Gerência Proativa de Redes com Técnicas de Inteligência Artificial

Fernando Augusto da Silva Cruz
cruz@lrg.ufsc.br

Mirela Sechi Moretti Annoni Notare
Mirela@lrg.ufsc.br

Bernardo Gonçalves Riso
riso@lrg.ufsc.br

João Bosco da Motta Alves
jbosco@inf.ufsc.br

Carlos Becker Westphall
westphal@lrg.ufsc.br

Universidade Federal de Santa Catarina (UFSC)
Centro Tecnológico (CTC)Departamento de Informática e de Estatística (INE)
Curso de Pós Graduação em Ciência da Computação (CPGCC)
Laboratório de Redes e Gerência (LRG)
P.O.Box 476 - Florianópolis, SC, Brazil - ZIP 88040-970, +55(48) 331-9739, fax +55(48)331-9770, www.lrg.ufsc.br

Resumo:. Este trabalho apresenta o projeto de um sistema de gerenciamento inteligente na área de performance. O sistema emprega **técnicas de reconhecimento de padrões**) e sugere soluções (utilizando técnicas de **raciocínio baseado em casos**) antes que haja qualquer degradação na performance de um sistema distribuído. O projeto do sistema envolve a especificação e validação formal (utilizando **técnicas de descrição formal**) e uma modelagem (utilizando **técnicas de orientação a objetos**) a fim de assegurar um sistema correto e exato.

Palavras-Chave : Gerenciamento de Performance de Redes, Orientação para Objetos, Técnicas de Descrição Formal, Reconhecimento de Padrões, Raciocínio Baseado em Casos.

1. Introdução

Atualmente a alta taxa de crescimento das redes de telecomunicações heterogêneas, tanto em tamanho como em complexidade, exige o uso de funções de gerenciamento mais poderosas, tanto aquelas associadas aos equipamentos bem como aos serviços oferecidos. Este trabalho tem por objetivo descrever o projeto de um sistema de gerenciamento na área de performance, com suporte a distribuição através da tecnologia CORBA. Este projeto trata de um sistema de gerenciamento inteligente baseado em uma abordagem cliente – servidor. Os clientes são os gerentes e os servidores são os agentes inteligentes proativos.

Um sistema Proativo inteligente significa que ações são executadas quando o estado atual das informações obtidas são comparadas com as informações contidas em uma Baseline (valores de referência), para identificar a existência de possíveis problemas antecipando-se a qualquer degradação de performance do sistema [1]. O sistema de grenciamento é composto de um gerente, agentes, MOs (Objetos Gernciados) e o módulo de distribuição (CORBA). Cada MO está relacionado a um recurso real. O conjunto de todos MOs constituem a MIB (Management Information Base) a qual possui as informações atuais de gerenciamento dos recuros a serem gerenciados. O agente (Proativo Inteligente) foi modelado como sendo uma composição da MIB, uma vez que o comportamento que o agente desmpenha é distribuído e executado nos MOs.

O agente Proativo Inteligente obtém as informações atuais do sistema (MOs) e compara-as com os valores de referência (Baseline). Os valores de referência são os valores dos parâmetros relacionados com a operação normal da rede que foram previamente estabelecidos pelo usuário. A abordagem proativa caracteriza-se pelo fato do sistema (através do agente) reconhecer um problema utilizando técnicas de inteligência artificial e tomar atitudes proativas para soluciona-lo, configurando uma aplicação de gerenciamento proativa para a prevenção de problemas em redes de computadores. O Agente Proativo Inteligente é programado para notificar ao sistema de gerenciamento sobre qualquer parâmetro que indique uma diminuição na performance da rede. Esta programação é baseada em técnicas de inteligência artificial.

Devido as características das redes de telecomunicações – heterogenêas e distribuídas – foi tomado como base alguns parâmetros de uma rede ATM para serem utilizados durante a simulação. Um objeto de controle de conexão foi projetado para desempenhar as tarefas de estabelecimento e liberação de uma

conexão requisitada entre dois pontos terminais distintos. Portanto, algumas informações relevantes descrevendo os pontos de origem e destino da conexão, bem como os identificadores para Canal Virtual e Caminho Virtual são requisitados. Informação adicional que diz respeito ao número do canal virtual garantirá um estabelecimento apropriado da conexão. Este sistema engloba o gerenciamento da interface Node-to-Network (NNI) em uma rede ATM [2, 3. 4].

A fim de realizar o gerenciamento, antes de qualquer requisição ou notificação, um estabelecimento de associação deve ser feito. Após o término do gerenciamento a associação é liberada. Este controle de Associação é feito pela arquitetura CORBA. A ISO e a comunidade de desenvolvedores da Internet conceberam padrões de gerenciamento baseado em protocolos, adotando o modelo gerente-agente que regula as interações entre aplicações de gerenciamento. O modelo gerente–agente não é inteiramente orientado a objetos. A arquitetura CORBA por sua vez pode ser vista como uma contrapartida pragmática da ODP (Open Distriduted Processing), em que projeta um paradigma orientado a objeto e endereça principalmente aspectos de interface programática, utilizando uma RPC (Remote Procedure Call) de propósito geral. CORBA é uma tecnologia de gerenciamento de objetos distribuídos. Estabelece um quadro básico para computação distribuída. Muitos serviços de objetos comuns são definidos em CORBA, os quais dão suporte aos pedidos de plataformas e aplicações distribuídas. Uma instância de objeto CORBA pode ser endereçado pelo tipo através de uma ORB, de maneira transparente a localização do objeto. A ORB irá encontrar uma instância daquele tipo e retorna uma referência ao objeto cliente. Instâncias do mesmo tipo podem ser distinguidas através de servidores de nomes ou negociadores.

Este trabalho está estruturado da seguinte forma. Seção dois apresenta o Projeto e Modelagem Orientada a Objeto. Seção três aborda o uso de Métodos Formais para especificação e validação. Seção quatro descreve o uso de Reconhecimento de Padrões para classificar os valores de performance. Seção cinco apresenta Simulações das Implementações utilizando técnicas de IA. Seção seis apresenta as Conclusões e Futuros Trabalhos. Finalmente, a seção sete apresenta as Referências Bibliográficas.

