UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uma Proposta para Gerência de Correio Eletrônico

por

Ana Cristina Benso da Silva

Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Liane Margarida Rockembach Tarouco
Orientador

Prof. Carlos Becker Westphall
Co-orientador

Porto Alegre, Janeiro de 1995.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Silva, Ana Cristina Benso da

Uma Proposta para Gerência de Correio Eletrônico / Ana Cristina Benso da Silva.—Porto Alegre: CPGCC da UFRGS, 1995.

123 p.: il.

Dissertação (mestrado)—Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1995. Orientador: Tarouco, Liane Margarida Rockembach; Co-orientador: Westphall, Carlos Becker

Dissertação: Redes de Computadores, Gerência de Redes

681.327.84(043)
S586p

Comunicação de Dados - SBD
Redes: Computadores
Correio eletrônico
Gerencia: Redes: Computadores
Protocolo X.400
Sistema: Manipulação: Mensagens

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Hélgio Trindade
Pró-Reitor de Pesquisa e Graduação: Prof. Cláudio Sherer
Diretor do Instituto de Informática: Prof. Roberto Tom Price
Coordenador do CPGCC: Prof. José Palazzo Moreira de Oliveira
Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

CNPq 1.03.03.00-6

AGRADECIMENTOS

Agradeço à Profa. Liane Tarouco pela sua orientação.

Agradeço ao Prof. Carlos Westphall pelo seu incentivo.

Agradeço ao CNPq o apoio financeiro.

Agradeço especialmente aos meus pais, pelo apoio, carinho,.....

Agradeço ao Fábio por toda sua paciência, compreensão, carinho e motivação.

Agradeço ao Camillo e ao Mauro pela sua paciência, atenção, bate-papo, etc.

Agradeço ao Mauricio os cházinhos e caronas.

Agradeço ao Luis Otávio, Jorge, Canal e a Eliane a gentileza com que sempre me tratam.

Agradeço aos colegas do pós pela convivência que torna muito agradável a passagem pelo curso. As festas, bate-papos, o amigo secreto (invisível)...

E enfim agradeço a DEUS.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

ACSE	Application Control Service Element
ADMD	Administration Management Domain
ARPA	Advanced Research Projects Agency
ASE	Application Service Element
ASN.1	Abstract Syntax Notation.1
ATM	Agente de Transferência de Mensagens
AU	Agente do Usuário
CCITT	Consultative Committee for International Telegraph and Telephone
CMIP	Common Management Information Protocol
DG	Domínio de Gerência
EIT	Encoded Information Type
IEC	International Eletrothecnical Commission
IETF	International Engeneering Task Force
IFIP	International Federation of Information Processing
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISO	International Organization for Standardization
ISODE	ISO Development Environment
ITU	International Telecommunications Union
LME	Layer Management Information
MADMAN	Management of Directory and Mail Applications
MAPDU	Management Application Protocol Data Unit
MHS	Message Handling System
MIB	Management Information Base
MIME	Multipurpose Internet Mail Extensions
MOTIS	Messaging Oriented Text Interchange System
OSI	Open Systems Interconnection
PDU	Protocol Data Unit

PRMD	Private Management Domain
RFC	Request for Comments
RM	Repositório de Mensagens
SMAE	System Management Application Entity
SMASE	System Management Application Service Element
SMI	Structure of Management Information
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
STM	Sistema de Transferência de Mensagens
TCP	Transmission Control Protocol
UA	Unidade de Acesso
UDP	User Datagram Protocol

RESUMO

A gerência de redes de computadores cresce à medida que são necessárias redes mais seguras, menos vulneráveis a falhas e para reduzir a sobrecarga de trabalho dos administradores de rede. Gerenciar elementos físicos das redes de computadores tem sido a tônica da gerência de redes. Porém, gerenciar as aplicações também é muito importante, pois são o ponto de interação do usuário com a rede.

Há grupos trabalhando na padronização da gerência das aplicações, através da definição de *Management Information Bases* (MIBs) específicas para as aplicações. Recentemente foi publicada a MIB específica para correio eletrônico Internet [KIL 94]. Além de grupos oficiais, como o IFIP Working Group 6.6, outras pessoas, também a nível de Brasil, tem desenvolvido trabalhos na áreas de gerência de aplicações [CAR 94].

Este trabalho propõe um paradigma para gerência de uma das entidades funcionais do *Message Handling System* (MHS) X.400, o Agente de Transferência de Mensagens (ATM). O MHS X.400 é a aplicação de correio eletrônico para o ambiente OSI. O correio eletrônico é uma das aplicações mais populares na comunidade de redes de computadores e atualmente tem sido cogitado como base para muitas outras aplicações.

A definição conceitual do paradigma de gerência baseou-se na estrutura de gerência do modelo OSI, mas a implementação efetiva do protótipo utiliza o protocolo de gerência da arquitetura Internet, o SNMP. As fases de definição e levantamento dos requisitos para gerência de um Agente de Transferência de Mansagens (ATM) basearam-se nos modelos organizacional, funcional e informacional definidos pela ISO.

O modelo organizacional define a hierarquia entre as entidades de gerência, ou seja, a hierarquia entre agentes e gerentes. O modelo funcional descreve os re-

quisitos para gerência do ATM segundo as áreas funcionais do modelo: gerência de falhas, gerência de configuração, gerência de contabilidade, gerência de desempenho e gerência de segurança. O modelo informacional define a MIB, ou seja, os objetos que serão monitorados.

A validação do modelo conceitual foi realizada através da implementação do protótipo. No protótipo foram utilizados os ambientes *ISO Development Environment* (ISODE), PP e *SunNet Manager* (SNM). A implementação do protótipo, além de validar os conceitos teóricos, foi útil para a obtenção de experiência com relação à integração entre os diversos ambientes.

Palavras_Chave: Redes de Computadores, Gerência de Redes, MHS X.400

TITLE: "A PROPOSAL FOR ELETRONIC MAIL MANAGEMENT"

ABSTRACT

The network management has been grown due to the need of more secure networks and to reduce the network adminstrator work. The network physical elements management has been the most important topic in network management, but application management is also important because it is the interaction point between users and networks.

There are groups working on standardization of application management defining Managemet Information Bases (MIBs) specifics for application management. A Management Information Base for Internet eletronic mail [KIL 94] was recently published. Beyond official groups, such as IFIP Working Group 6.6, there are other people investigating the management process of several well-known applications [CAR 94].

This work proposes a paradigm for the management of one funcional entity of Message Handlig System (MHS) X.400, the Message Transfer Agent (MTA). The MHS X.400 is the eletronic mail application for OSI environment. The eletronic mail is one of most popular network application and it has been indicated to be the base for other applications.

The conceptual definition of management paradigm was based in OSI management framework, but the actual prototype implementation uses the SNMP protocol from Internet framework. The definition and elicitation of management requirements for MTA management were based on organization, function and information models defined by ISO.

The organization model defines the hierarchy between the management entities, i.e., the hierarchy between agents and managers. The organization model describes the MTA management requirements by the functional areas: fault management, configuration managemet, accounting management, performance management and security managent. The information model defines the MIB, i.e., the managed objects.

The conceptual model validation was realized throught prototype implementation. For this validation some softwares were used such as *ISO Environment Development* (ISODE) and PP that are sharewares commonly used in academic environments and SunNet Manager (SNM) that is a comercial software. This implementation prototype was helpful to apply the network management concepts and to get knowhow about real network managemnt softwares utilization and integration.

Keywords: Network Computer, Network Management, MHS X.400

1 INTRODUÇÃO

O correio eletrônico é uma das aplicações mais populares no mundo das redes de computadores. O correio é um serviço de entrega e como tal deve oferecer aos seus usuários confiabilidade e a certeza de que suas mensagens chegarão ao seu destino.

A mensagem é composta por um conteúdo e um envelope. O conteúdo é a mensagem em si. O envelope identifica os destinatários da mensagem além de conter outras informações relevantes ao serviço. O usuário submete uma mensagem ao sistema de entrega, e este assume a responsabilidade de entregar a mensagem com sucesso, ou devolvê-la caso ocorra algum erro.

O cenário apresentado pode ser aplicado tanto ao serviço de correio convencional quanto ao correio eletrônico. Na realidade a implementação de ambos é completamente diferente, mas os conceitos, os papéis e as responsabilidades são idênticos [ROS 90].

As experiências pioneiras na ARPAnet adotaram uma estrutura baseada em memorando (*memo-based*) para o conteúdo da mensagem. O protocolo adotado para transferência das mensagens foi o *Simple Mail Transfer Protocol* (SMTP) [POS 82].

Na estrutura baseada em memorando, o conteúdo consiste de duas partes: os cabeçalhos e o corpo da mensagem. Ambas as partes são constituídas de caracteres ASCII. O cabeçalho é rigorosamente estruturado e o corpo tem formato livre [CRO 82]. Pode-se verificar que a mensagem consiste:

- do envelope, o qual tem significado para o sistema;
- dos cabeçalhos, os quais têm significado para as interfaces de correio eletrônico (programas) com as quais o usuário interage;

- do corpo da mensagem, o qual tem significado para o usuário.

O correio eletrônico pioneiro na ARPAnet é ainda hoje o mais utilizado nas redes acadêmicas que utilizam os protocolos TCP/IP [COM 91]. Este serviço ainda adota o SMTP e tem como objetivo principal estabelecer um meio de comunicação. Frente aos avanços de tecnologia e o desejo crescente de tranferir infomações multi-mídia, novos mecanismos para o sistema de correio eletrônico estão sendo criados com o objetivo de tornar o correio da Internet mais atraente, como por exemplo o MIME [BOR 92]. A adição destes mecanismos não implica na substituição do protocolo de transferência de mensagens, pois estes mecanismos realizam suas tarefas de maneira transparente ao protocolo.

Em 1979, a *International Federation of Information Processing* (IFIP) definiu um modelo para sistema de manipulação de mensagens (*Message Handling System* - MHS) para capturar a riqueza do ambiente do sistema de mensagens.

No novo modelo, definido pelo IFIP, a estrutura multi-mídia foi adotada ao invés da estrutura baseada em memorando. Isto significa que o conteúdo da mensagem, do ponto de vista do usuário, consiste de um cabeçalho e um corpo de mensagem que contém várias partes. A função do cabeçalho é similar a do antigo modelo, embora a estrutura seja radicalmente diferente. O corpo da mensagem não está limitado somente ao uso de caracteres ASCII, podendo conter facsimile, voz, telex ou qualquer outra estrutura.

O IFIP teve uma visão mais ambiciosa de não proporcionar somente uma versão eletrônica do serviço de entrega de cartas, mas proporcionar um verdadeiro sistema *store-and-forward.* Isto significa que as mensagens não são necessariamente somente comunicação entre pessoas, mas podem ser comunicação entre processos. Logo, o conteúdo da mensagem pode assumir qualquer formato sem qualquer problema para o sistema de transferência, que deverá entregá-la sem restrições.

Em 1984, o *Consultative Committee for International Telegraph and Telephone* (CCITT) publicou um conjunto de normas sobre o MHS baseado no trabalho do IFIP. Em 1988, o CCITT publicou um novo conjunto de normas do MHS, no qual foram efetuadas correções e adicionadas novas características em relação à versão de 1984. O MHS X.400 de 1988 foi desenvolvido juntamente com a *International Standard Organization / International Eletrotechnical Committee* (ISO/IEC). Para a ISO/IEC a série de padrões é designada como *Messaging Oriented Text Interchange System* (MOTIS).

O MHS X.400 é uma aplicação do nível sete do modelo OSI (*Open System Interconnection*) e foi projetada para satisfazer as exigências de um serviço de correio eletrônico completo, contendo:

- serviço de transporte confiável;
- mecanismos para armazenar as mensagens;
- segurança na transmissão das mensagens;
- interoperabilidade entre sistemas;
- utilização do serviço de diretórios;
- serviço de notificação sobre sucesso ou falha de entrega das mensagens.

Este tem sido o padrão cogitado também para servir de base para outras aplicações que necessitam de um meio de comunicação em um ambiente *store-and-forward*. Atualmente, pode-se pensar no serviço de troca de mensagens interpessoais apenas como uma das muitas facilidades que o correio eletrônico pode oferecer [REI 93]. Aplicações que irão ou estão sendo construídas sobre a infraestrutura de mensagens incluem transferência de informações multi-mídia, roteamento de facsimile, acesso a bases de dados, compartilhamento de documentos, entre outros.

Por exemplo, se atendidas as considerações de segurança, aplicações como transferência eletrônica de fundos (*Eletronic Funds Transfer - ETF*) e intercâmbio

eletrônico de informações (*Eletronic Data Interchange - EDI*) podem utilizar um sistema baseado no padrão X.400 para suas comunicações.

1.1 Gerência do Correio Eletrônico

Independentemente de ser Internet ou X.400, o sistema de correio eletrônico é composto por uma série de mecanismos mais ou menos complexos, variando de padrão para padrão, transparentes ao usuário e dos quais dependem o bom desempenho e confiabilidade da aplicação.

O correio eletrônico é uma aplicação distribuída. O processamento de uma mensagem é realizado através do sistema de correio eletrônico por um ou mais Agentes de Transferência de Mensagens (ATM) até ser entregue ao seu destino. Esse tipo de processamento está sujeito a problemas como:

- perda da mensagem pelo sistema;
- violação dos dados;
- corrupção do formato da mensagem (conversões);
- corrupção de endereços através de sistemas;
- configurações não compatíveis;
- problemas de desempenho, e
- falhas no sistema.

Tais problemas muitas vezes tornam árdua a tarefa do administrador do sistema, fazendo-se necessária a existência de ferramentas de apoio para tornar mais ágil a tarefa de administração.

O ambiente de gerência de redes tem como objetivo apoiar o administrador em suas tarefas, detectando falhas, avaliando o desempenho, e monitorando o funcionamento dos componentes da rede. Segundo a definição da ISO [CAR 93]: o gerenciamento de redes provê mecanismos para a monitoração, controle e coordenação de recursos em um ambiente OSI para a troca de informações entre estes recursos. Esta definição é específica para o modelo de gerenciamento OSI, mas sem dificuldade alguma pode ser aplicada a outros modelos.

