22,consumo21”/>
</formula>
```

- **desvio_padrao**: Calcula o desvio padrão de uma lista de valores, que pode ser uma propriedade (com uma lista de valores), uma constante (com uma lista de valores) ou uma lista de atributos da base. Exemplo:

```
<formula nome="desvio_padrao">
  <lstAtributos atributos=”consumo23,consumo22,consumo21”/>
</formula>
```

- **condicional**: Retorna o segundo argumento se o primeiro argumento é verdadeiro. Senão, retorna o terceiro argumento. O primeiro argumento deve ser

uma fórmula condicional. O segundo e terceiro argumentos podem ser elementos básicos (propriedade, atributo da base e constante) ou fórmulas numéricas. Exemplo:

```
<formula nome="condicional">
      <formula nome="maior"/>
            <base atributo="consumo24"/>
            <const valor="30"/>
      </formula>
      <base atributo="consumo23"/>
      <base atributo="consumo24"/>
</formula>
```

- **subtraiMes**: Calcula a subtração, em meses, de um elemento, que pode ser um elemento básico (propriedade, atributo da base e constante) no formato 'YYYYMM'. Exemplo:

```
<formula nome="subtraiMes">
      <base atributo="ano_mes_ultimo_consumo"/>
      <propriedade nome="n"/>
</formula>
```

- **atribui**: Um outro tipo de fórmula definida é a de atribuição. Esta fórmula atribui um valor a uma variável. O primeiro argumento deve ser uma variável e o segundo argumento pode ser um elemento básico (propriedade, atributo da base e constante) ou uma fórmula numérica. Exemplo:

```
<formula nome="atribui">
      <variavel nome="cont"/>
      <formula nome="incrementa"/>
            <propriedade nome="cont"/>
      </formula>
</formula>
```

Bibliografia

(ANEEL, 2000) Resolução 456 de 29 de novembro de 2000, Condições Gerais de Fornecimento de Energia Elétrica, Agência Nacional de Energia Elétrica.

(Bicharra et al., 2003) Bicharra, A.C., Varejão, F.M. e Ferraz, I.N. Aquisição de Conhecimento, Sistemas Inteligentes, Capítulo 3, Rezende, S.O. (coordenadora), Ed. Manole, 2003.

(Bolton & Hand, 2002) Bolton, R., Hand, D.J., - Statistical Fraud Detection: A Review, 2002.

(Brause et al., 1999) Brause, R., Langsdorf, T., Hepp, M., Neural Data Mining for Credit Card Fraud Detection, J.W. Goethe-University exemplares de 2004.

(Burge et al., 1997) Burge, P., Taylor, J.S., Moreau, Y., Verrelst, H., Stoermann C. e Gosset, P., BRUTUS – A Hybrid Detection Tool, ACTS Mobile Summit 97, Proceedings of ACTS Mobile Summit 1997.

(CLIPS, 1985) "What is CLIPS?" em http://www.clips.com - última consulta realizada em 23/02/2006.

(Codi, 1997) Comitê de Distribuição de Energia Elétrica. Documento técnico CODI-08.05. Brasília, 1997

(Cometti et al., 2005) Cometti, E.S, ESCELSA, Varejão, F.M., UFES, "Melhoramento na identificação de perdas comerciais através da análise computacional inteligente do perfil de consumo e dos dados cadastrais de consumidores", Relatório técnico final do projeto de desenvolvimento e pesquisa da UFES em parceria com a ESCELSA, 2005.

(Cover & Hart, 1967) Cover, T. M., Hart, P. E., Nearest neighbour pattern classification. IEEE Transactions on Information Processing, IT-13:21-27, 1967.

(Drago, 2005) Drago, I. "Seleção de características no problema de identificação de perdas comerciais do sistema de energia elétrica", Universidade Federal do Espírito Santo, 2005. Monografia (curso de Engenharia de Computação).

(Eller, 2003) Eller, N. A., Arquitetura de informação para o gerenciamento de perdas comerciais de energia elétrica, Programa de Pós Graduação, Eng. da Produção, UFSC, 2003.

(Engels & Theusinger, 1998) Engels, R. e Theusinger, C. Using a Data Metric for Preprocessing Advice for Data Mining Applications, European Conference on Artificial Intelligence, ECAI 1998

(Fayyad et al., 1996) Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The KDD Process for Extracting Useful Knowledge from Volumes of Data. AI Magazine,1996.

(Haykin, 2001) Haykin, S., Redes Neurais Princípios e Prática, Ed. Bookman, 2ª Edição, 2001.

(jDREW, 2006) "Purpose of the jDREW project" http://www.jdrew.org/jDREWebsite/jDREW.html - última consulta realizada em 25/03/2006.

(Kou et al., 2004) Kou, Y., Lu, C.T., Sirwongwattana, S., Huang, Y.P., "Survey of Fraud Detection Techniques," Proceedings of the 2004 International Conference on Networking, Sensing, and Control, Taipei, Taiwan, March 21-23, 2004

(Lee & Sohn, 2003) Lee, J. K., Sohn, M. M. Enhanced Knowledge Management with eXtensible Rule Markup Language. Proceedings of the 36th Hawaii International Conference on System Sciences – 2003

(Loureiro et al., 2005) Loureiro, S. M., Margoto L. R., Varejão, F.M., Queiroga, R. M. "Um mecanismo automático para busca de parâmetros de técnicas de classificação utilizando algoritmos genéticos". ENIA, 2005.

(MCPT, 2004) Metodologia de Cálculo de Perdas Técnicas – TPPI – Gerência de Planejamento e Investimento – Escelsa.

(Mitchel, 1997) Mitchell, T. M., Machine Learning, MacGraw-Hill, 1997 ISBN 0-07-115467-1.

(Monard & Baranauskas, 2003) M. C. Monard, J. A. Baranauskas, "Conceitos sobre Aprendizado de Máquina", Sistemas Inteligentes, Capítulo 4, Rezende, S. O. (coordenadora) , Ed. Manole, 2003.

(Nogueira et al., 1996) NOGUEIRA, J.H.M., SILVA, R.B.A, ALCÂNTARA, J.F.L., ANDRADE, R.C. - Expert SINTA, uma Ferramenta Visual Geradora de Sistemas Especialistas, anais da VI Semana de Informática, Salvador, BA, 1996.

(Perim, 2005) Perim, G. T. "SAUIPE: Um Sistema Baseado em Conhecimento para Auxílio na Investigação de Perdas Elétricas". Universidade Federal do Espírito Santo, 2005. Monografia (curso de Ciência da Computação).

(Pressman, 2001) Pressman, R.S., "Software Engineering: A Practitioner's Approach", 5th Edition, New York: McGraw-Hill, 2001.

(Queiroga, 2005) Queiroga, R.M. "Uso de técnicas de data mining para detecção de fraudes em energia elétrica", Dissertação de mestrado em informática, UFES, 2005.

(Quinlan, 1986) Quinlan, J. R., Induction of decision trees, Machine Learning, 1986.

(Rezende et al., 2003) Rezende, S.O., Pugliese, J. B. e Varejão, F.M. Sistemas Baseados em Conhecimento, Sistemas Inteligentes, Capítulo 2, Rezende, S.O.(coordenadora), Editora Manole, 2003.

(Rijsbergen,1979) Rijsbergen, C.J. v. Information Retrieve. 2 ed. London: Butterworth, 1979.

(Rosset et al.,1999) Rosset, S., Murad, U., Neumann, E., Idan, Yizhak e Pinkas G. "Discovery of Fraud Rules for Telecommunications -Challenges and Solutions"

(RuleML, 2006) "The Rule Markup initiative" http://www.ruleml.org/ - última consulta realizada em 29/03/2006.

(Stefik, 1995) Stefik, M. - Introduction to Knowledge Systems. Ed Morgan Kaufmann, 1995.

(Wrightt, 1996) Wrightt, Peggy "Knowledge Discovery Preprocessing: Determining Record Usability" , ACM, 1996.