2. Projeto e Modelagem Orientada a Objetos

O projeto orientado a objetos deste sistema de gerenciamento proativo inteligente envolve três tipos de modelagens: estática, dinâmica e funcional [5, 6, 7]. O projeto foi desenvolvido segundo o modelo OMT (Object Modelling Technique).

O sistema, basicamente, engloba três módulos principais:
(1) um gerente (que recebe as notificações dos agentes);
(2) um agente proativo inteligente (que fica monitorando os parâmetros da rede e comparando-os com os parâmetros/limiares de uma baseline a fim de enviar as notificações de tendências de degradação ao gerente); e
(3) um canal de interface para a comunicação (utiliza-se suporte a distribuição CORBA e a funcionalidade de comunicação ATM).

Veja a Figura 2.1.

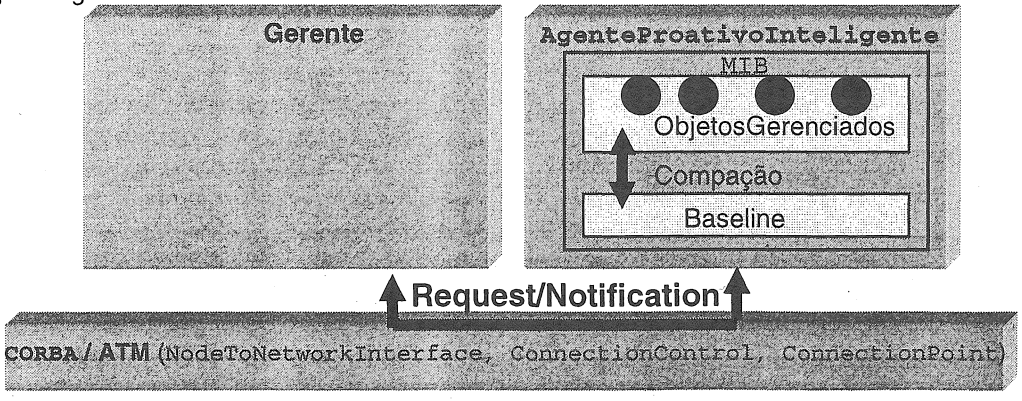

Figura 2.1 - Classes do Sistema de Gerenciamento Pró-ativo.

Apesar do fato de que CORBA ser um barramento para comunicação entre objetos, nós vamos considera-lo neste modelo como um objeto intermediário entre gerente e agente. Este trabalho não apresenta maiores detalhes em relação a modelagem orientada a objeto, pois não é este o foco principal do trabalho. A modelagem completa pode ser obtida por requisição aos autores.

3. Especificação Formal e Validação do Agente Proativo

Usando técnicas de descrição formal (LOTOS, neste caso) [8, 9, 10] é possível obter a prova correta das especificações usando verificação de equivalência. A verificação da equivalência é feita através da interface da ferramenta Caesar com a ferramenta Aldébaran. A utilização da ferramenta Caesar permite a geração do automata no formato Aldébaran. O uso do Aldébaran possibilita alguém a provar que ambos automatas do agente – o automato mais abstrato (proa) e o automato refinado (proagent) – são equivalentes observacionalmente. Isto significa que eles são indistinguíveis quando observados externamente.

Em um nível mais alto de abstração o Agente é descrito abaixo (tanto a especificação como o automato).

```
specification proag[notif,req,baseline_data,mo_notif,operat]:noexit
behaviour
proa[notif,req,baseline_data,mo_notif,operat]
where
process proa[notif,req,baseline_data,mo_notif,operat]:noexit:=
       operat; baseline_data;    (req;proa[notif,req,baseline_data,mo_notif,operat]
                                 [] notif;proa[notif,req,baseline_data,mo_notif,operat])
    [] mo_notif; baseline_data; (req;proa[notif,req,baseline_data,mo_notif,operat]
                                 [] notif;proa[notif,req,baseline_data,mo_notif,operat])
endproc
endspec
```

```
des (0, 8, 5)          (1, BASELINE_DATA,     (2, BASELINE_DATA,     (3, NOTIF, 0)
(0, OPERAT, 1)         3)                     4)                     (4, REQ, 0)
(0, MO_NOTIF, 2)                              (3, REQ, 0)            (4, NOTIF, 0
```

Em um nível de abstração mais refinado o Agente Proativo é descrito abaixo (tanto a especificação como o automato).

```
specification proagent[notif,req,baseline_data,mo_notif,operat]:noexit
behaviour
hide calculated_data, colected_data in
        verify[notif,req,baseline_data,calculated_data,colected_data]
        |[colected_data]|
        remote_monitor[colected_data,mo_notif,operat]
        where
        process verify[notif,req,baseline_data,calculated_data,colected_data]:noexit:=
                calculate[calculated_data,colected_data]
                |[calculated_data]|
                comparison[notif,req,baseline_data,calculated_data]
                where
                process calculate[calculated_data,colected_data]:noexit:=
                        colected_data;calculated_data;calculate[calculated_data,colected_data]
                endproc
                process comparison[notif,req,baseline_data,calculated_data]:noexit:=

        calculated_data;baseline_data;notif;comparison[notif,req,baseline_data,calculated_data]
                []
        calculated_data;baseline_data;req;comparison[notif,req,baseline_data,calculated_data]
                []  calculated_data;baseline_data;comparison[notif,req,baseline_data,calculated_data]
                endproc
        endproc
        process remote_monitor[colected_data,mo_notif,operat]:noexit:=
                operat;colected_data;remote_monitor[colected_data,mo_notif,operat]
            []  mo_notif;colected_data;remote_monitor[colected_data,mo_notif,operat]
        endproc
endspec
```

```
des (0,45,18)
(0,MO_NOTIF,13
)
(0,OPERAT,13)
(1,BASELINE_DA
TA,3)
(2,MO_NOTIF,5)
(2,OPERAT,5)
(2,NOTIF,13)
(3,REQ,11)
(4,MO_NOTIF,3)
(4,OPERAT,3)
(4,REQ,13)
(5,NOTIF,11)
(6,MO_NOTIF,1)
(6,OPERAT,1)
(6,BASELINE_DA
TA,4)
(7,MO_NOTIF,2)
(7,OPERAT,2)
(7,NOTIF,0)
(8,MO_NOTIF,4)
(8,OPERAT,4)
(8,REQ,0)
(9,MO_NOTIF,6)
(9,OPERAT,6)
(9,BASELINE_DA
TA,8)
(10,BASELINE_D
ATA,11)
(11,i,6)
(11,i,12)
(11,i,15)
(12,MO_NOTIF,1
0)
(12,OPERAT,10)
(12,BASELINE_D
ATA,13)
(13,i,9)
(13,i,14)
(13,i,16)
(13,MO_NOTIF,1
1)
(13,OPERAT,11)
(14,MO_NOTIF,1
2)
(14,OPERAT,12)
(14,BASELINE_D
ATA,0)
(15,MO_NOTIF,1
7)
(15,OPERAT,17)
(15,BASELINE_D
ATA,2)
(16,MO_NOTIF,1
5)
(16,OPERAT,15)
(16,BASELINE_D
ATA,7)
(17,BASELINE_D
ATA,5)
```