A gerência de sistemas de correio eletrônico deve incluir a habilidade de controlar componentes do sistema, AU (Agente do Usuário), RM (Repositório de Mensagens), ATM (Agente de Transferência de Mensagens) e Gateways para outros sistemas de correio eletrônico, bem como os serviços por estes prestados. Os resultados da gerência devem auxiliar os usuários na avaliação do sistema e manutenção da qualidade do serviço através de resultados estatísticos, do reconhecimento de falhas de forma ágil, de histórico de eventos do sistema e de ferramentas inteligentes, que baseadas nos históricos, são capazes de diagnosticar, analisar tendências e reagir a eventos do sistema.

O desenvolvimento da gerência de correio eletrônico é o tema do trabalho de grupos internacionais que estão buscando padronizar e definir as diretrizes para gerência desta e de outras aplicações:

- ISO/IEC e ITU-T estão trabalhando na gerência do MHS X.400;
- IETF MADMAN (Mail and Directory) está trabalhando na gerência do correio eletrônico e serviço de diretório;
- IFIP está trabalhando na gerência do correio eletrônico, especialmente Internet para o qual recentemente publicou uma MIB específica [KIL 94].

Neste trabalho é apresentada uma proposta de um paradigma de gerência de um Agente de Transferência de Mensagens do MHS X.400. A opção pelo X.400 deve-se ao fato deste ser o padrão internacional para correio eletrônico, e a tendência

para os próximos anos é a expansão dos sistemas públicos baseados em X.400 e a integração dos demais mediante o uso de *gateways*. padrão TCP/IP para o padrão OSI.

Os objetivos do trabalho são definir uma MIB específica para o ATM X.400 e implementar de um protótipo para validação da MIB. Além da MIB e do protótipo, espera-se também como resultado a realização de um levantamento do que é necessário inspecionar para a implementação da gerência de aplicações.

Através da MIB ora proposta torna-se possível obter informações sobre o comportamento do ATM de forma mais rápida e fácil, pois a maioria das informações definidas para esta MIB não são facilmente visualizadas por outros meios.

O capítulo dois apresenta os modelos de gerência OSI e Internet. O capítulo três apresenta o MHS X.400. O capítulo quatro apresenta a proposta e requisitos de gerência, e o capítulo cinco apresenta o protótipo implementado para validação do paradigma proposto. Por fim, o capítulo seis apresenta as conclusões sobre o trabalho.

2 ARQUITETURAS DE GERÊNCIA

A gerência em redes de computadores é recente e surgiu da necessidade de controlar as redes complexas e heterôgeneas que estão surgindo com o avanço da tecnologia. O crescimento das redes em tamanho e complexidade representa para os administradores muito trabalho no sentido de manter o funcionamento e o nível de confiabilidade.

A necessidade de gerenciar o ambiente de redes fez com que fossem criadas ferramentas de gerência para auxiliar no trabalho de administração. As ferramentas hoje disponíveis implementam, protocolos proprietários para a comunicação entre os processos de gerência, e também suportam a comunicação com o protocolo de gerência da Internet.

2.1 Arquitetura de Gerência Internet

A atual arquitetura de redes Internet é descrita em *Requests For Comments* (RFCs). Esses documentos descrevem a *Structure of Management Information* (SMI) [McC 90], a *Management Information Base* (MIB) [McC 91], e o protocolo de gerência *Simple Network Management Protocol* (SNMP) [SCH 90].

No modelo conceitual da gerência de redes Internet, um sistema de gerência de redes contém três componentes, figura 2.1:

- nodos gerenciados - sendo que cada um contém um processo de gerência chamado agente;
- estação de gerência - que possui um processo, chamado gerente, que interage com o agente;

- protocolo de gerência - o qual é utilizado pelo gerente e agente para a troca de informações.

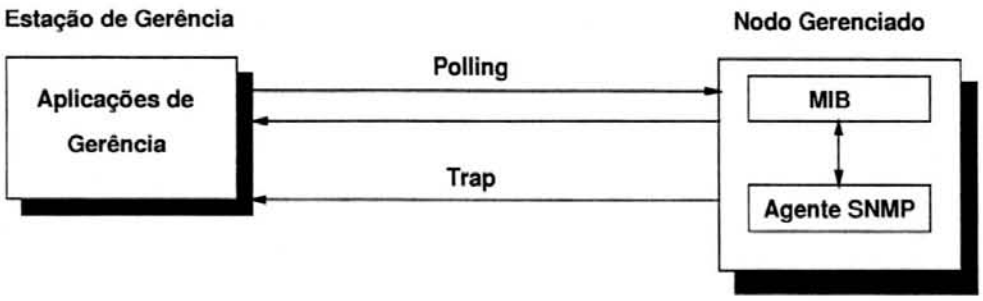

Figura 2.1: Componentes do Modelo Internet

Entende-se por nodo gerenciado quaisquer dispositivos ligados à rede tais como: *hosts* (estações de trabalho, terminais, impressoras), sistemas de *gateways*, equipamentos de comunicação (pontes, *hub*, multiplexadores, modems).

Cada nodo gerenciado contém um conjunto de variáveis que assumem valores conforme as operações executadas. A estação de gerência monitora os nodos gerenciados através da obtenção dos valores das variáveis ou da alteração destes para provocar um determinado comportamento. O paradigma adotado, neste modelo, é o paradigma de monitoração remota, onde o agente coleta informações no nodo gerenciado e as informações são repassadas ao gerente somente sob requisição explícita deste, ou quando ocorre um evento extraordinário.

Outro elemento presente na arquitetura Internet é o agente procurador, *proxy agents*, que traduz as requisições do gerente para a linguagem entendida pelos nodos gerenciados. Este agente é necessário quando as redes são heterôgeneas e alguns equipamentos não suportam a implementação do agente nos moldes do padrão Internet. Ainda pode ocorrer que a ferramenta de gerência utilize um protocolo proprietário para a comunicação entre agentes e gerentes, e o nodo gerenciado não suporte o padrão proprietário, figura 2.2.

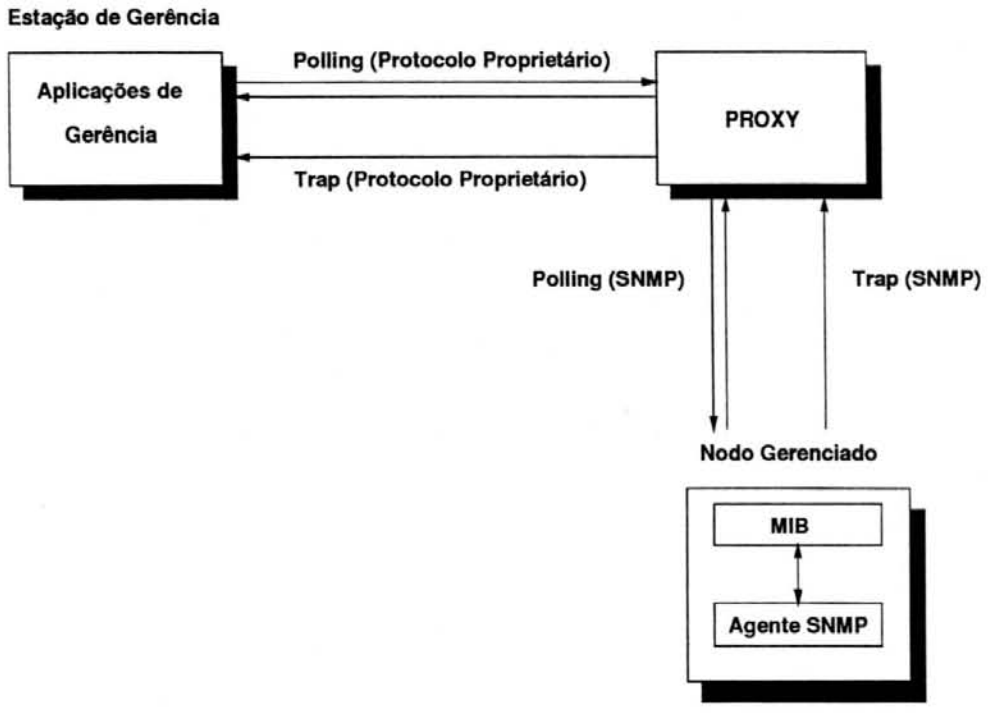

Figura 2.2: Relação entre Gerentes, Agentes e Agentes Proxy

2.1.1 A SMI e a MIB Internet

Na introdução dos conceitos do padrão de gerência Internet, utilizou-se o termo variável, mas deve-se fazer a distinção entre este termo e os termos utilizados na *Structure of Management Information* (SMI). A SMI trabalha com os conceitos de objeto gerenciado e tipo de objeto, significando orientação a objeto, ou seja, um objeto de gerência tem associado a ele sintaxe e semântica, as quais são inteiramente abstratas. Em contraste, uma variável refere-se a uma instância em particular de um objeto em particular. Normalmente, utiliza-se ao invés de variável o termo instância do objeto.

A SMI define o esquema conceitual para a base de dados composta pelos objetos gerenciados que devem ser implementados pelos nodos gerenciados que suportam o conjunto de protocolos da Internet. Esta base de dados é chamada *Management Information Base* (MIB).

A SMI descreve a forma de apresentação dos objetos, as operações sobre cada objeto, a exigência de implementação de cada objeto e o tipo do objeto. A forma de apresentação dos objetos é feita em um subconjunto da linguagem ASN.1 tal como descrito em [McC 90]. A forma de apresentação dos objetos é a seguinte:

OBJECT-TYPE MACRO ::= BEGIN

TYPE NOTATION ::= "SYNTAX" type

"ACCESS" access

"STATUS" status

VALUE NOTATION ::= Value (Value Object Name)

ACCESS ::= "read-only" | "write-only" | "read-write" | "not-accessible"

STATUS ::= "mandatory" | "optional" | "obsolete" | "deprecated"

END

O tipo do objeto é dado pela cláusula SYNTAX. As operações permitidas sobre o objeto são dadas pela cláusula ACCESS, que define implicitamente o nível de acesso aos objetos. A exigência de implementação é dada pela cláusula STATUS. Os objetos gerenciados definidos neste trabalho encontram-se, descritos em ASN.1, no anexo A-1.

A primeira versão da MIB Internet é referenciada como MIB I e foi projetada para incluir um número mínimo de objetos úteis para gerência da Internet. A versão seguinte da MIB, a MIB II, apresenta um acréscimo de 50% em relação à MIB I. A MIB II manteve a compatibilidade com a SMI, portanto contém os mesmos nomes de objetos da MIB I, e foi a MIB que se disseminou sendo adotada por fabricantes às vezes com extensões proprietárias.

A MIB II contém nove grupos de objetos gerenciados: system, interfaces, ip, icmp, tcp, udp, egp, transmission e snmp [ROS 91]. Estes grupos apresentam os objetos fundamentais que devem ser implementados em um nodo gerenciado. As novas inciativas de definições de MIBs específicas para produtos são incentivadas pelo SNMP Working Group, o qual define que as novas MIBs são implementadas em um período experimental até tornarem-se padrões.

2.1.2 O Protocolo SNMP

O protocolo de gerência é o meio pelo qual entidades de gerênica trocam informações. O protocolo SNMP é dito simples, pois suporta um pequeno conjunto de operações que podem ser efetuadas nos objetos da MIB, além de prover um mecanismo de segurança para troca de informações de gerência simples e frágil.

As operações suportadas pelo SNMP são:

- *get* - usado para recuperar uma informação de gerência específica;
- *get-next* - usado para recuperar informações de gerência de objetos tais como tabelas;
- *set* - usado para manipular informações de gerência;
- *trap* - usado para reportar eventos extraordinários.

Para a segurança das informações de gerência, o projeto do SNMP determina os mecanismos de autenticação e autorização usados entre entidades de aplicação SNMP.

O protocolo SNMP utiliza como método de autenticação o conceito de comunidade. Comunidade é a relação entre um gerente e um ou mais agentes SNMP. Quando um agente reporta seus *traps* a um gerente e/ou aceita comandos de um

gerente diz-se que eles pertencem à mesma comunidade. Cada comunidade definida tem um nome que é apresentado no cabeçalho das mensagens do protocolo SNMP. As mensagens SNMP contém duas partes:

- o cabeçalho, onde consta o nome da comunidade juntamente com informações adicionais requeridas para validar a entidade SNMP que está enviando a mensagem;
- e os dados, contendo a operação e operandos associados.

O mecanismo de autenticação baseado em comunidades é trivial e não oferece muita segurança, uma vez que o nome da comunidade é especificado de forma clara. A autenticação se dá quando as entidades envolvidas na operação pertencem à mesma comunidade.

Uma vez autenticadas as entidades SNMPs envolvidas, deve ser determinado o nível de acesso permitido às informações mantidas pelo agente. Um subconjunto arbitrário de objetos visíveis a uma particular comunidade é chamado visão (*view*). Os níveis de acesso permitido a entidades SNMP são:

- somente leitura (*read-only*), ou
- leitura e escrita (*read-write*).

Os mecanismos podem ser exemplificados da seguinte maneira: a comunidade A é composta por três agentes, agente1, agente2 e agente3, e por um gerente, o gerente X. O gerente X tem permissão de leitura e escrita sobre os objetos gerenciados pelos agentes 1, 2 e 3, por pertecerem à mesma comunidade A. Quando o gerente X faz uma solicitação a um dos agentes, ele é autenticado como membro da comunidade A, e então é determinado que ele tem permissões de leitura e escrita (mecanismo de autorização).

Uma comunidade B é formada pelo gerente Y e pelos agentes 4, 5 e 6 e tem comportamento análogo ao descrito para a comunidade A. Mas o fato de serem de comunidades distintas não impede que o gerente X consulte dados dos agentes da comunidade B, nem que o gerente Y consulte a comunidade A. Apenas é criada uma visão com os objetos da comunidade A que podem ser vistos pela comunidade B e são associadas à visão somente permissões de leitura. Assim quando o gerente Y consultar um agente da comunidade A, o processo de autenticação identifica a comunidade e a visão associada. Após, o processo de autorização dará somente permissões de leitura ao gerente Y, figura 2.3.