Esta geração dos automatos pode ser feita também através de uma interface gráfica. Utilizando-se para isso o software Eucalyptyus.

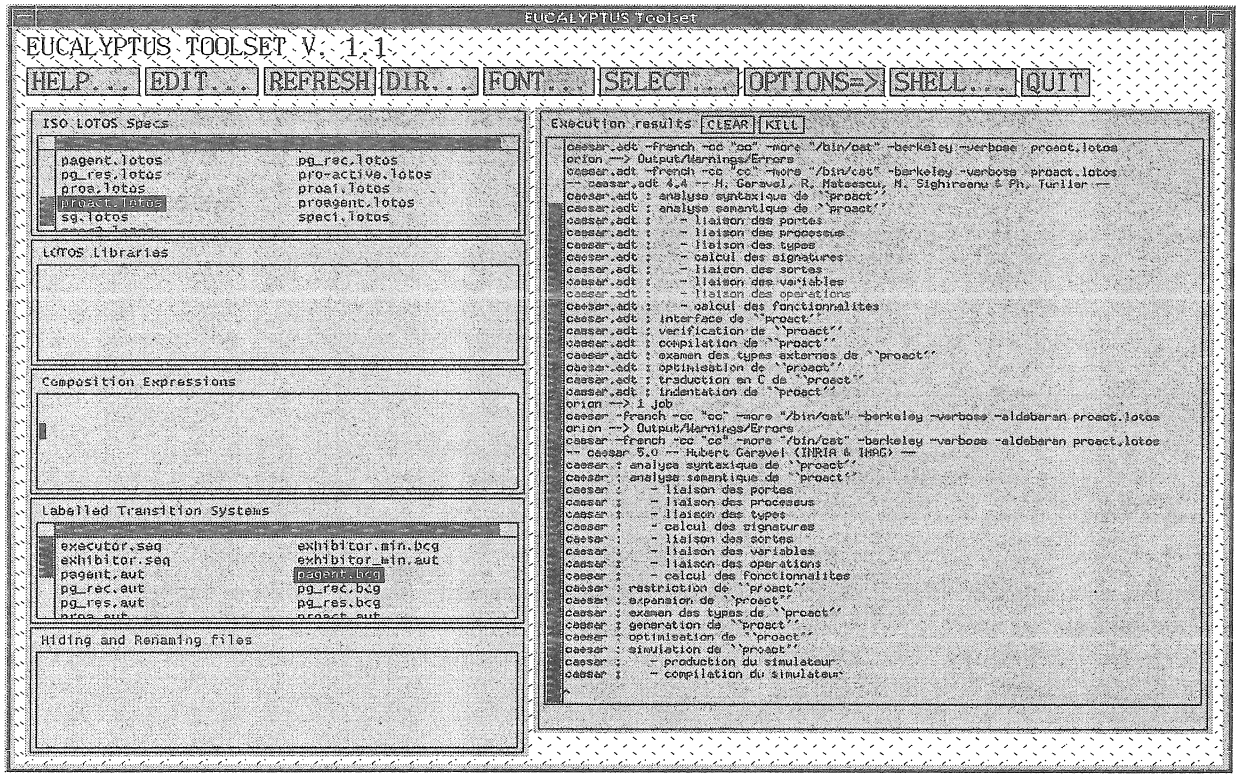

Figura 3.1 – Geração do autômato do agente no seu mais alto nível de abstração.

A Figura 3.1 mostra a geração do automato a partir das especificações do agente, em um nível de abstração mais alto.

Verificação da equivalência observacional entre ambas as especificações:

```
aldebaran -oequ proa proagent
TRUE
```

O resultado 'TRUE' mostra que os automatos (associados às especificações mais abstrata e mais refinada) são equivalentes quanto a observação. Esta verificação é a prova formal que uma espeficicação correta foi produzida. Outros tipos de validação, tais como simulações e testes apenas encontram erros, mas não provam a corretude.

4. Reconhecimento de Padrão para Avaliação de Performance

A fim de definir os parâmetros de operação da rede (perfil) que conduzem a uma boa performance, e os parâmetros da rede com valores que proporcionam um péssimo desempenho (isto é, a determinação dos valores dos parâmetros da rede que estão num limiar que conduzem a rede a operar nas piores condições) nós utilizamos técnicas de reconhecimento de padrões para classificar a

rede, utilizando alguns de seus atributos relevantes que influenciam a performance, ou seja, traçamos o perfil da rede.

A rede foi clasiificada em três tipos de performance: (c1) boa performance; (c2) performance no limite da degradação; e (c3) performance degradada. Neste trabalho nós consideramos somente dois parâmetros para realizar esta avaliação: x1 = `BurstTolerance`; e x2 = `PeakCellRate`. A base de dados utilizada contém 50 observações (n_1=n_2=n_3=50). Estas observações são supostas medidas dos valores destes parâmetros da rede. Para o conjunto de treinamento nós utilizamos as primeiras 35 observações e as outras 15 observações nós utilizamos para testar o classificador. Para implementação dos algoritmos que perfazem as técnicas de reconhecimento de padrões foi utilizada a Função de Fisher e a Distância Euclidiana [12]. O software MATLAB foi utilizado para implementar o algoritmo de classificação.

O método de Fisher consiste em transformar observações multi-variadas em variáveis simples, ou seja, é capaz de obter uma combinação linear que corresponde as caracteristicas dos padrões. A idéia de Fisher é transformar observações multi-variada X em observações de variável simples Y, até que os Y's obtidos das amostras das populações sejam separadas o máximo possível. Se nós considerarmos que m_1 é a média dos Y_s obtidos a partir dos X's de n_1 populações and m_2 a média dos Ys obtidos pelos Xs das n_2 populações, então Fisher seleciona as combinações lineares que maximiza a distância entre m_1 e m_2, relacionanda as variâncias de Ys. Detalhes de implementação podem ser visto no Anexo **A.**

Os resultados obtidos no processo de classificação podem ser considerados satisfatórios pois a taxa de erro encontrada (de aproximadamente 5%) justifica nossa proposta do emprego desta técnica/algoritmo como forma de gerenciar proativamente a performance de uma rede.

5. Simulações Utilizando Técnicas de IA

Nesta simulação do Sistema Inteligente de Gerênciamento de Performance o software Kappa e o ESTEEM foram utilizados. Enquanto o primeiro é utilizado para identificação do problema, o segundo é utilizado para sugerir uma solução ao problema detectado (estas operações são o que caracterizam uma atitude proativa).

O Kappa é um shell usado para o desenvolvimento de sistemas especialistas, cujo o funcionamento é baseado em regras de produções e *frames*. Desta maneira , não foi necessário implementar a máquina de inferência, uma vez que já se encontra embutida no próprio shell. O ESTEEM por sua vez é um shell comumente utilizado para testar desenvolvimento de sistemas inteligentes tendo como suporte "Raciocínio Baseado em Casos".

5.1 Identificação do Problema através do software KAPPA

Para a implementação do sistema de gerenciamento proativo para redes ATM, além das classes que o Kappa oferece quando um novo sistema é inicializado, duas novas classes foram criadas: Gerente e Agente Inteligente. O Agent1 é uma instância da classe Agente. Nós ressaltamos que neste trabalho somente três objetos gerenciados foram implementados para demonstar a funcionalidade do sistema (Agente Inteligente). Para adicionar novos objetos é uma tarefa simples, necessita-se somente adicionar um novo *slots* para a Classe Agente.

O sistema de gerenciamento apresentado neste trabalho é um sistema híbrido, no qual foram utilizados: (1) *frames slots* e para representar o Conhecimento; e (2) regras de produção para suportar raciocínio sobre a base de conhecimento. O uso de *frames* e *slots* pelo Kappa é adequado devido a similariedade com orientação baseada em objetos. Entretanto, o uso de regras de produção é adequado devido a possibilidade de implementar raciocínio para frente bem como raciocínio para trás. Veja as Figuras 5.1 e 5.2.

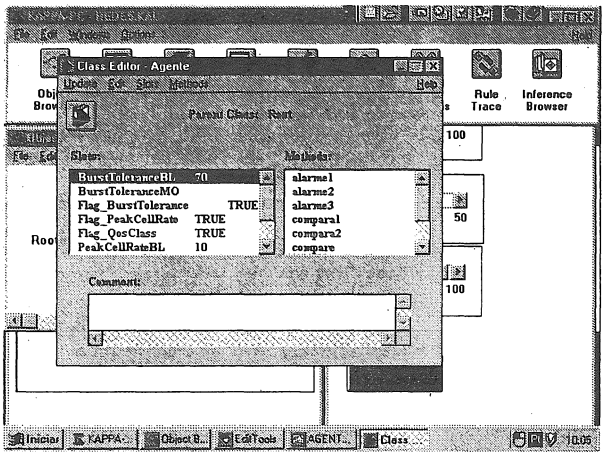

Figura 5.1 – Uso de frames e slots.

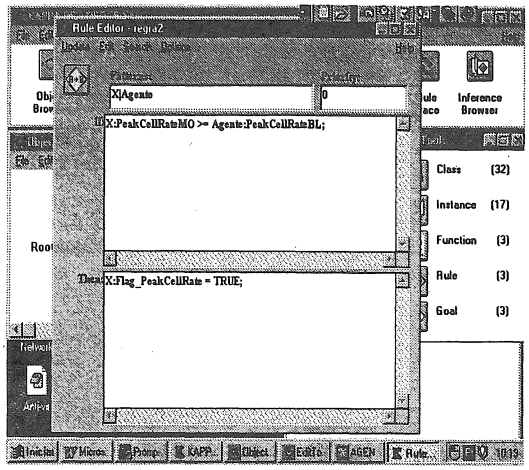

Figura 5.2 - Forward X Backward.

Geralmente o raciocínio para frente é mais adequadamente empregado na solução de problemas associados com simulação, enquanto raciocínio para trás é mais adequado para resolver problemas associados com diagnóstico.

Veja na Figura 5.3 o funcionamento do sistema. Esta Figura apresenta o agente inteligente monitorando os valores de Rajadas, Taxa de Pico de Células e `Qualidade de Serviço`, verificando a tendência para a degradação da rede. Desta maneira, o gerente recebe os respectivos alarmes (Alerta Tolerância a Rajadas, Alerta Pico Máximo de `Células` em uma Conexão, Alerta `Qualidade de Serviço`) e mostra os alarmes de "Existência de Problemas em uma Rede" com a informação "Rede com Problemas" para o gerente da rede. Os objetos apresentados neste trabalho estão em conformidade com a norma M3.100 do modelo TMN definida pelo ITU-T (International Telecommunication Union - Telecommunications). Os valores limiares incluídos nas regras são os valores obtidos pela classificação feita através de técnicas de reconhecimento de padrões.