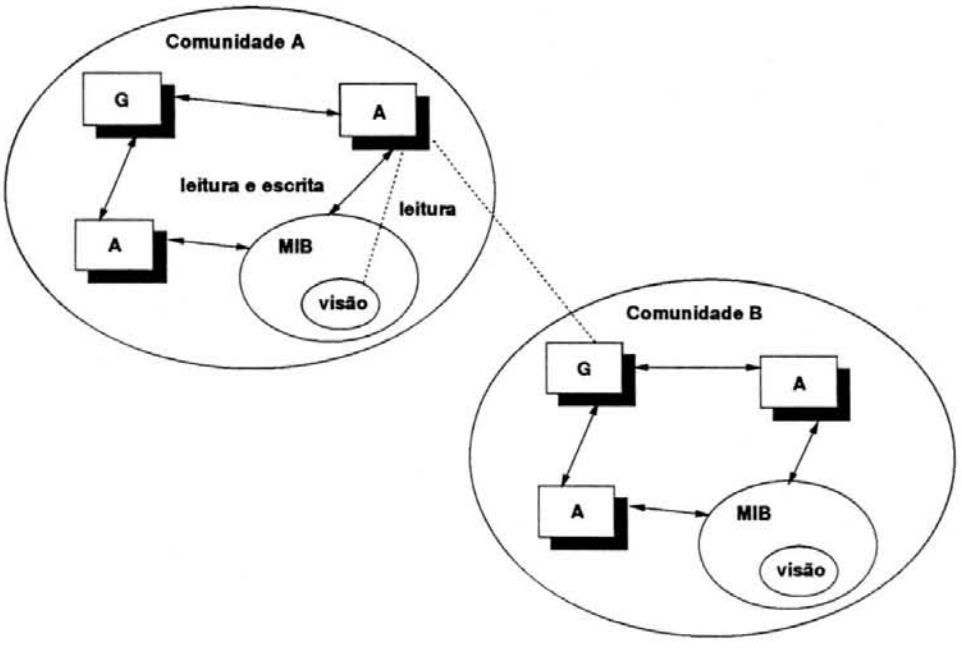

Figura 2.3: Comunicação entre Domínios

O protocolo SNMP foi projetado para ser independente do serviço de transporte subjacente. O protocolo SNMP pode ser mapeado em serviço de transporte UDP [KOC 90], diretamente em mensagens Ethernet, e em protocolos OSI orientados à conexão e não orientados à conexão. Porém o protocolo de transporte mais utilizado para mapear mensagens SNMP é o protocolo UDP.

2.2 Arquitetura de Gerência OSI

Os padrões de gerência OSI foram propostos a fim de solucionar os problemas de integração e limitação de sistemas de gerência de redes. O objetivo destes padrões é o de possibilitar o desenvolvimento de sistemas de gerência e sistemas de comunicação em ambientes heterogêneos, que interoperem entre si.

Como no modelo Internet, o modelo OSI trabalha com os conceitos de, figura 2.4:

- gerente - entidade que obtém informações atualizadas sobre os objetos gerenciados e os controla. Para tal fim, transmite operações de gerência aos agentes;
- agente - entidade que executa operações de gerência sobre os objetos gerenciados, podendo, ainda, transmitir ao gerente as notificações emitidas pelos objetos gerenciados;
- objeto gerenciado - na concepção OSI, é a representação de um recurso que está sujeito à gerência, podendo ser uma entidade de camada, uma conexão ou um dispositivo de comunicação.

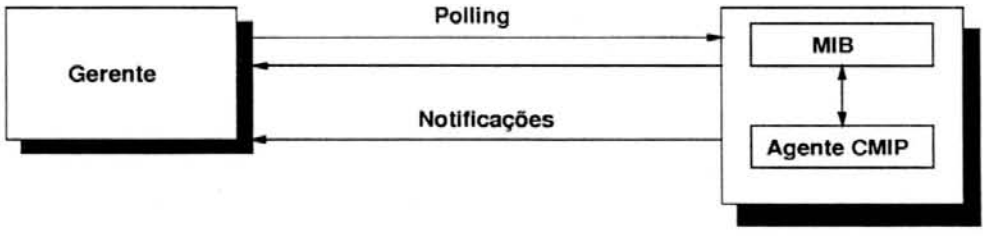

Figura 2.4: Componentes do Modelo OSI

Os objetos gerenciados são definidos em termos de seus atributos ou propriedades das operações a que podem ser submetidos, as notificações que podem emitir para informar sobre a ocorrência de eventos de gerência e das suas relações com outros objetos gerenciados. O conjunto de objetos gerenciados dentro de um sistema constitui, juntamente com seus atributos, a MIB.

Para definir os objetos gerenciados, primeiramente deve-se observar os requisitos de gerência de um sistema. No modelo OSI são definidas cinco áreas funcionais, nas quais tais requisitos devem ser classificados. As áreas funcionais são:

- Gerência de Falhas - abrange a detecção de falhas, assim como o isolamento e a correção de operações anormais do ambiente OSI. Inclui, entre outras, funções para investigar a ocorrência de falhas, identificar falhas, realizar seqüências de testes para fins de diagnósticos e correção de falhas;

- Gerência de Configuração - fornece subsídios para a preparação, a iniciação, a partida, a operação contínua e a posterior suspensão dos serviços de interconexão de sistemas abertos. Para isso, identifica dados, coletando-os e fornecendo-os aos sistemas. Inclui, entre outras, funções para coletar informações sobre as condições do ambiente de comunicação de dados, para obter avisos relativos a mudanças significativas na situação do sistema aberto e para modificar a configuração do mesmo;

- Gerência de Contabilização - inclui funções para informar aos usuários os custos ou recursos consumidos, para permitir a associação do uso de recursos com escalas de tarifação e para possibilitar a combinação de custos no caso de vários recursos serem solicitados para que um dado objetivo de comunicação seja atingido;

- Gerência de Desempenho - possibilita a avaliação do comportamento de recursos no ambiente OSI, assim como o cálculo da eficiência das atividades de comunicação. Inclui, por exemplo, funções para obter informações estatísticas, manter e examinar históricos de sistemas e determinar o desempenho do sistema sob diferentes condições;

- Gerência de Segurança - dá apoio à aplicação de políticas de segurança. Inclui funções para criar, controlar e eliminar mecanismos de segurança, distribuir informações relevantes à segurança e registrar eventos.

A MIB guarda as informações transferidas ou modificadas pelo uso de protocolos de gerência OSI. Tais informações podem ser fornecidas por agentes administrativos locais (pessoas ou programas), ou por sistemas abertos remotos.

O modelo de gerência OSI, figura 2.5, proporciona uma interface, via MIB, com cada uma das sete camadas, oferecendo as operações necessárias para executar a gerência da rede em todas as camadas. A interface específica para cada camada é chamada *Layer Management Entity* (LME). Cada LME concentra a funcionalidade da camada de sua responsabilidade. A integração destas entidades e a função de interface com o gerente é feita pelo *System Management Application Element* (SMAE). Adicionalmente, o SMAE providencia a interface entre os LMEs de um nó da rede com os LMEs de outro nó (LMEs pares), através do protocolo de gerência OSI, o *Common Management Information Protocol* (CMIP).

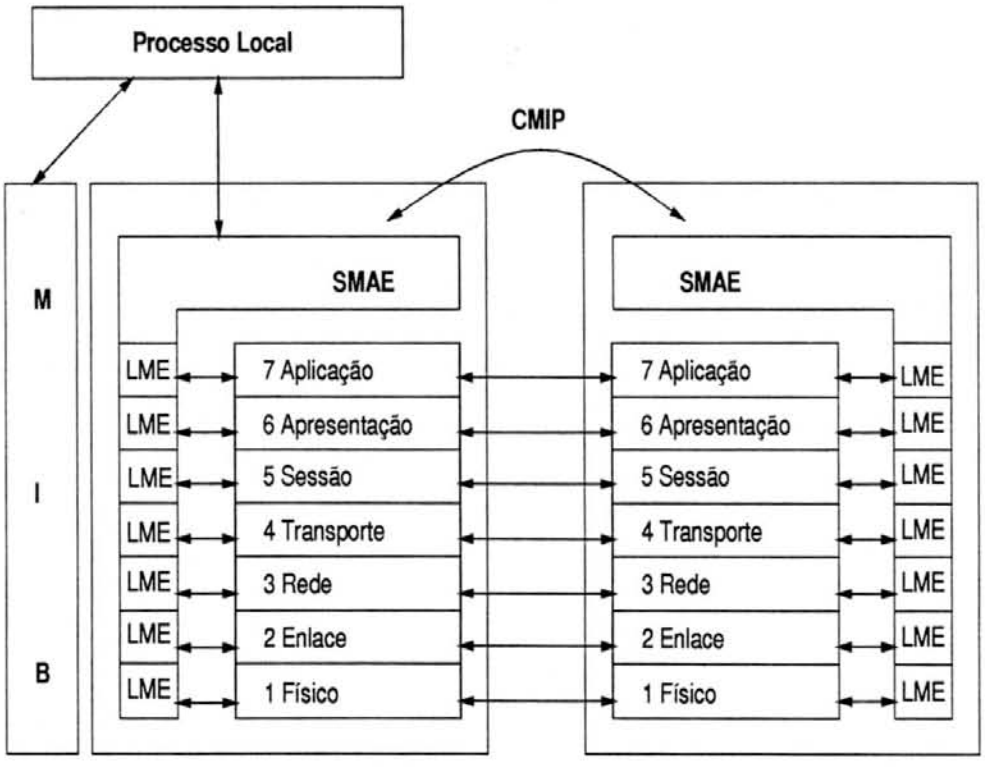

Figura 2.5: Modelo de Gerência OSI

O SMAE, figura 2.5, consiste no *System Management Application Service Element* (SMASE), do *Associantion Control Service Element* (ACSE), e em outros *Application Service Elements* (ASEs).

O SMASE define a semântica e a sintaxe abstrata da informação transferida nas *Management Application Protocol Data Units* (MAPDUs), e especifica a informação de gerência a ser trocada entre SMAEs. Os demais ASEs envolvidos desempenham as funções de comunicação tradicionais especificadas no modelo de referência OSI [CCITT 89a].

2.2.1 Funções de Gerência

A ISO dividiu as atividades de gerência em cinco áreas funcionais, anteriormente citadas. Dentro de cada área funcional foram desenvolvidos padrões de funções para a gerência das redes. Devido à sobreposição dos resquisitos e necessidade dos usuários, algumas funções podem muitas vezes ser utilizadas como suporte a várias áreas funcionais.

As funções definidas pela ISO através das normas 10164-1 a 10164-13 são [CAR 93]:

- Função de Gerenciamento de Objeto - permite ao usuário gerenciar a criação, a remoção, o exame e a modificação das características do objeto gerenciado;
- Função de Gerenciamento de Estado - sua função é controlar a disponibilidade geral do recurso, tornar visível informações em torno desta disponibilidade e, no caso de um recurso não estar em uso, indicar as ações necessárias a serem tomadas para torná-lo viável;
- Atributos para Representação de Relacionamento - tem com função reportar a mudança no valor de um ou mais atributos de relacionamento de um objeto gerenciado em decorrência de uma operação interna do recurso ou de uma operação de gerenciamento.

- Função de Relatório de Alarme - tem como objetivo prover informações ao gerente para que ele possa atuar sobre as condições operacionais e a qualidade de serviço do sistema gerenciado;

- Função de Gerenciamento de Relatórios de Eventos - esta função tem como objetivos: definir um serviço de controle de relatórios de eventos que permita selecionar os relatórios de eventos que devem ser enviados a um sistema de gerenciamento particular; especificar um mecanismo para controlar o repasse de relatórios de eventos que permita, por exemplo, suspender e retomar a transmissão de tais relatórios; possibilitar que um sistema de gerenciamento externo modifique as condições usadas na emissão de relatórios de eventos; designar endereços alternativos para onde os relatórios de eventos possam ser encaminhados em caso de não-disponibilidade do endereço primário;

- Função de Controle de Log - a função de *log* deve satisfazer as seguintes características: o serviço de controle de *log* deve ser flexível de forma a permitir a seleção dos registros num dado *log* que devem ser preservados por um sistema de gerenciamento; um sistema externo deve ser capaz de modificar os critérios usados na preservação dos registros; um sistema externo deve ser capaz de determinar se alguma característica de preservação foi modificada ou se algum registro de *log* foi perdido; devem existir mecanismos que controlem o tempo durante o qual a atividade de preservação das informações deve ocorrer, por exemplo, mecanismos para suspender e retomar a atividade de *login*; um sistema externo deve ser capaz de recuperar e eliminar registros de *log* bem como criar e eliminar *logs*;

- Função de Relatório de Alarme de Segurança - é o meio pelo qual o usuário do gerenciamento de segurança recebe notificações relativas à segurança. Tem a função de informar sobre as operações errôneas nos serviços e mecanismos de segurança, sobre os atentados à segurança dos sistemas e

sobre as violações contra a segurança, quando as mesmas são detectadas por mecanismos de segurança e outros processos relacionados;

- Função de Controle de Acesso - tem por objetivo assegurar que somente usuários autorizados possam ter acesso a um recurso de gerência específico;

- Função de Medida de Contabilização - tem por objetivo a coleta e registro de informações sobre a utilização dos recursos do ambiente para possível associação de valores posteriormente;

- Função de Monitoração de Carga de Trabalho - tem por objetivo avaliar a demanda e a real utilização dos recursos do ambiente e a eficiência das atividades de comunicação, que inclui: obtenção de informações, manutenção e análise dos registros de histórico do sistema, determinação do desempenho do sistema sob condições naturais e artificiais, alteração do modo de operação do sistema;

- Função de Gerenciamento de Testes - tem como objetivo satisfazer as necessidades básicas inerentes ao controle remoto de teste, bem como estabelecer uma estrutura básica para a especificação de testes a serem realizados sobre os recursos gerenciados;

- Função de Sumarização - tem por objetivo obter informações a partir de observações relativas a múltiplos objetos gerenciados.

2.2.2 A SMI e a MIB

A SMI do modelo de gerência OSI especifica o modelo de informação a ser adotado de forma a garantir a interoperabilidade de diferentes sistemas de gerência através de uma visão comum da informação de gerência. A SMI inclui a definição da estrutura da informação armazenada, as operações que podem ser realizadas sobre

as informações e as notificações que podem ser emitidas em decorrência de algumas operações ou alterações destas informações.