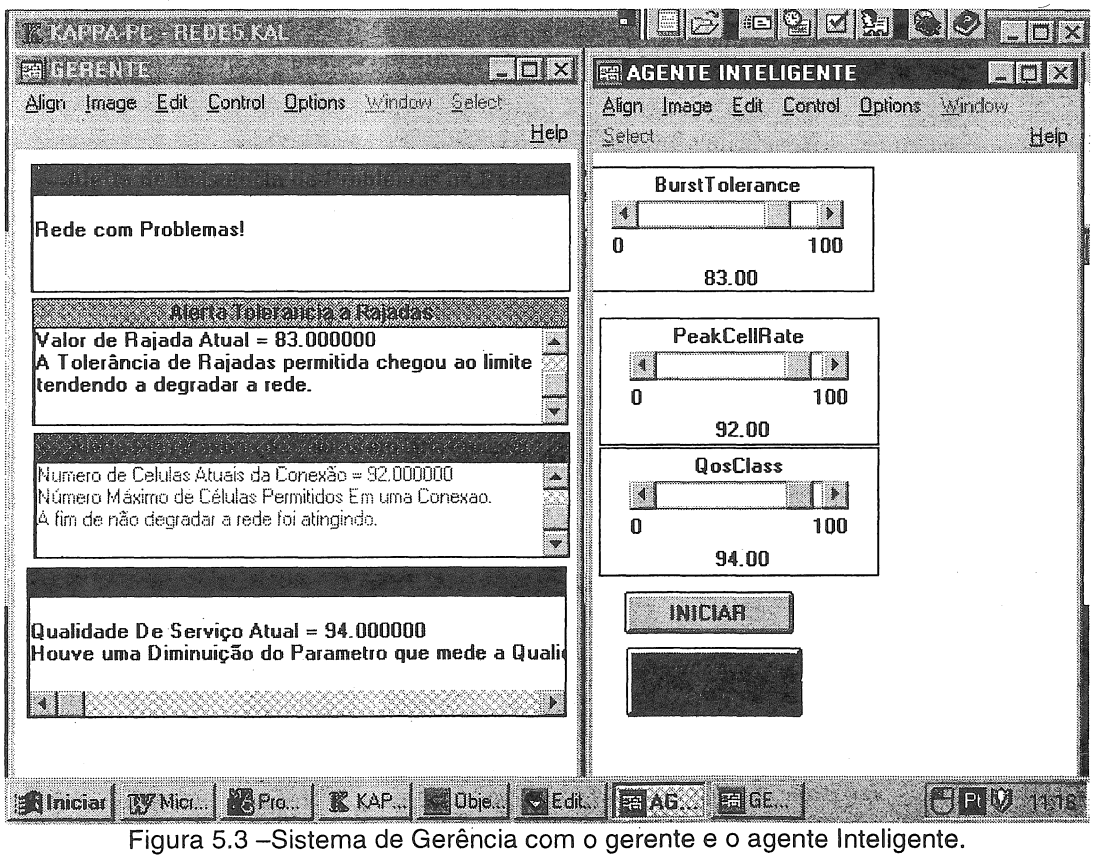

Figura 5.3 –Sistema de Gerência com o gerente e o agente Inteligente.

Mais especificamente, o funcionamento do sistema é descrito a seguir:

(a) O agent inteligente é um agente remoto que apresenta um comportamento proativo monitorando os três objetos gerenciados, os quais estão representados na Figura 5.3 nos três slides `BurstTolerance` (existe um limite máximo, que quando excedido dispara um alarme, indicando uma tendência de degradação da rede), `PeakCellRate` (existe um certo valor de números de pacotes, que quando alcançado aciona o correspondente alarme, indicando uma tendência de degradação da rede) e `QosClass` (existe um número limite de `QosClass`, que quando alcançado dispara o correspondente alarme);

(b) Cada vez que um dos valores dos objetos é modificado as regras são acionadas. Nos eventos em que as regras verificam a existência de uma tendência em direção a dedradação (um valor foi alcançado) um flag é estabelecido. De acordo com os flags (`Flag_BurstTolerance`, `Flag_PeakCellRate`, `Flag_QosClass`), um procedimento é executado de modo que os alarmes possam ser acionados;

(c) O Gerente , quando recebe os alarmes, com seus respectivos valores de objetos, mostra o alarme com a informação de que a rede tem problemas. Este alarme é uma sugestão, que o administrador pode usar para tomar as providências necessárias, portanto previnindo a rede de ser afetada por um processo de ruptura ou degradação;

(d) O botão `BEGIN` serve para estabelecer os valores das variavéis que corresponde aos objetos gerenciados;

(e) O botão INICIAR serve para inicializarlizar os Scripts e para limpar as áreas onde as mensagens são mostradas na tela, portanto indicando o início de um novo processo de monitoração.

Apesar de gerente e agente estarem sendo mostrados em uma mesma tela neste trabalho, um sistema distribuído não apresentaria esta configuração.

5.2 Solução do Problema através do software ESTEEM

Nesta seção, Técnicas de Raciocínio Basedo em Casos são utilizadas para resolver problemas de performance, baseando-se em problemas que ocorreram em situações similares. Após a detecção e identificação de uma possível degradação do sistema através do software KAPPA (o problema é encontrado), então o software ESTEEM é usado para encontrar uma solução a partir de situações anteriormente ocorridas.

O problema a ser resolvido, o caso alvo, é descrito através de atributos relevantes os quais são comparados com casos que foram problemáticos e resolvidos pelo administrador da rede (estes casos constituem a especificação da Base de Casos). Deste modo, o ESTEEM recebe automaticamente do KAPPA o problema identificado, na forma de um caso alvo. O ESTEEM procura em sua Base de Casos os casos mais similares segundo uma métrica de similaridade estabelecida. Como resultado desta pesquisa, obtém-se as soluções dadas aos problemas similares ao caso alvo, e portanto, obtem-se sugestões de como resolve-lo.

A métrica de similaridade é realizada comparando-se os atributos do caso alvo com os atributos dos casos que compõe a Base de Casos. O procedimento é realizado usando regras do tipo If-Then. Após a recuperação de todos os casos similares, onde o grau de similariedade é dado pelos atributos e sua relevância (pesos) é necessário escolher o caso mais similar. Veja a Figura 5.2.

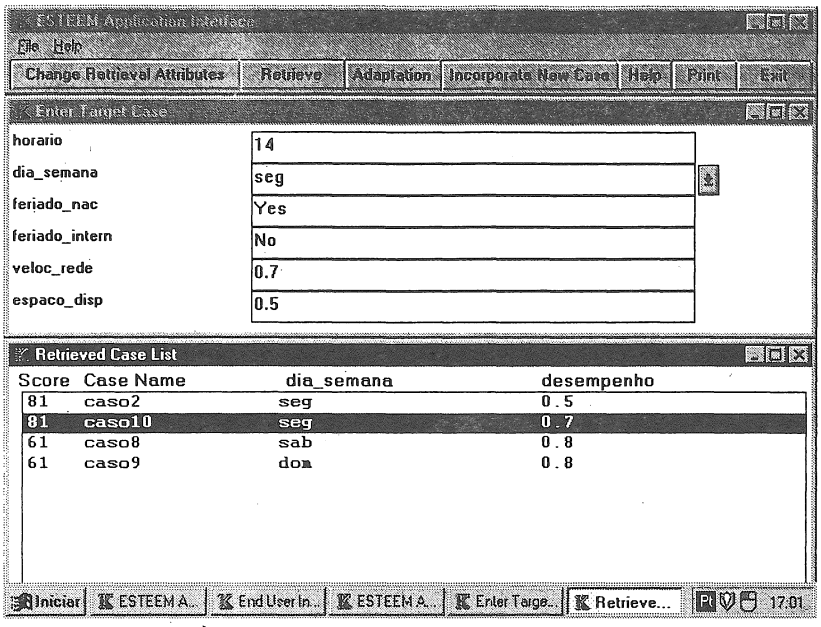

Figura 5. 2 – Recuperação de Casos Utilizando a Técnica de Raciocínio Baseado em Casos.

Algumas vezes, existe a necessidade de realizar adaptações de um problema específico que não esteja contemplado na Base de Casos. Por exemplo, se algum tipo de degradação da rede ocorre em uma Segunda-feira, às 14h (geralmente um horário de pico), mas em uma Segunda-feira em particular é feriado, então existe a necessidade de uma adaptação. Tal caso adaptado torna-se um caso adicional a ser inserido na Base de Casos e constitui a principal fonte de aprendizagem do sistema proposto.

6. Conclusões

Este trabalho propõe uma solução para melhorar a performance de gerenciamento de redes de alta velocidade (ATM), dentro da filosofia de gerenciamento proativo. O gerenciamento proativo é caracterizado por uma reação automática do sistema quando uma tendência de degradação é percebida dentro da rede, onde uma mera solução reativa para um dado problema não é suficiente para uma boa performance.

Nós concluimos que o uso da abordagem orientada a objetos e métodos formais foi muito útil para descrever o sistema de uma maneira segura e eficiente. Estes métodos permitem modificações rápidas e melhoramentos (através por exemplo do uso do software WithClass), bem como validações rigorosas (utilizando software Eucalyptus). O uso de Técnicas de Inteligência Artificial contribuiu bastante para o desenvolvimento do sistema. Técnicas de Reconhecimento de Padões permite uma classificação eficiente dos parâmetos considerados para avaliação de performance (usando o software MatLab). Regras de Produção e técnicas de Raciocínio Basedo em Casos permitem a implementação de simulações importantes do sistema (utilizando as ferramentas de software KAPPA e ESTEEM). Os resultados completos, tabelas e diagramas estão disponíveis através de pedidos diretamente aos autores.

A principal contribuição deste trabalho é apresentar um projeto completo, bem como um protótipo validado de um sistema de gerenciamento de performance. O projeto começa com uma especificação e validação rigorosa bem como projeto e modelagem detalhada utilizando uma metodologia atual baseada na abordagem de orintação a objetos. Após isto, o projeto usa uma técnica confiável baseada em cálculos de reconhecimento de padrões para definir o limite entre boa e péssima performance. Para concluir, o projeto utiliza Técnicas de Racicínio Baseado em Casos para sugerir uma solução automática do problema. Em resumo, o projeto envolve desde a especificação formal até a implementação final que inclui uma sugestão para a solução do problema. Os autores acreditam que a característica principal, comparado com produtos comerciais na área de gerenciamento de sistema (que podem utilizar tabelas de conf
iguração, como por exemplo. HP IT/O, Tivoli, BMC Patrol), utiliza algoritmos up-to-date para definir os limiares de uma maneira mais automática para --alertar aos clientes (bem como descobrir problemas, sugerir soluções para o sistema empregando técnicas de racicínio baseado em casos).

Futuros trabalhos, novos algoritmos serão utilizados para implementar as técnicas de reconhecimento de padrões. Atualmente estudos estão sendo feitos com o objetivo de se utilizar uma função de base radial para implementar uma rede neural sendo que o algoritmo de treinamento da rede a ser utilizado será o, "Orthogonal Least Squares". Dados mais completos e reais também serão utilizados. O uso de novo hardware ATM nas simulações em breve será possivel em virtude de estar em fase de implantação a rede ATM da UFSC. E a partir desta implantação, dados reais serão utilizados, bem como, a tradução do protótipo para código C++ para uma futura integração com a plataforma de gerência OSIMIS, disponíivel atualmente em nosso laboratório.

7. Referências

[1] M.A ROCHA, C.B. WESTPHALL, Proactive Management to Computer Networks Using Agents and Artificial Intelligence Techniques, in Proceedings of the *Fifth IFIP/IEEE International Symposium on Integrated Network Management*, San Diego, California, USA, 05, 1997.

[2] G. CHEN, J. RIXON, Q. KONG, Integration CORBA and Java for ATM Connection Management. *DSOM'97*. Sydney, Australia, 1997, pp. 104-117.

[3] C. G. OMIDYAR et al. Introduction to Mobile and Wireless ATM. *IEEE Communications Magazine*. Nov. 1997, Vol 35, N 11, pp 30-122.

[4] L.F. KORMANN, K.O. ROGERIO, C.B. WESTPHALL, A TMN Network Element for the ConFiguration Management of ATM Networks: Development Aspects. *SEMISH 97.* Brasilia, 02-08/08/97.

[5] G. CHEN, Q. KONG, The Business Process and Object Modelling for Service Ordering. *DSOM'97*. Sydney, Australia, pp. 104-117.

[6] C. ERNST, *Object-Oriented Modelling and Design.* Class notes of UFSC/EPS 365919. 1997.

[7] G. PAVLOU, From Protocol-based to Distributed Object-based Management Architectures. *DSOM 197*. Sydney, Australia, 1997, pp. 25-40.