No modelo em questão, a SMI define três tipos de hierarquias para representação dos objetos gerenciados, são elas [CAR 93]:

1. Hierarquia de Herança: este tipo de hierarquia está intimamente relacionada com uma abordagem orientada a objetos para representação dos objetos de um sistema de gerência. Nesta hierarquia os objetos são descritos através de seus atributos, comportamento, pacotes condicionais, operações e notificações. Aqui são definidos os conceitos de classe de objetos, herança, superclasses e subclasses;

2. Hierarquia de Nomeação: esta descreve as relações entre instâncias de objetos com seus respectivos nomes, e pode ser também referenciada como hierarquia de *containment*. Este tipo define uma hierarquia através de um relacionamento de "estar contido em" aplicado aos objetos;

3. Hierarquia de Registro: esta é usada para identificar de maneira universal os objetos, independentemente das hierarquias de herança e nomeação. É especificada segundo as regras estabelecidas pela notação *Abstract Syntax Notation.One* (ASN.1) para árvore de registros usada na atribuição de identificadores de objetos. Cada nó da árvore está associado a uma autoridade de registro (por exemplo a ISO), que determina como são atribuídos os seus números. Desta maneira, cada objeto é identificado por uma seqüência de números, cada um correspondente a um nó.

Os objetos gerenciados são representados na MIB através de atributos, notificações e operações. Os objetos da MIB sofrem a ação de operações executadas por sistemas de gerência. Para que se obtenha sucesso na execução de uma operação, o sistema de gerência invocador deve ter os direitos de acesso necessários e as restrições de consistência associadas à classe do objeto gerenciado. A MIB OSI é especificada usando GDMO (*Guidelines for the Definiton of Managed Objects*).

No modelo OSI, existem dois tipos de operações de gerenciamento que podem ser realizadas sobre os objetos gerenciados:

1. Operações orientadas a atributos:
   - *get-attribute-value*: obtém o valor de um atributo;
   - *replace-attribute-value*: altera o valor de um atributo;
   - *set with default value*: altera para o valor especificado;
   - *add member*: inclui valores;
   - *remove member*: remove valores.
2. Operações sobre objetos gerenciados como um todo:
   - *create*: cria um objeto;
   - *delete*: remove um objeto;
   - *action*: executa uma ação específica para um objeto.

Além das operações efetuadas sobre os objetos e as ações definidas para os objetos, o objeto gerenciado tem definido um conjunto de notificações que podem ser emitidas quando algum evento interno ou externo ocorre. Estas notificações são específicas aos objetos gerenciados que as emitem. As notificações e as informações contidas nas operações são parte da definição da classe de objeto gerenciado do qual o objeto em questão é uma instância.

A execução de operações sobre os objetos da MIB deve estar dentro das normas estabelecidas para garantir a integridade e a segurança dos objetos gerenciados. As normas ISO não tratam de aspectos de integridade, que ficam a cargo de cada implementação. Em relação aos aspectos de segurança as normas estabelecem filosofias de controle de acesso que protegem a MIB contra acessos não autorizados.

2.2.3 O Protocolo CMIP

Gerentes e agentes trocam informações sobre os recursos gerenciados, armazenados na MIB através do protocolo de aplicação *Common Management Information Protocol* (CMIP).

O CMIP suporta vários tipos de *Protocol Data Units* (PDUs) que são mapeadas em operações equivalentes sobre os objetos gerenciados, os quais representam os recursos gerenciados. Estas PDUs são, basicamente, as seguintes:

- M-GET: leitura dos atributos de objetos gerenciados;
- M-SET: modificação dos atributos de objetos gerenciados;
- M-ACTION: execução de uma ação qualquer sobre um objeto gerenciado;
- M-CREATE: criação de uma instância de um objeto gerenciado;
- M-DELETE: remoção de uma instância de um objeto gerenciado;
- MEVENT-REPORT: emissão de notificação sobre a ocorrência de um evento associado a um objeto gerenciado.

Além destas mensagens de protocolos, são definidas facilidades adicionais que permitem selecionar o grupo de objetos sobre o qual é aplicável uma dada operação. A facilidade denominada *scoping* permite selecionar um grupo de instâncias de objetos sobre as quais é realizada uma única operação. A facilidade de filtro, por sua vez, permite definir um conjunto de testes aplicáveis a um grupo de instâncias de objetos, anteriormente selecionado através da facilidade de *scoping*, de modo a extrair um subgrupo ainda menor sobre o qual deve ser efetuada uma operação de gerenciamento.

2.3 Comparação entre Arquiteturas Internet e OSI

As arquiteturas diferem no grau de complexidade. Basicamente, os protocolos, SNMP e CMIP, possuem o mesmo objetivo: transferir informações entre sistemas de gerência da rede, dando condições ao gerente de atuar sobre os recursos gerenciados, recuperar informações e identificar problemas. As diferenças entre os dois referem-se, principalmente, à filosofia de acesso aos dados, à funcionalidade, à complexidade, ao desempenho, ao suporte de comunicação requerido e à disponibilidade de produtos.

As diferenças entre os dois modelos podem ser resumidas da seguinte maneira:

- Aquisição de Informações de Gerência: o SNMP emprega o paradigma de *polling*, e adicionalmente é definido um mecanismo de *trap*, por meio do qual um recurso informa ao gerente a ocorrência de uma situação de exceção o que indica que precisa ser executado o *polling* no agente. O CMIP utiliza-se tanto de notificações quanto de *polling*. Não há necessidade de haver um procedimento de *polling* após geração de uma notificação, como ocorre no SNMP, pois as notificações são mais abrangentes;

- Funcionalidade: o CMIP apresenta maior número de diretivas que o SNMP, o que mostra a sua maior capacidade e funcionalidade específica;

- Comunicação: Tanto SNMP quanto CMIP fazem poucas exigências para os serviços de comunicação. Uma diferença chave entre os dois é a exigência do CMIP quanto a um serviço orientado à conexão, enquanto que o SNMP requer um serviço de transporte não orientado à conexão. O CMIP utiliza os serviços de níveis inferiores da pilha OSI orientados à conexão, enquanto o SNMP utiliza o protocolo de transporte UDP;

- Estrutura de Informação:

- o SNMP não provê um mecanismo formal de encapsulamento dos objetos, que são acessados e alterados separadamente. O CMIP permite, através dos mecanismos de *scopping* e filtro, que uma operação seja efetuada em um conjunto de instâncias de objetos;
- no SNMP não há o conceito de herança e classes de objetos. No modelo OSI, um objeto CMIP simples pode modelar um recurso complexo, ou seja, enquanto um objeto CMIP pode modelar várias propriedades de um recurso, um objeto SNMP modela apenas uma propriedade de um recurso, e
- a SMI da Internet restringe o uso dos tipos de dados do ASN.1 para a definição dos objetos.

O SNMP por ser simples tende a ter uma implementação mais rápida e menor que uma implementação CMIP, que obviamente requer maior capacidade de processamento. A discussão entre emprego do SNMP ou CMIP, e sobre a funcionalidade de cada um frequentemente vem à tona nas listas de discussão sobre gerência de redes. Todos concordam com a soberania do CMIP em termos de funcionalidade, sendo que há correntes que afirmam ser a versão dois do SNMP é uma tentativa de aproximação do CMIP e que o substitui com vantagens, uma vez que opera sobre a base instalada, TCP/IP, e oferece funcionalidades equivalentes. Atualmente o SNMP é o protocolo mais utilizado e continuará sendo até que a arquitetura OSI passe a ser mais disseminada.

3 O MESSAGE HANDLING SYSTEM X.400

O MHS executa tarefas de processamento de informação distribuída que interagem com as seguintes sub-tarefas intrínsecas:

- transferência de mensagens: aplicação que transfere objetos de informação entre partes usando computadores como intermediários;
- armazenamento de mensagens: armazenamento automático para recuperação posterior dos objetos de informação transferidos pelo serviço de transferência de mensagens.

A execução destas tarefas é feita através das entidades funcionais do MHS X.400, figura 3.1. As entidades funcionais são:

- Repositório de Mensagens (RM): um RM é uma entidade funcional do MHS X.400 introduzida nas recomendações de 1988. Esta entidade tem como propósito primário armazenar e permitir a recuperação das mensagens do usuário. O RM também permite a submissão de mensagens diretamente ao STM;
- Agente do Usuário (AU): o AU interage diretamente com o usuário, auxiliando a compor e submeter as mensagens para transmissão. O AU também assiste o usuário em outras funções como armazenar, recuperar, responder e apagar mensagens. Este elemento submete mensagens ao ATM para transmissão através da rede, usando o protocolo P3, que é usado entre um agente do usuário remoto e o STM para prover o acesso aos serviços do STM definidos na recomendação X.419 [CCITT 89]. O AU também permite a recuperação de mensagens do RM usando o protocolo P7, que é aplicado entre um agente do usuário remoto e um RM para prover o acesso aos serviços do RM especificados na recomendação X.419 [CCITT 89];

- Agente de Transferência de Mensagens (ATM): o ATM aceita as mensagens submetidas pelo AU ou RM e as entrega aos destinatários. Caso o destinatário seja conhecido pelo ATM a mensagem é entregue. Caso contrário, o ATM transfere a mensagem para outro ATM até que a mensagem chegue ao seu destino. As mensagens são transferidas usando o protocolo P1, cujo propósito é prover as operações distribuídas do STM conforme especificado na recomendação X.419 [CCITT 89];

- Unidade de Acesso (UA): uma UA usa os serviços de transferência de mensagens oferecidos pelo STM. Uma UA é uma entidade funcional associada com um ATM para prover intercomunicação entre o MHS e outros sistemas ou serviços, tais como fax ou correio tradicional.

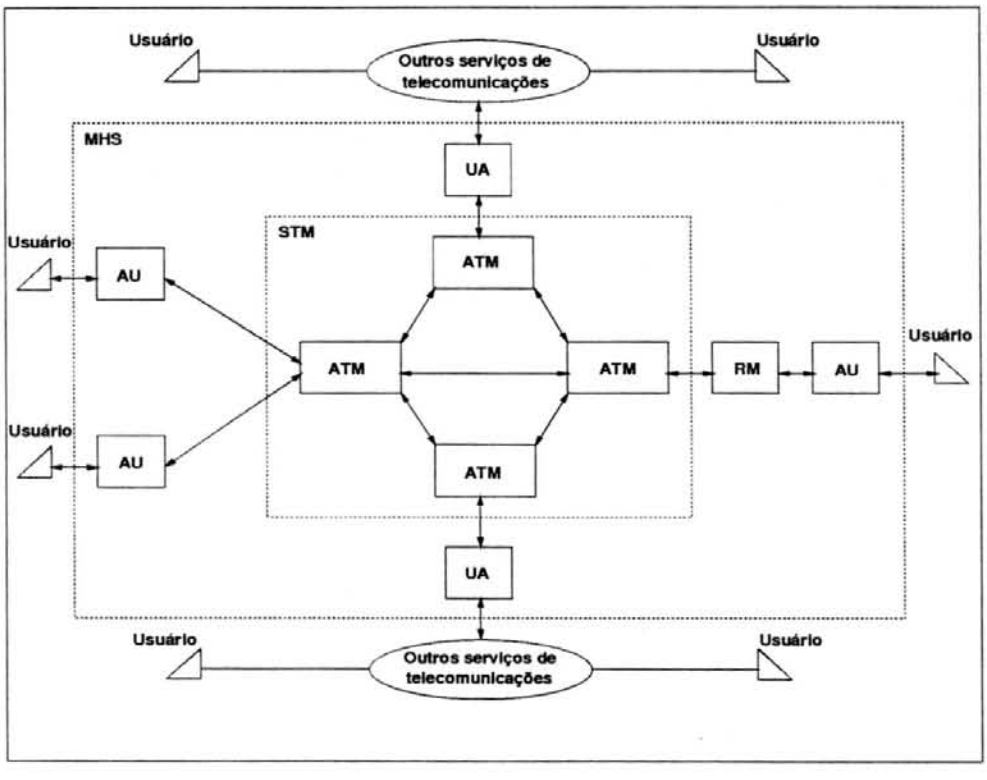

Figura 3.1: Modelo Funcional do MHS X.400

Neste modelo um usuário é uma pessoa ou um processo. Usuários podem ser diretos (usam diretamente o MHS) ou indiretos (precisam de um sistema de comunicação intermediário para acessar o MHS). Um usuário é referido como sendo

o emissor ou receptor das mensagens. As recomendações da série X.400 definem um conjunto de tipos de mensagens e permissões que habilitam o emissor a transferir mensagens para um ou mais receptores. Um emissor prepara uma mensagem com a ajuda de seu AU. O STM (Sistema de Transferência de Mensagens) transmite as mensagens submetidas a ele, para um ou mais AU receptores, UAs ou RMs e retorna a notificação de entrega ao emissor. O STM é formado por um conjunto de ATMs que operam juntos de maneira *store-and-forward* para transmitir as mensagens até seu destino. O MHS é uma aplicação distribuída e define domínios de autoridade e responsabilidade chamados domínios de gerência. O padrão X.400 define domínios de gerência (DG - Domínio de Gerência) como uma coleção de uma ou mais ATMs, zero ou mais AUs e zero ou mais RMs operados por uma administração ou organização. Um DG gerenciado por uma administração é chamado de *Administration Management Domain (ADMD)*. Uma organização que não seja uma administração pode ter um ou mais ATMs, zero ou mais AUs e zero ou mais RMs formando um *Private Management Domain (PRMD)*, o qual pode interagir com um ADMD, como mostra a figura 3.2.

Em um país pode existir um ou mais ADMDs. Um ADMD é caracterizado por prover funções de transferência entre outros domínios de gerência e por prover serviços de transferência de mensagens para aplicações existentes em um ADMD.

Os PRMDs podem ter acesso a mais de um ADMD como mostra a figura 3.2. No entanto, no caso específico da interação entre um PRMD e um ADMD, considera-se que o PRMD está associado a somente um ADMD. É importante salientar também que um PRMD não age como um agente de transferência entre dois ADMDs.

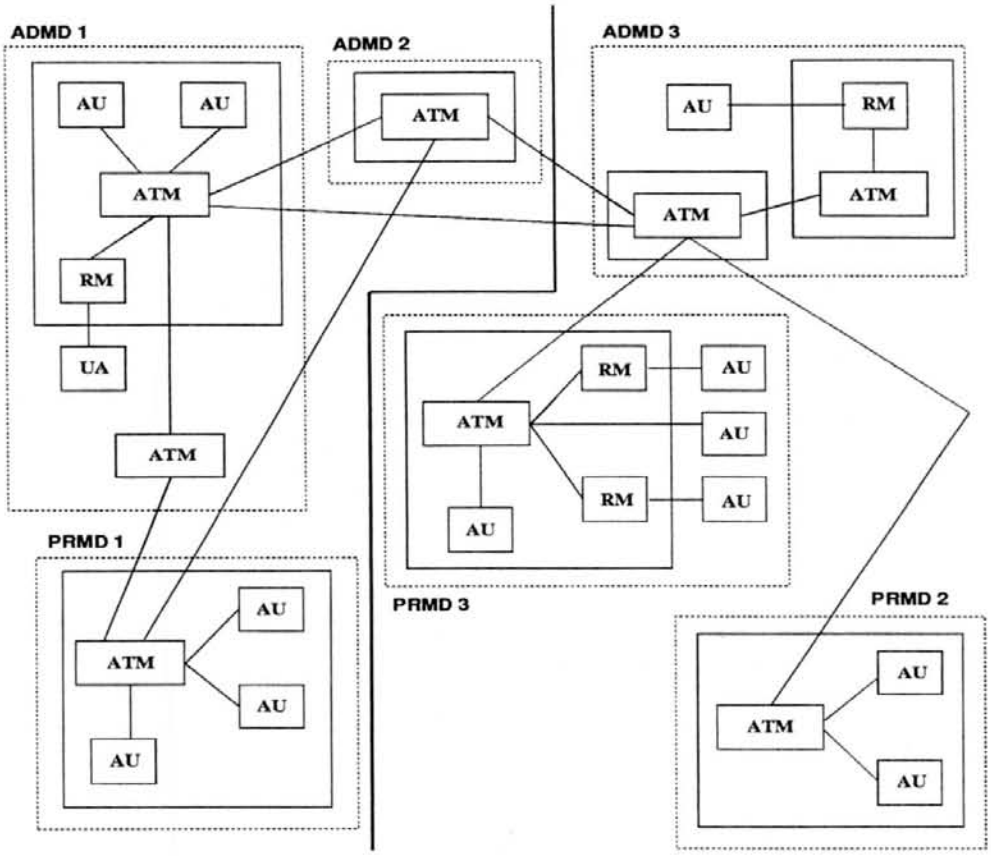

Figura 3.2: Domínios de Gerência

3.1 Modelo do MHS X.400

A norma X.402 [CCITT 89] define quatro modelos, nos quais são estruturados os componentes do MHS: modelo funcional, modelo informacional, modelo operacional e modelo de segurança.

3.1.1 Modelo Funcional

No modelo funcional as entidades e objetos que compõem o MHS são classificados em primários, secundários e terciários, facilitando a visão do modelo. As entidades e objetos classificados em primários, figura 3.3, são:

- o MHS como um todo;

- os usuários;
- as listas de distribuição.

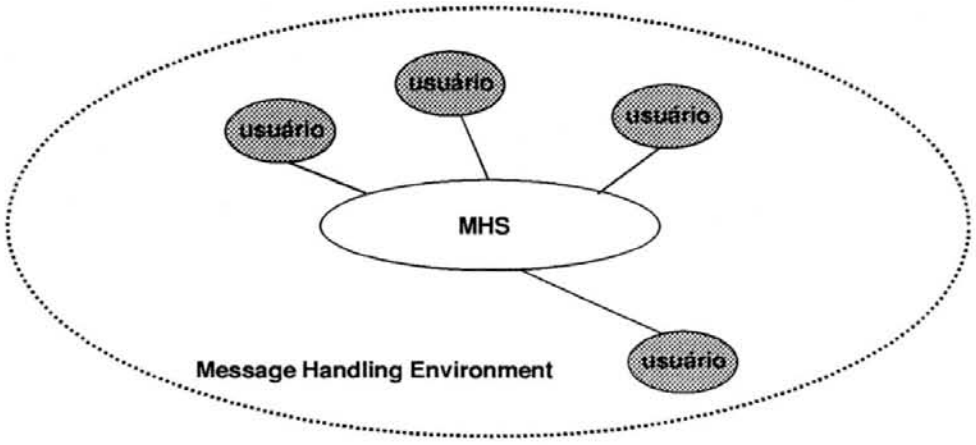

Figura 3.3: Objetos Primários

Os objetos secundários, figura 3.4, são:

- o sistema de transferência de mensagens (STM);
- o agente do usuário (AU);
- o repositório de mensagens (RM);
- a unidade de acesso (UA).

Os objetos terciários, figura 3.5, são:

- o agente de transferência de mensagens;
- alguns tipos de unidades de acesso como:
  - *Physical Delivery Access Unit* (PDAU);
  - TELEX.

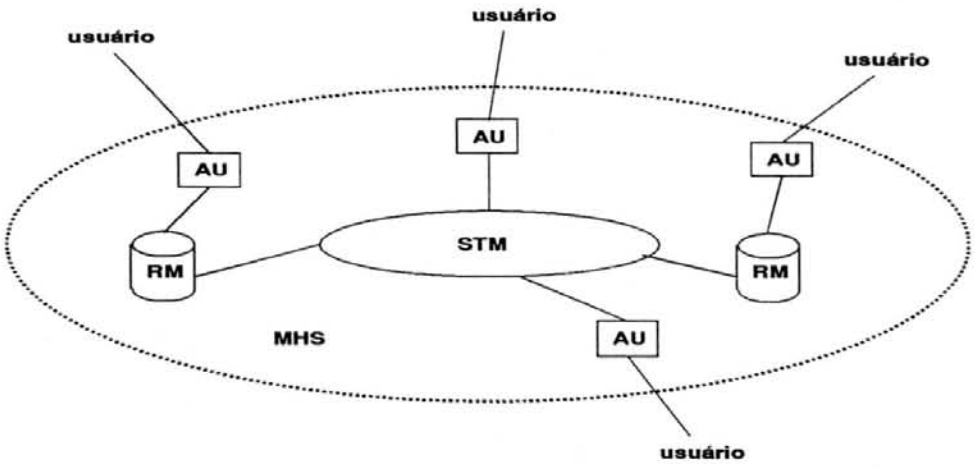

Figura 3.4: Objetos Secundários

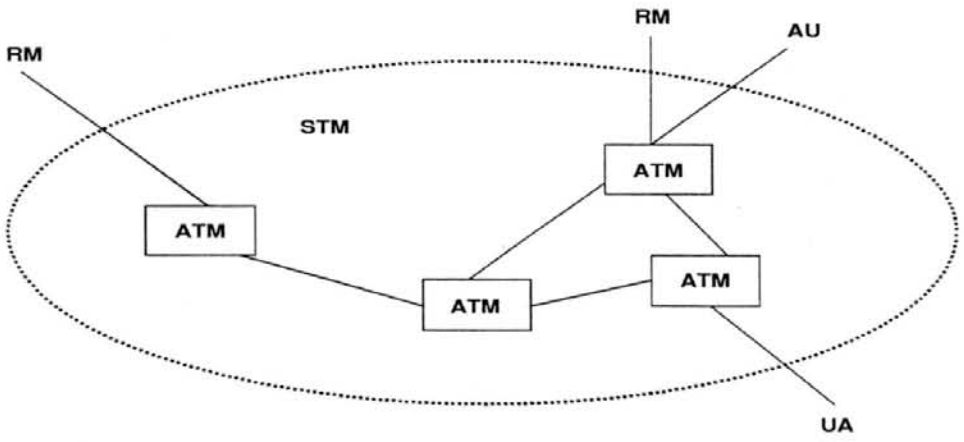

Figura 3.5: Objetos Terciários

3.1.2 Modelo Informacional

No modelo informacional são definidos os objetos de informação que podem ser transmitidos no MHS. Foram definidos três tipos de objetos:

1. Mensagens: objetos transferidos de um usuário para outro. A mensagem é dividida em duas partes: envelope e conteúdo. O envelope contém informações de controle sobre a mensagem, e o conteúdo é a mensagem em si;

2. Testes (*probes*): um objeto que tem como objetivo verificar a viabilidade de entrega de uma mensagem a um destinatário;

3. Relatórios: relato do progresso de transmissão de uma mensagem ou teste para um ou mais destinatários. Existe dois tipos de relatórios: relatório de entrega e o relatório de não entrega, os quais relatam respectivamente o sucesso e o insucesso de entrega de uma mensagem.

3.1.3 Modelo Operacional

No modelo operacional é definido o processo de transmissão dos objetos, que é composto por etapas e eventos. A transmissão de uma mensagem é realizada do seu emissor para o destinatário em potencial. Já a transmissão de um teste é feita do emissor para um ATM que possa confirmar a viabilidade de entrega de uma mensagem para o seu destinatário. E a transmissão de um relatório tem como destinatário o emissor da mensagem ou teste que requer a geração de relatório. As etapas compreendidas pelo processo de transmissão são:

1. Emissão: um usuário direto ou indireto transfere uma mensagem para o sistema de comunicação que o serve. Neste passo o emissor identifica o destinatário da mensagem;

2. Submissão: neste passo uma mensagem é transferida para o ATM, e então é transmitida através do sistema de transferência de mensagens (STM);

3. Importação: a UA transfere a mensagem para o ATM;

4. Transferência: um ATM transfere um objeto para outro ATM. A transferência pode ser interna, dentro de um mesmo domínio, ou externa, entre domínios de gerência diferentes;

5. Exportação: o ATM transfere o objeto para uma UA. A informação é transferida para os limites do STM para outro sistema de comunicação;

6. Entrega: um ATM transfere a mensagem para um RM ou AU;

7. Recuperação: um usuário transfere a mensagem ou relatório do RM para o seu AU;

8. Recepção: um AU transfere a mensagem ou relatório ao seu usuário direto, ou ao sistema de comunicação que serve o usuário indireto.

Os eventos que podem ocorrer durante a transmissão de objetos são:

1. Duplicação: um ATM duplica o objeto quando o próximo passo ou evento exige a transferência do objeto para destinatários diferentes por caminhos diferentes;

2. Combinação: um ATM combina várias instâncias de um mesmo objeto;

3. Resolução de Nomes: um ATM adiciona o correspondente endereço O/R ao nome O/R que identifica os destinatários do objeto;

4. Expansão da Listas de Distribuição: um ATM expande uma lista de distribuição para identificação dos usuários;

5. Redirecionamento: um ATM troca um usuário ou lista de distribuição por um emissor especificado ou destinatário alternativo;

6. Conversão: um ATM transforma partes do conteúdo da mensagem de um tipo de codificação para outro. A conversão pode ser explícita, o emissor seleciona os EITs (*Encoded Information Types*) inicial e final, ou implícita, o ATM seleciona o EIT final baseado no tipo inicial e na capacidade do AU;

7. Não Entrega: um ATM determina que o STM não pode entregar uma mensagem para o destinatário imediato, ou não pode entregar um relatório ao emissor de uma mensagem;

8. Não Afirmação: um ATM determina que o STM não pode entregar uma mensagem de teste. Este evento determina a resposta negativa de uma mensagem de teste;

9. Afirmação: um ATM determina que o STM pode entregar uma mensagem ao destinatário determinado. Este evento determina a resposta afirmativa de uma mensagem de teste;

10. Roteamento: um ATM seleciona um ATM adjacente para o qual transferirá o objeto. Este evento determina a rota de um objeto através do STM.

3.1.4 Modelo de Segurança

No modelo de Segurança são definidas as caracterísitcas de segurança do MHS que podem ser usadas para minimizar o risco de exposição a violações de segurança. O objetivo é prover características independentes dos serviços de comunicação providos pela entidades.

Os serviços de segurança do STM são divididos em classes apresentadas a seguir:

1. Serviços de Autenticação da Origem - provê a autenticação das entidades pares e da fonte (origem) dos dados. Este serviço encarrega-se de confirmar a origem das mensagens, mensagens de teste e relatórios, além de habilitar o emissor da mensagem a obter a confirmação que a mensagem foi recebida pelo STM para entrega ao destinatário originalmente especificado;

2. Serviço de Segurança para Acesso aos dados de Gerência - provê a proteção dos recursos contra o uso não autorizado. O serviço de segurança de acesso aos dados é provido em dois níveis: a nível de estabelecimento de

conexões, confirmando a identidade da entidade que está se conectando e provendo confidencialidade dos dados durante o tempo de conexão; e a nível de passagem de mensagens entre entidades pela referência ao *label* de segurança associado às mensagens;

3. Serviço de Confidencialidade dos Dados - provê a proteção dos dados contra acessos não autorizados. Este serviço provê a confidencialidade do conteúdo da mensagem, confirmando que somente o emissor e o receptor conhecem o conteúdo da mesma;

4. Serviço de Integridade dos Dados - provê a integridade do conteúdo das mensagens, determinando se o conteúdo da mensagem foi alterado;

5. Serviço de Nomeação da Mensagem - provê a associação de um *label* a todas as entidades no MHS, isto é, ATMs e usuários;

6. Serviço de Gerência de Segurança - provê a segurança das credenciais das entidades do sistema e dos *labels* associados as mensagens.

Até aqui foi apresentada uma visão estrutural do MHS, seus componentes e processos. O texto passará a explorar os serviços do STM cujas entidades funcionais que o compõe são o alvo deste trabalho.

3.2 Serviços Abstratos do MHS X.400

Uma descrição do processamento de informações distribuídas leva a especificação de serviços abstratos. Estes serviços são baseados nos conceitos de:

- Procedimentos Abstratos: tarefa que um objeto realiza segundo a requisição de outro;

- Operações Abstratas de Associação: procedimento que associa pares de portas abstratas. Neste caso o objeto que invoca uma operação abstrata de associação é chamado iniciador e o que executa é o respondedor;
- Operações Abstratas de Desassociação: procedimento que encerra a associação de um conjunto de portas;
- Operações Abstratas: procedimentos de propósito geral para transferência de informações;
- Erros Abstratos: condições excepcionais que podem ocorrer durante a execução de uma operação abstrata, levando esta a uma situação de falha.