[8] E. BRINKSMA, *ISO 8807* – LOTOS – Language of Temporal Ordering Specification, 1988.

[9] M. S. M. A. NOTARE, B. G. RISO, P. S. LORENA, M. C. De O. PENNA, C. B. WESTPHALL, Formal Design of a Telecommunications Networks Management System. AT&T, *IEEE ISCC'97 – International Symposium on Computers and Communications*, Alexandria, Egypt, 07/1997.

[10] M. S. M. A. NOTARE, B. G. RISO, P. S. LORENA, M. C. De O. PENNA, C. B. WESTPHALL, Formal Design of a Platform for Telecommunication Heterogeneous Networks Management. University of Western Sydney – Department of Computing, Nepean, Australia, *DSOM'97 – 8th IFIP/IEEE International Workshop for Distributed Systems Operations and Management*, Sydney, Australia, 21-23/10/1997.

[11] *MATLAB: User's Guide* – For Microsoft Windows: High-Performance Numeric Computation and Visualisation Software. The Math Works Inc., Prentice-Hall, Englewood Cliffs, NJ, 1992.

[12] R.O. DUDA, P.E. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

ANEXO A – Implementação em MATLAB do Classificador com base no Método de Fischer

Se dois conjuntos de padrões são π_1 e π_2, então é possível definir
$\mu 1$ = valor experado da observation multi-variada de π_1
$\mu 2$ = valor experado da observation multi-variada de π_2

Considerando a matrix covariancia $\Sigma = \mathbb{E}(X - \mu_i)(X - \mu_i)'$ i=1,2 igual para ambos os padrões.

Considerando a combinação linear:

$$\frac{Y}{(1x1)} = \frac{l'}{(1xp)} \frac{X}{(px1)}$$

nós encontramos que $\mu_{1Y} = \mathbb{E}(Y \mid \pi_1) = \mathbb{E}(l'X \mid \pi 1) = l'\mu_1$ ou $\mu_{2Y} = \mathbb{E}(Y \mid \pi_2) = \mathbb{E}(l'X \mid \pi_2) = l'\mu_2$

e a variancia relacionanda, $\sigma_Y^2 = Var(l'X) = l'Cov(X)l = l'\Sigma l$ igual para todas as populações.

A melhor combinação linear é obtida da seguinte maneira:

$$\frac{\text{Distância quadrada Da média de Y}}{\text{variância de Y}} = \frac{(\mu_{1y} - \mu_{2y})^2}{\sigma^2_y} = \frac{(l'\mu_{1y} - l'\mu_{2y})^2}{l\sum l} = \frac{l'(\mu_1 - \mu_2)(\mu_1 - \mu_2)l'}{l'\sum l} = \frac{(l'\delta)^2}{l'\sum l}$$

onde $\delta = (\mu_1 - \mu_2)$ é a diferença entre os vetores média.

Portanto, os coeficientes da combination linear de Fisher $l' = \lfloor l_1, l_2, ..., l_p \rfloor$ são aqueles que maximizam a razão acima.

Considere $\delta = \mu_1 - \mu_2$ e $Y = l'X$, então :

Distance quadrada

$$\frac{\text{Da média de Y}}{\text{variância de Y}} = \frac{(l'\delta)^2}{l'\sum l}$$ é maximizada pela escolha de $l = c\sum^{-1}(\mu_1 - \mu_2)$ para $c \neq 0$.

Escolhendo c = 1 a combinação linear gerada é : $Y = l'X = (\mu_1 - \mu_2)\sum^{-1} X$

A qual é a Função Discriminante Linear de Fisher .

A razão máxima é dada por : $\frac{\max(l'\delta)^2}{l} = \frac{\delta'\Sigma^{-1}\delta}{l'\sum l}$

As formulações acima são utilizadas quando a média dos padrões e a matriz covariância são conhecidas. De outra forma, matriz covariância amostra (Spooled), dada por:

$$S_{pooled} = \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{(n_1 + n_2 - 2)}$$

é utilizada para obter Função Discriminante Linear de Fisher por amostragem:

$$\bar{Y} = \bar{l}'\bar{X} = (X_1 - X_2)' S_{pooled}^{-1} X$$

Neste trabalho , uma extensão deste método de Fisher (mostrada acima para duas populações) é utlizada, i.e., o método de Fisher para várias populações.

Seja g padrões, μ_i a média dos padrões, $\bar{\mu}$ o vetor média dos padrões combinada e Bo a matriz entre grupos de padrões:

$$B_0 = \sum_{i=1}^{g} \Sigma\left(\mu_1 - \bar{\mu}\right)\left(\mu_1 - \bar{\mu}\right)' \quad \text{onde} \quad \bar{\mu} = \frac{1}{g \sum_{i=1}^{g} \mu_1}$$

O conjunto de autovalores não iguais a zero $\lambda 1 >= \lambda 2, >= \lambda 3 >= ... >= \lambda s > 0$ de Σ^{-1} B0,

Que satisfazem $s <= \min (g-1), p)$ e os autovetores relacionados (normalisado) (e1, e2, ..., es),

Então o vetor de coeficientes l que maximiza a razão

$$\frac{l' B_0 l}{l' \sum_{i=1}^{g} l} = \frac{l'\left[\Sigma\left(\mu_1 - \bar{\mu}\right)\left(\mu_1 - \bar{\mu}\right)'\right] l}{l' \Sigma l}$$

é dado por $l_1 = e_1$. The combinação linear é chamada Primeiro Discriminante. A combinação linear $l_2'X$ é chamada Segundo Discriminante. Uma estimação da soma é baseada na matriz amostragem de W agrupamentos, dado por:

$$W = \sum_{i=1}^{g} (n_i - 1) S_i = \sum_{i=1}^{g} \sum_{j=1}^{g} (X_{ij} - X)(X_{ij} - X)'$$

Então, $\frac{W}{n_1 + n_2 + ... + n_g - g}$ = Matrix Spooled.