3.2.1 Serviços Abstratos do STM

O MHS provê a transferência das mensagens entre usuários de maneira *store-and-forward*. Uma mensagem submetida por um usuário, o emissor, é transferida através do STM e entregue a um ou mais usuários, os destinatários.

O STM é modelado como um objeto que descreve o seu comportamento sem referenciar a estrutura interna. Os serviços providos pelo STM estão disponíveis em formas de portas. Um tipo de porta corresponde a um conjunto de operações abstratas.

Antes que os serviços possam ser invocados uns pelos outros, uma operação de associação deve ter sido realizada. Uma associação entre objetos estabelece o relacionamento entre os objetos e só é desfeita com a liberação da associação. Uma associação sempre é liberada pelo iniciador da mesma. Uma associação estabelece as credenciais dos objetos que interagem e os contextos de aplicação e de segurança da associação.

O STM suporta três tipos de portas: de submissão, de entrega e de administração. Essas portas abstratas são serviços suportados por *Application Service Elements* (ASEs) específicos da aplicação de correio eletrônico X.400.

A porta de submissão habilita um usuário do STM a submeter mensagens ao STM para transmissão e entrega da mesma a outros usuários do STM. A porta de submissão habilita também a submissão de testes (*probes*). Os serviços abstratos da porta de submissão são suportados pelo *Message Submission Service Element* (MSSE).

A porta de entrega habilita um usuário do STM a aceitar a entrega das mensagens do STM, bem como os relatórios de entrega, os relatórios de não entrega e os testes. Os serviços abstratos da porta de submissão são suportados pelo *Message Delivery Service Element* (MDSE).

A porta de adminstração habilita o usuário do STM a mudar os parâmetros do STM com a entrega da mensagem e habilita o STM ou o usuário do STM a mudar suas credenciais. Os serviços abstratos da porta de submissão são suportados pelo *Message Administration Service Element* (MASE).

3.2.2 Serviços Abstratos do Sistema de Transferência de Mensagens

Os seguintes serviços são compreendidos pelo STM:

- STM bind - habilita entidades usuárias do STM a estabelecer uma associação com o STM, ou o STM a estabelecer associações com entidades usuárias;
- STM unbind - habilita a liberação da associação pelo iniciador da mesma.

Os serviços abstratos a seguir estão relacionados com as portas abstratas do STM, são eles:

- Porta de Submissão:
  - submissão de mensagens - habilita os usuários do STM a submeterem mensagens ao STM para transferência e entrega para um ou mais usuários do STM;
  - submissão de testes (*probes*) - habilita os a usuários do STM a submeterem testes, de forma a determinar se uma mensagem pode ou não ser transferida e entregue para um ou mais destinatários;
  - cancelamento de entrega - habilita os usuários do STM a requisitarem o cancelamento de uma mensagem previamente submetida;
  - controle de submissão - habilita o STM a limitar o uso da operações abstratas da porta de submissão pelos usuários do STM.
- Porta de Entrega:
  - entrega de mensagens - habilita o STM a entregar uma mensagem aos usuários do STM;
  - entrega de relatório - habilita o STM a enviar confirmação a usuários do STM, o resultado de invocações prévias de submissão de mensagens e testes;
  - controle de entrega - habilita os usuários do STM a limitarem o uso das operações abstratas da porta de entrega por parte do STM.
- Porta de Administração:
  - registro - habilita os usuários do STM a mudarem parâmetros das entidades mantidas pelo STM, associados com a entrega de mensagens;

- mudança de credenciais - possibilita que os usuários do STM mudem suas credenciais com o STM, ou ao STM mudar suas credenciais com o usuário.

Cada um dos serviços abstratos possui uma série de argumentos e resultados [CCITT 89], através dos quais é possível identificar alguns dos objetos gerenciados.

O STM compreende uma coleção de ATMs, as quais cooperam e oferecem os serviços abstratos do STM ao usuário. É o ATM que executa as funções ativas do STM, ou seja, transfere mensagens, testes e relatórios, gera relatórios e executa conversões de conteúdo das mensagens.

Um ATM também tem portas, algumas das quais são aquelas visíveis nos limites do STM, ou seja, porta de submissão, porta de entrega e porta de administração. No entanto o ATM tem outros tipos de portas que dizem respeito à distribuição dos serviços abstratos entre os ATMs, e não estão visíveis nos limites do STM.

A porta de transferência habilita um ATM transferir mensagens, teste e relatórios para outros ATMs. Em geral, os objetos são transferidos inúmeras vezes entre ATMs diferentes até chegar ao destino pretendido.

Uma mensagem endereçada para múltiplos destinatários, servidos por diferentes ATMs, é transferida através do STM por diferentes caminhos. A duplicação da mensagem é realizada até que as cópias cheguem aos seus destinos.

Relatórios, sobre a entrega ou não entrega das mensagens para os seus destinatários, são gerados de acordo com a requisição do emissor da mensagem. Um ATM pode gerar relatório de entrega se a mensagem chegar com sucesso ao destino, ou relatório de não entrega caso a mensagem não seja entregue a um dos destinatários, ou ainda se for incapaz de transmitir a mensagem para o ATM que tem a responsabilidade de entregar ou tranferir a mensagem para outro ATM.

Quando exigido, o ATM pode executar a conversão do conteúdo da mensagem. O emissor pode requisitar explicitamente a conversão da informação para o tipo de informação esperada pelo destinatário.

Os serviços abstratos do ATM, além do serviços demonstrados para o STM, são:

- ATM bind - habilita um ATM a estabelecer associações com outro ATM;
- ATM unbind - habilita a liberação da associação entre dois ATMs pelo inciador da mesma.

Os serviços da porta de transferência do ATM são:

- transferência de mensagens - habilita um ATM a transferir mensagens para outro ATM;
- transferência de testes - habilita um ATM a transferir mensagens de teste para outro ATM;
- transferência de relatórios - habilita um ATM a transferir relatórios para outro ATM.

3.2.3 Procedimentos para operações distribuídas do Sistema de Transferência de Mensagens

O STM é composto por uma série de procedimentos para execução de operações distribuídas, as quais são executadas pelo ATMs. Cada ATM individualmente executa os procedimentos e as ações coletivas dos ATMs provêem os serviços abstratos do STM aos usuários.

Usando técnicas de modelagem, um processo de aplicação ATM pode ser refinado como por exemplo para cada operação abstrata que possa existir entre o

ATM e o usuário que o ATM serve, ou entre ATMs que cooperam existe um único módulo externo responsável pela entrada e saída de mensagens, mensagens de teste e relatórios no ATM, e o suporte de operações como STM *bind* e STM *unbind*, *register*, controle de submissão e controle de entrega.

O ATM é orientado a eventos e permanece "dormindo" até que um evento seja detectado em uma de suas portas. A seguir estão relacionadas as funções do módulo principal do MHS, que dá suporte aos serviços das portas abstratas:

- Processamento de *trace* da mensagem: para cada ATM, pelo qual passa a mensagem, é adicionado um registro de *trace* à mensagem que identifica operações de conversão, roteamento e redirecionamento realizadas;
- Detecção de *loop*: o ATM provê mecanismos para a prevenção de laços nos procedimentos de roteamento, reroteamento e redirecionamento;
- Roteamento e reroteamento: o ATM transfere a mensagem através do STM fazendo o "roteamento" e "reroteamento" das mensagens. Estes procedimentos determinam a rota da mensagem no STM;
- Redirecionamento: o ATM tem a capacidade de redirecionar a mensagem para um destinatário alternativo em caso de não ser possível a entrega para o destinatário original;
- Conversão de Conteúdo: o ATM é capaz de executar a conversão do conteúdo da mensagem para um formato suportado pelo ATM ou AU. As conversões são realizadas quando for explicitamente exigida pelo emissor da mensagem ou quando o sistema não suporta o tipo de conteúdo original e o emissor permite a realização de conversões;
- Expansão das listas de distribuição: o ATM executa a expansão dos destinatários da lista para identificá-los, realizar a replicação da mensagem e determinar a rota das cópias das mensagens;

- Replicação de mensagens: o ATM executa a replicação de mensagens para a transferência destas a diferentes usuários;
- Autenticação da origem: o ATM executa a autenticação da origem da mensagem através da verificação das credencias do usuário com o STM;
- Resolução de nomes: o ATM mapeia nomes em endereços OR.

De forma a executar várias operações abstratas pelas quais é reponsável, o ATM deve executar certos processamentos nas mensagens que entram ou originam-se no ATM.

Baseado na descrição das operações do ATM pode-se salientar os pontos no processamento executado pelo ATM mais sensíveis a falhas. Este levantamento é particularmente importante também para a implementação do tratamento e reconhecimento de falhas que possam ocorrer no ATM e devem ser registradas no histórico do sistema.

Os problemas associados ao processamento de objetos no ATM são:

- Congestionamento: os problemas de congestionamento referem-se a máxima utilização dos recursos do ATM, sendo o ATM incapaz de tratar novas requisições;
- *Loops* de redirecionamento e roteamento: o próprio ATM provê mecanismos de detecção de laços, mas o acompanhamento destas operações pelo processo de gerência adiciona um maior grau de certeza e confiabilidade;
- Conversões: controlar as conversões efetuadas pelo ATM, a fim de manter a integridade das mensagens que trafegam pelo STM;
- Replicação de mensagens: controlar a replicação de mensagens para que usuários não autorizados sejam contemplados com cópias de mensagens indevidamente;

- Resolução de nomes: controle da conversão de nomes em endereço, verificando a correta configuração de servidores de endereços;
- Submissão: controle da taxa de rejeição de submissões ao ATM, auxiliando na detecção de congestionamento;
- Entrega: controle da entrega das mensagens, verificando se o ATM está apto a entregar mensagens em tempo hábil aos seus destinatários;
- Transferência: controle da transferência de mensagens a outros ATMs para detectar possíveis falhas em ATMs adjacentes.

4 PROPOSTA DE GERÊNCIA DO ATM X.400

Este trabalho propõe a gerência do ATM X.400. A figura 4.1 mostra os elementos envolvidos na gerência desta aplicação, sendo implementado pelo trabalho o agente SNMP, e o gerente provido por um pacote de gerência que forneça recursos gráficos, capacidade de geração de alarmes, recursos para monitoração contínua da aplicação, entre outros, facilitando a interface com o usuário. A opção por utilização de um ambiente "pronto" de gerência deve-se à possibilidade de aproveitamento de todas as características e facilidades que um ambiente destes pode oferecer, poupando esforços de implementação.

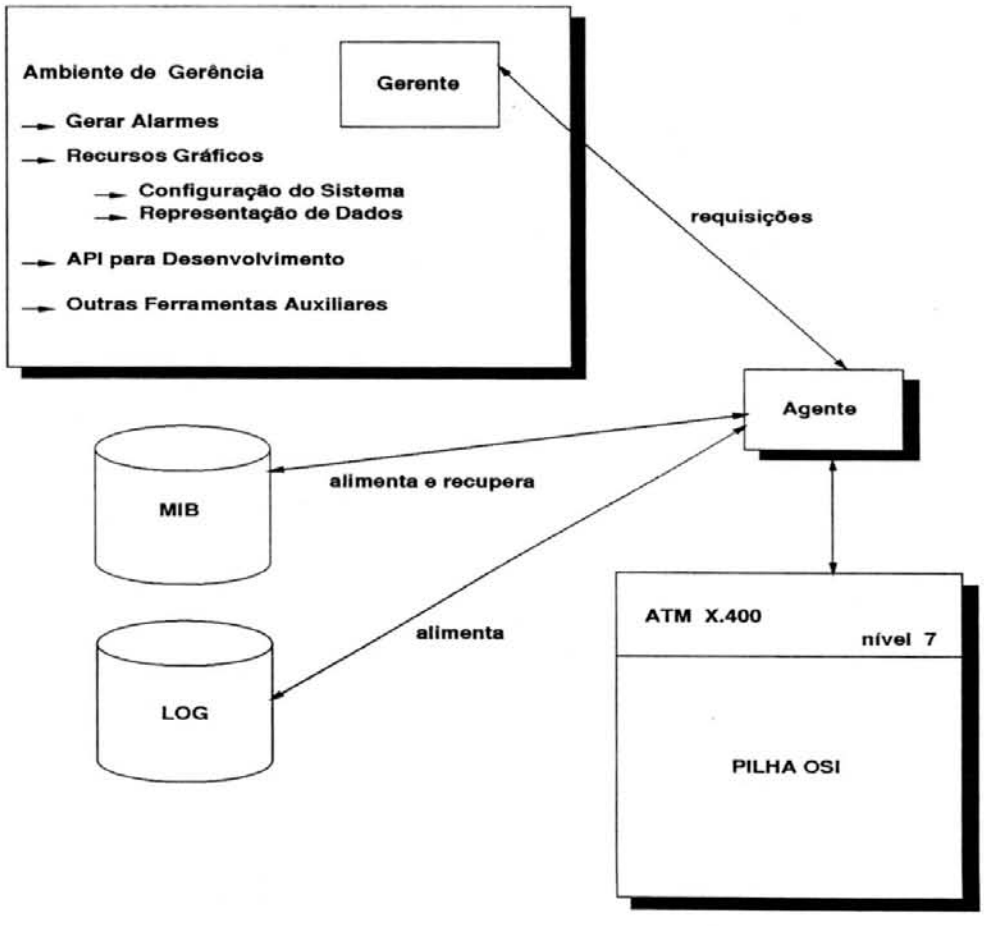

Figura 4.1: Modelo Proposto

As tarefas básicas da gerência de redes, simplificadamente, são obter informações da rede, tratar estas informações, possibilitando um diagnóstico, e encaminhar as soluções dos problemas. Para cumprir estes objetivos, funções de gerência devem ser embutidas nos diversos componentes de uma rede, possibilitando descobrir, prever e reagir a problemas.

Para transferir as informações de gerência existem protocolos de gerência em redes de computadores. O modelo Internet utiliza o protocolo *Simple Network Management Protocol* (SNMP). O modelo OSI utiliza o protocolo *Common Management Information Protocol* (CMIP). Este trabalho adota o protocolo de gerência SNMP [ROS 91], porque, embora a aplicação gerenciada seja construída usando a arquitetura OSI, os sistemas de gerência, bem como as aplicações de gerência atualmente usados nas redes acadêmicas são baseados neste protocolo, e assim o gerenciamento da aplicação de correio eletrônico ficará inserido no contexto geral. Todavia, a MIB projetada para permitir o gerenciamento da aplicação ATM poderá ser facilmente adaptada para permitir o gerenciamento num contexto OSI-CMIP, quando o mesmo se tornar um lugar comum.

O modelo OSI, como visto anteriormente, é bastante rico em comparação ao paradigma de gerência Internet. Um dos objetivos do trabalho é incorporar características do modelo OSI para tornar o agente SNMP mais inteligente, no sentido de oferecer novos mecanismos que aumentem o potencial do agente. A incorporação das funções de *log* no modelo Internet é uma forma simples e elegante de proporcionar novos atrativos. Os dados armazenados em arquivos de *log* tem a intenção de manter um histórico dos eventos do sistema, permitindo gerar dados estatísticos, avisos de possíveis falhas, diagnóstico de falhas e avaliação de desempenho.

Além das características operacionais, como a geração de arquivos de *logs*, foi utilizado para o desenvolvimento do projeto conceitos do modelo OSI para melhor definir o processo de gerência.

Para resolver os problemas associados à gerência em redes, o OSI/NM propôs três modelos [WES 91]:

- O Modelo Organizacional que estabelece a hierarquia entre sistemas de gerência em um domínio de gerência, dividindo o ambiente a ser gerenciado em vários domínios;
- O Modelo Funcional que descreve as funcionalidades de gerência: gerência de falhas, gerência de configuração, gerência de desempenho, gerência de contabilidade e gerência de segurança;
- O Modelo Informacional que define os objetos de gerência, as relações e as operações sobre esses objetos. Uma MIB define conceitualmente os objetos gerenciados.

Estes modelos foram utilizados para definir a política de acesso aos dados, os requisitos a serem gerenciados segundo as áreas funcionais e os objetos que formam a MIB.

4.1 Modelo Organizacional

Os domínios de gerência para o correio eletrônico podem ser baseados na definição de domínio de autoridade e responsabilidade desta aplicação: ADMDs e PRMDs.

Como pode-se deprender na figura 4.2, a gerência do sistema de correio eletrônico depende da interoperabilidade entre domínios de gerência e da habilidade de trocar informações entre elementos de um mesmo domínio e entre elementos de diferentes domínios. Essa troca de informações é fundamental quer pela necessidade de reunir dados recolhidos em diversas localidades para comparação e análise conjunta, quer pela inadequação para apresentação de dados dos equipamentos que os

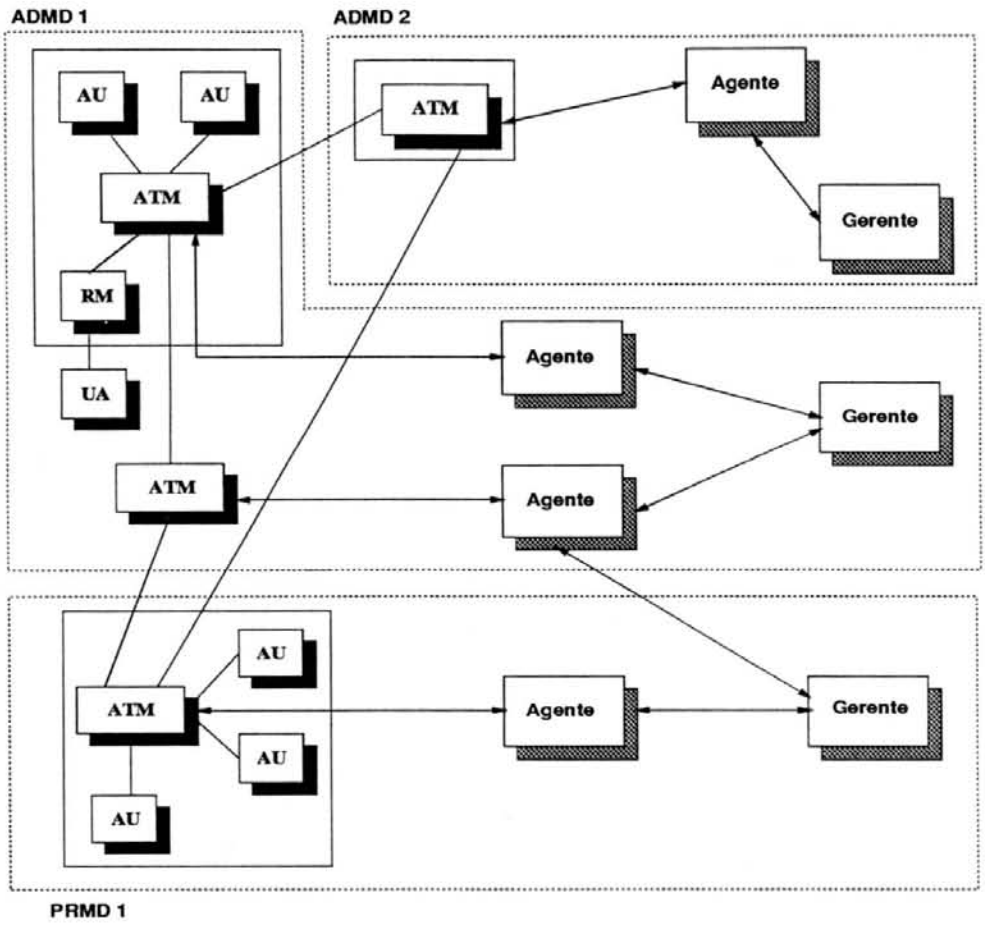

Figura 4.2: Domínios de Gerência

coletam. Conceitualmente no modelo OSI essas relações são: relações intradomínios e relação interdomínios [CAR 93]. A relação intradomínios, neste caso, é exemplificada pela comunicação entre entidades SNMP que pertencem ao mesmo domínio de gerência. A relação interdomínios seria a relação entre entidades SNMP que não pertencem ao mesmo domínio de gerência e então a comunicação seria estabelecida entre gerentes (comunicação *manager-to-manager*). Os gerentes envolvidos na comunicação atuam ora como gerentes, ora como agentes. O gerente que recebe a requisição passa a agir como um agente em relação ao outro gerente, emissor da requisição, e como gerente em relação aos agentes subordinados. Devido à complexidade inerente, neste trabalho, não é utilizada tal arquitetura.

Como o gerenciamento efetivo da rede da UFRGS utiliza o contexto SNMP e também o software SunNet Manager, a relação interdomínios é estabele-

cida diretamente entre o agente e o gerente de domínios diferentes, figura 4.3. Cabe observar que o gerente SunNet Manager utiliza um protocolo proprietário, transportado por RPC (Remote Procedure Call) [SUN 89a], e portanto não comunica-se com o protocolo SNMP diretamente, precisando de um *proxy* para interagir com o SNMP. Como mostra a figura 4.3 as requisições RPC do gerente são enviadas a um processo, local, *proxy* e este as envia como requisição SNMP ao agente local ou remoto.

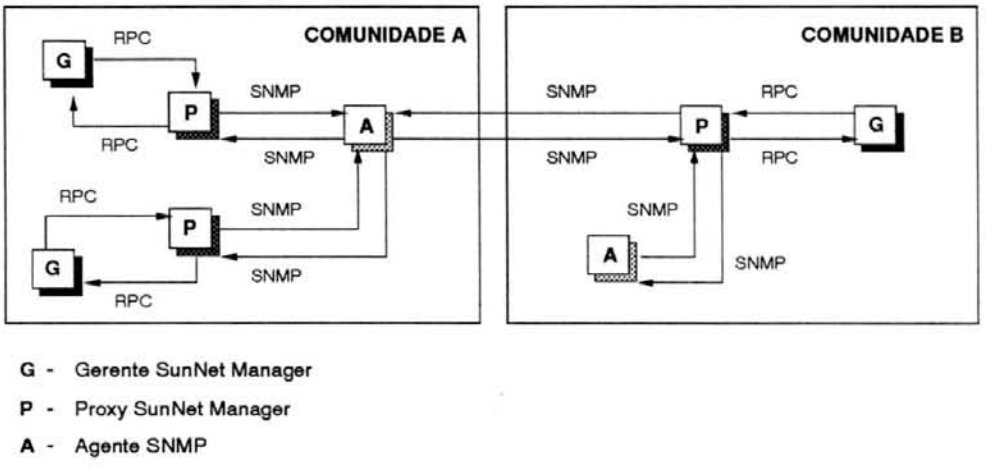

Figura 4.3: Comunicação entre entidades de gerência

Essas relações exigem segurança para os dados de gerência. Neste sentido o protocolo SNMP define dois mecanismos: comunidade e direitos de acesso [ROS 91]. Através do conceito de comunidade o SNMP v1 implementa uma forma trivial de autenticação, isto é, para uma entidade SNMP acessar informações de outra entidade, ela deve identificar-se como membro da comunidade.

O segundo mecanismo, direitos de acesso, descreve as operações que podem ser efetuadas nos objetos gerenciados. Os objetos da MIB tem o atributo "ACCESS:", que descreve os direitos de acesso ao objeto. Este atributo pode assumir os valores: *read-write*, *read-only*, *write-only* e *not_accessible*. Este conceito aplica-se também a entidades SNMP, como forma de proteger os dados da MIB. Uma entidade SNMP que deseja acessar dados de outra entidade SNMP, pertencente ou não à mesma comunidade, deve ser primeiramente autenticada, e conforme

a comunidade a que pertence são estabelecidos os direitos de acesso, desta entidade, aos objetos gerenciados, e quais são os objetos acessíveis.

O SNMP v1 do ISODE [ROS 90], usado na implementação do protótipo em desenvolvimento na UFRGS, provê um arquivo de configuração, o *snmpd.rc*, onde são definidas diretivas que estabelecem as comunidades, os direitos de acesso e os objetos que são visíveis para determinada comunidade. O *snmp.rc* também permite configurar qual gerente deve receber os *traps* gerados pelo agente. As diretivas são:

- **community name address access view**

  Define uma comunidade SNMP chamada **name** com o nível de acesso indicado em **access**. O token **address** pode ser um *hostname*, um endereço IP, ou um endereço de rede. Se o valor deste token for diferente de 0.0.0.0, as requisições a esta comunidade devem ser oriundas do endereço especificado. O token **access** pode assumir os valores: *read-write* ou *read-only*. Caso este token não tenha um valor associado, o valor *default* é *read-only*. O token **view** é um identificador de objeto, que identifica a visão da MIB a qual esta comunidade tem acesso. Se este token não possui um valor associado, o *default* é a visão conter todos os objetos conhecidos pelo agente.

- **view name subtree**

  Define uma coleção de objetos gerenciados com um dado identificador de objeto como **nome**. O token **subtree** define as subárvores (grupos SNMP) que compõem a visão. Se nenhuma subárvore for especificada, a visão é composta por todas variáveis conhecidas pelo agente.

- **trap name address view**

  Define um **trap** para a comunidade SNMP chamada **name**, o qual é enviado ao endereço especificado em **address**. Por *default* o parâmetro **view** não é utilizado.

O ATM X.400 instalado na máquina "espora", da sub-rede gaudéria, pertence ao PRMD=UFRGS.INF. O agente SNMP instalado na mesma máquina pertence à comunidade *public* em função dos gerentes SunNet Manager usarem esta comunidade em suas requisições. Portanto o PRMD=UFRGS.INF está na comunidade de gerência *public*, e os dados dos agentes que monitoram os ATMs deste PRMD podem ser vistos por todas entidades da comunidade pública. O protótipo implementado utiliza apenas a comunidade *public*, pois os gerentes SunNet Manager estão configurados nesta comunidade. No arquivo *snmpd.rc*, para o protótipo, foi adicionada a diretiva *community public*. Não é definida uma *view*, ficando todos os objetos visíveis aos gerentes da comunidade *public*.

4.2 Modelo Funcional

As classes funcionais definidas pelo modelo funcional são: gerência de falhas, gerência de configuração, gerência de contabilidade, gerência de desempenho e gerência de segurança.

Gerenciar um sistema de correio eletrônico é gerenciar os elementos que compõem este sistema, ou seja, gerenciar AUs, RMs, ATMs e UAs. Um ponto importante é a monitoração das mensagens fim-a-fim para verificar se as mensagens submetidas ao MHS estão sendo entregues aos destinatários corretos com alto grau de certeza, integridade e segurança.

A seguir estão especificados os requisitos para gerência de ATMs, segundo as áreas funcionais. Estes requisitos são baseados no estudo do padrão X.400 e nos levantamentos feitos em [McC 93].

4.2.1 Gerência de Falhas

Caracteriza-se por gerência de falhas o conjunto de funções que habilitam um gerente ou aplicação detectar, isolar e corrigir falhas de componentes e/ou serviços. Os requisitos identificados que dizem respeito à área de gerência de falhas são:

- detectar, isolar e corrigir falhas em ATMs;
- emitir notificações sobre falhas aos responsáveis;
- interação entre entidades de gerência para detecção e correção de falhas, bem como para possíveis alterações de configuração decorrentes de falhas;
- manter *trace* das mensagens fim-a-fim;
- emitir notificações sobre congestionamento no sistema de correio eletrônico;
- criar arquivos de *log* de falhas para auxiliar em diagnósticos futuros, análises de comportamento e na correção automática e inteligente de falhas através da análise de soluções, anteriormente aplicadas. Um registro em um *log* deve conter os seguinte itens básicos:
  - data da ocorrência da falha;
  - identificação do ATM;
  - descrição do problema;
  - solução adotada;
  - *status* dos demais elementos do sistema;
  - responsável pela correção da falha.