A fim de calcular a média de cada das três classes de performancesomente as 35 observações de cada classe foram utilizadas, isto é , as observações selecionadas para o treinamento do classificador. Então é necessário obter a média de cada dos três pontos médios anteriormente obtidos. Então:

Seja g padrões , μ_i a média dos padrões, $\bar{\mu}$ o vetor média dos padrões combinados e Bo a matriz entre um grupo de padrões $Bo = \sum (X_i - X)(X_i - X)'$

Uma estimação da soma é baseada na matriz de amostragem entre agrupamentos W, dado por:

$$W = \Sigma(n_i - 1) S_i = \Sigma\Sigma(X_{ij} - X)(X_{ij} - X)'$$

Então, $\frac{W}{n_1 + n_2 + ... + n_g - g}$ = Matrix Spooled.

Portanto , o Discriminante de amostragem de Fisher é dado por :

$$\frac{l' B_{0\Lambda} l}{\hat{} W \hat{} l} = \frac{l'\left[\sum_{i=1}^{g} (\mu - \bar{\mu})(\mu - \bar{\mu})'\right] l}{\sum_{i=1}^{g} \sum_{i=1}^{n_i} (Xij - X)(Xij - X)'}$$

Em MatLab:

```
m1=mean(c1trei);
m2=mean(c2trei);
m3=mean(c3trei);
%%% concatenacao das 3 medias em
uma so matriz
m1 = reshape(m1,2,1);
m2 = reshape(m2,2,1);
m3 = reshape(m3,2,1);
mediaux = [m1,m2,m3];
%%% calculo da media das 3
medias
MM = mean([m1,m2,m3]');
MM = reshape(MM,2,1);
```

A fim de se obter a Matriz Covariância foi necessário calcular ^B0. Para se obter a Matrix Combinada, foi necessário calcular as matrizes Spooled e W. Então foi possível obter as matrizes que contém os autovetores e autovalores. Após isto, os autovetores são normalisados fazendo-se a multiplicação com um fator de normalização obtido para cada um dos dois vetores. As funções obtidas são o Discriminante de Fisher. Em MatLab:

```
%%% calculo da matriz covariancia
B0 = zeros(2);
for i=1:3
B0 = B0 + ((mediaux(:,1)-MM)*(mediaux(:,i)-MM)')
end;
%%% calculo da matriz combinada (spooled)
S = ((n1trei-1)*cov(c1trei)+(n2trei-
1)*cov(c2trei)+(n3trei-
1)*cov(c3trei))/n1trei+n2trei+n3trei-3;
%%% calculo de w
W = (n1trei+n2trei+n3trei-3)*S;
WINV = inv(W);
WINVB0 = WINV*B0;
%%% calculo dos autovalores e autovetores
AUTOVAL = eig(WINVB0);
[V,D] = eig(WINVB0);
%%% normalizacao do vetor1
V1 = V(1,:);
V1 = reshape(V1,2,1);
FATNORM1 = 1/sqrt(V1'*V1);
V1NORMAL = FATNORM1*V1;
%%% normalizacao do vetor2
V2 = V(2,:);
V2 = reshape(V2,2,1);
FATNORM2 = 1/sqrt(V2'*V2);
V2NORMAL = FATNORM2*V2;
```

Com o objetivo de se obter as três coordenadas no plano de classificação as últimas 15 observações are utilizadas a partir de cada classe destinada para o teste do classificador (até agor, outras 35 observações foram utilizadas). Em MatLab:

```
%%% leitura das 15 observacoes para teste de cada
uma das 3 classes
load c1teste.dat;
load c2teste.dat;
load c3teste.dat;
%%% rearranjo dos vetores normalizados 1 e 2
(matrizes transpostas) para possibilitar o calculo
seguinte
V1NORMAL = V1NORMAL';
V2NORMAL = V2NORMAL';
%%% calculo das coordenadas dos 3 pontos
(Cluster) no plano de classificacao
Y11 = [(V1NORMAL * m1),(V2NORMAL * m1)];
Y21 = [(V1NORMAL * m2),(V2NORMAL * m2)];
Y31 = [(V1NORMAL * m3),(V2NORMAL * m3)];
```

Utilizando o Cálculo da Distância Euclidiana é possível obter a distância de cada um dos 15 pontos das três classes, em relação aos três pontos médios do plano de classificação. A menor distância de cada ponto em relação aos três pontos médios, faz com que um ponto pertença a um dos três grupos.

A função `'minor.m'` calcula a menor distância entre as coordenandas obtidas. Considerando que a variável q mantém as característica das classes, e r e s são os autovetores e t, u e v são os vetores que possui os pontos médios das classes, então A é o vetor que contém a menor distância calculada e I é o vetor que possui o índice dos menores elementos de A. The função `'classi.m'` foi utilizada para classificar as observações de teste. Esta classificação é conduzida baseada no índice do menor elementodo vetor que possui o cálculo da Distância Euclidiana. A variável p é o vetor de índice. O resultado da função é: A é a matriz que contém os pontos pertencentes a class_1; B é a matriz que contém os pontos pertencentes class_2; C é a matriz que contém os pontos pertencentes a class_3. Em MatLab:

```
%%% obtencao das menores distancias atraves da
chamada de funcao (menor)
[MENOR1,INDICE1] =
menor1(c1teste,V1NORMAL,V2NORMAL,Y11,Y21,Y31);
YAUX1 = Y;
[MENOR2,INDICE2] =
menor1(c2teste,V1NORMAL,V2NORMAL,Y11,Y21,Y31);
YAUX2 = Y;
[MENOR3,INDICE3] =
menor1(c3teste,V1NORMAL,V2NORMAL,Y11,Y21,Y31);
YAUX3 = Y;
```

To confirm the 100% correctness in the classification, it is necessary to calculate the error rate, that proves the inexistence of errors (0%). In MatLbab:

```
% Avaliacao das funcoes discriminantes com base no
procedimento holdout de de Lachenbruch
[POPUL11]  = classi(INDICE1,c1teste);
[POPUL22]  = classi(INDICE2,c2teste);
[POPUL33]  = classi(INDICE3,c3teste);
ERRO = (((n1teste - POPUL11(1,1)) + (n2teste -
POPUL22(1,2)) + (n3teste -
POPUL33(1,3)))/(n1teste+n2teste+n3teste))*100;
```

Este cálculo foi conduzido baseado no holdout por Lachenbruch: ^E (ERA) = ∑niMH / ∑ ni