4.2.2 Gerência de Configuração

Caracteriza-se por gerência de configuração o conjunto de atividades usadas para controlar a configuração do sistema. Gerência de configuração também inclui *engineering support*, que é o processo usado para determinar o sistema ótimo, baseado nos dados de desempenho, utilização dos recursos e requisitos do sistema. Os requisitos identificados que dizem respeito à área de gerência de configuração são:

- suportar serviço de diretórios;
- capacidade de alterar a configuração do sistema automaticamente;
- prover uma base de dados de configuração para suportar operações de administração, análise e planejamento;
- capacidade de localizar componentes e recursos de software;
- suportar informações de roteamento sobre ATMs e suas interfaces;
- capacidade de alterar a configuração de roteamento, conforme análise do tráfego e filas do ATM.

4.2.3 Gerência de Contabilidade

Caracteriza-se por gerência de contabilidade o conjunto de funções usadas para medir o uso do serviço e prover informações de cobrança sobre os serviços utilizados. Os requisitos identificados que dizem respeito à área de gerência de contabilidade são:

- medir o uso do correio eletrônico, pelo volume de mensagens transmitidas e recebidas;

- coletar dados de contabilidade;
- capacidade de analisar o custo operacional de componentes e sumarizar o custo de múltiplos recursos;
- capacidade de gerar informações de cobrança para agências de cobrança;
- registrar informações de contabilidade relacionado ao uso dos recursos em arquivos de *log*.

4.2.4 Gerência de Desempenho

Caracteriza-se por gerência de desempenho o conjunto de funções que avaliam, relatam e otimizam o comportamento operacional do sistema e serviços do usuário. Os requisitos identificados que dizem respeito à área de gerência de desempenho são:

- monitorar e diagnosticar condições de tráfego de correio eletrônico insatisfatórias;
- capacidade de coletar dados para análise de desempenho em ATMs. Em especial controlar as filas do ATM;
- capacidade de gerar análises estatítiscas de desempenho a curto, médio e longo prazo;
- capacidade de gerar notificações sobre problemas de desempenho aos responsáveis.

4.2.5 Gerência de Segurança

Caracteriza-se por gerência de segurança o conjunto de funções usadas para manter a autenticação dos componentes do sistema, contabilidade, confiden-

cialidade, integridade e controle de acesso. Os requisitos identificados que dizem respeito à área de gerência de segurança são:

- autenticação e controle de acesso às informações de gerência;
- proteção das mensagens do usuário contra acessos não autorizados;
- manter a integrigadade das mensagens transferidas no STM.

4.3 Modelo Informacional

O Modelo Informacional define os objetos de gerência, as relações e as operações sobre estes objetos. Uma MIB (*Management Base Information*) é necessária para armazenar os objetos gerenciados.

Estes objetos são informações relevantes à gerência e que devem satisfazer os requisitos estabelecidos no Modelo Funcional. Os objetos aqui apresentados formam um conjunto de informações para gerência do ATM X.400. Esta aplicação é bastante complexa, sendo que a especifiação da MIB pode ser exaustivamente detalhada prejudicando o procedimento de gerência com informações desnecessárias.

Este trabalho propõe um conjunto de objetos, especificados a partir do estudo dos serviços e operações de um ATM. O objetivo desta MIB é facilitar a administração do sistema, tornando disponíveis informações, que nem sempre são visíveis ao administrador, de forma rápida e menos complexa.

Nem todos objetos da MIB podem ser avaliados isoladamente, devido à complexidade da aplicação de correio eletrônico. A interpretação de um evento é dependente da avaliação de vários objetos gerenciados. Outras vezes os objetos podem, por si só, conterem informações que tem significado sem a necessidade de combiná-los com outros objetos.

Os objetos foram identificados pelo estudo do padrão X.400 [CCITT 89] com base nos requisitos de gerência do modelo funcional. Abaixo estão especificados os objetos e sua justificativa. A MIB foi escrita conforme a estrutura de informação do modelo Internet e está descrita em anexo.

4.3.1 Informações de Configuração

Informações sobre a configuração do ATM servem para verificações de configuração por parte das entidades do domínio, e por entidades de outros domínios de gerência, para teste de compatibilidade, e/ou simples reconhecimento do ATM. Adicionando informação sobre o estado do ATM pode-se detectar possíveis problemas de falhas e congestionamento.

- atmlocname - nome do ATM;
- atmlocaddr - endereço do ATM;
- atmlocdomain - domínio a que pertence o ATM (PRMD);
- applocname - nome da aplicação;
- protocolname - identificação do protocolo;
- protlocver - versão do protocolo;
- atmlocstatus - estado do ATM;
- convertable - tabela com os tipos de conversões que podem ser executadas no ATM;
- eittalbe - tabela com os tipos de *Encoded Information Types* suportados pelo ATM.

Além das informações do ATM local, mantém-se outras informações em forma de tabela, sobre ATMs adjacentes. Estas informações podem ser obtidas

através de uma requisição do gerente ao ATM remoto, mas há especial interesse em manter-se uma tabela sobre os ATMs adjacentes para traçar um esquema de roteamento das mensagens, detectar rotas alternativas, possíveis falhas ao longo do sistema de transferência, e também o reconhecimento do sistema de transferência de mensagens ao menos parcialmente. Os objetos mantidos na tabela são:

- atmadjname - nome do ATM;
- atmadjaddr - endereço do ATM;
- atmadjdomain - domínio do ATM;
- atmadjstatus - status do ATM adjacente.

4.3.2 Informações de Controle de Fluxo

Estas informações possibilitam a avaliação da capacidade de processamento do ATM, dimensionamento do sistema e a contabilização do uso dos recursos. As informações deste grupo permitem a avaliação do volume de dados que trafegam pelo ATM, através da medição do volume total de mensagens que foram processadas pelo ATM. Também é monitorado o tamanho da fila de submissão e transferência de mensagens do ATM, permitindo a avaliação da capacidade de processamento do mesmo. As informações são:

- atmqueueIns - número de mensagens submetidas à fila;
- atmqueueOuts - número de mensagens transferidas da fila;
- submitedMsgtots - total de mensagens submetidas ao ATM;
- transfMsgtots - total de mensagens transferidas do ATM;
- storedMsgtots - total de mensagens mantidas no ATM;

- rejectMsgtots - total de mensagens rejeitadas pelo ATM;
- submitedMsgVols - volume total de mensagens submetidas ao ATM;
- transfMsgVols - volume total de mensagens transferidas pelo ATM;
- storedMsgVols - volume total de mensagens mantidas no ATM;
- reporttotals - número total de relatórios transferidos pelo ATM;
- probetotals - número total de mensagens de teste *probes* transferidas pelo ATM.

As informações sobre o número de total de relatórios e mensagens de teste são interessantes para que em comparação ao número total de mensagens do ATM possam ter-se uma idéia do volume de tráfego desses objetos bem como das mensagens normais.

4.3.3 Informações de Controle de Falhas

Informações que permitem identificar a fragilidade de um ATM frente ao número de falhas. Estas informações servem para manter um histórico de estatísticas de falhas ocorridas no sistema durante o seu tempo de vida útil. Os objetos são:

- atmdateStart - a data de inicialização do ATM;
- atmlastfault - a última data da ocorrência de uma falha;
- atmfaultTotals - número total de falhas do ATM na história;
- atmactivetime - tempo total de atividade do ATM desde que foi inicializado (atmdateStart);
- atminactivetime - tempo total em que o ATM ficou inativo desde a última falha.

4.3.4 Informações de Controle de Mensagens

Informações de controle das mensagens permitem verificar e monitorar possíveis problemas de integridade, conversões, confiabilidade da mensagem fim-a-fim.

Manter informações sobre todas as mensagens que trafegam por um ATM é uma prática inviável pelo grande volume de dados que representam. Portanto optou-se por manter informações sobre mensagens que por ventura sejam alvo de erro no ATM.

Os objetos são:

- msgId - identificação da mensagem;
- msgsubtime - data de submissão da mensagem ao ATM;
- msgdeltime - tempo estimado para entrega da mensagem;
- atmFromAdd - endereço do ATM que transferiu a mensagem;
- atmToAdd - endereço do ATM para o qual será transferida a mensagem;
- msgsize - tamanho da mensagem;
- originorname - nome do emissor da mensagem;
- reciporname - nome do receptor da mensagem;
- priority - prioridade da mensagem;
- implconverprohib - permissão de conversão implícita do conteúdo da mensagem;
- convlossprohib - permissão de conversão do conteúdo da mensagem com perda de dados;
- defdeltime - tempo programado para entrega da mensagem;

- latestdeltime - a última data programada para entrega da mensagem;
- physforwarprohib - permissão para transferência da mensagem para outros sistemas de comunicação;
- origreporreq - indicação de requisição de geração de relatório pelo emissor;
- origencodtype - *encoded-information type* original da mensagem;
- submissionstat - resultado da submissão da mensagem;
- deliverystat - resultado da entrega da mensagem;
- transfstat - resultado da transferência da mensagem;
- reasonreject - razão da rejeição da mensagem, caso haja rejeição na submissão;
- nondeliveryreason - razão da não entrega, caso ocorra algum problema;
- nondeldiagnostic - o diagnóstico da não entrega;
- tracemsg - *trace* da mensagem durante sua transferência no STM.

4.3.5 Informações sobre Testes e Relatórios

O mesmo interesse relativo às mensagens que trafegam no sistema aplica-se às mensagens de teste e relatórios. Estas informações devem permitir a visualização de possíveis falhas do sistema com respeito a esses elementos. Os objetos são:

- origprobname - nome do emissor da mensagem de teste;
- recipprobname - nome do destino da mensagem de teste;
- probid - identificação da mensagem de teste;

- submissiontime - data da submissão do teste;
- submissionstatus - estado da submissão da mensagem especial;
- repreasreject - razão, caso haja rejeição na submissão;
- reportid - identificação do relatório;
- repreciporname - nome do emissor da requisição de relatório;
- result - resultado da geração de relatórios.

4.3.6 Informações sobre Associações

Informações que permitem monitorar o comportamento do ATM no que concerne ao estabelecimento de associações com outros ATMs para transferência de mensagens. Estes dados permitem, também, a contabilização dos recursos por tempo de duração de uma associação e volume de dados transferidos por associação.

As informações sobre as associações realizadas devem ser analisadas levando em consideração outras informações como tempo de atividade do ATM, pois se analisadas de forma isolada podem não ter significado expressivo. Por exemplo, numa amostragem A foram identificadas 200 associações estabelecidas e em uma amostragem B foram identificadas 1300 associações estabelecidas. Estes dados nada representam se não forem consideradas outras informações para que se possa avaliar em qual dos dois casos pode haver comportamento anormal, ou qual valor representa utilização intensa ou ociosidade do ATM, entre outros.

- inAssocTotals - total de associações requisitadas ao ATM;
- outAssocTotals - total de associações requisitadas pelo ATM;
- lastInAssocs - data da última associação requisitada ao ATM e estabelecida;

- lastOutAssocs - data da última associação requisitada pelo ATM e estabelecida;
- rejectInAssocTotals - total de associações requisitadas ao ATM e rejeitadas;
- rejectOutAssocTotals - total de associações requisitadas pelo ATM e rejeitadas;
- abortedAssocTotals - total de associações abortadas;
- associd - identificação da associação;
- initAdd - identificação da entidade requisitante da associação;
- acceptorAdd - identificação da entidade par que recebe a solicitação;
- assocTime - tempo de duração da associação;
- transfmsgVolume - volume de mensagens de correio eletrônico transferidas na associação;
- assocStatus - estado da associação;
- assocReason - razão da rejeição ou fim da associação.

4.3.7 Informações sobre Histórico

Os arquivos de *log* são gerados para manter-se o histórico do sistema, permitindo avaliações *a posteriori* dos eventos. Para que o agente SNMP possa acessar os dados do *log* é necessário que se defina objetos na MIB.

Os arquivos de *log* fazem parte da MIB através dos objetos gerenciados, mas ao contrário dos demais objetos são gravados em arquivo. O formato do arquivo é dados por uma estrutura em linguagem C, conforme será explicado no capítulo seguinte. Os objetos são:

- data - data em ocorreu uma falha (evento);
- atmname - endereço IP do ATM;
- problema - descrição do problema;
- solução - descrição da solução adotada;
- responsável - identificação do responsável pela solução da falha.

4.4 Outros Mecanismos

Além das variáveis da MIB é necessário definir outros mecanismos para obtenção de informações e ativação de procedimentos perante falhas e para avaliação do sistema.

Como mencionado anteriormente o sistema incorpora a geração de arquivos de *log* para utilização com ferramentas inteligentes para geração de diagnósticos de falhas, avaliação de desempenho e prevenção de falhas em potencial.

Além dos arquivos de *log* há determinados requisitos de gerência que podem ser satisfeitos através da implementação de *traps* específicos melhor do que através de objetos da MIB. Os eventos do sistema, sujeitos a implementações através de *traps*, são:

- falhas no ATM adjacente;
- violação das credenciais;
- violação de integridade da mensagem;
- tentativas de acesso não autorizados à MIB.

Quando da ocorrência de um destes eventos extraordinários, o *trap* será gerado para o gerente, para que este seja informado do evento tão logo este ocorra. No protótipo implementado não está sendo tratada a geração de *traps*.

5 O PROTÓTIPO

Para validar o paradigma de gerência proposto, foi implementado um protótipo. O trabalho foi desenvolvido em ambiente TCP/IP, utilizando implementações disponíveis para a área acadêmica: o ambiente ISODE versão 8.0, o agente SNMP versão 1 do ISODE e o PP versão 6.0, o qual implementa um ATM que suporta vários protocolos de transferência de mensagens, entre eles o X.400 P1. As informações de gerência estão sendo canalizadas para o ambiente SunNet Manager.

5.1 ISO Development Environment

O ISODE [ROS 90] é uma implementação não proprietária de alguns protocolos definidos pela ISO/IEC. O propósito de tornar este software *shareware* é acelerar o processo de desenvolvimento de aplicações no conjunto de protocolos OSI.

Este softaware pode suportar diferentes serviços de rede abaixo do *transport service access point* (TSAP). Um desses serviços de redes é o TCP. Isto permite o desenvolvimento de protocolos dos níveis superiores do modelo OSI em ambiente Internet.

O ISODE é composto basicamente pelos seguintes módulos:

- serviço de transporte OSI classe 0 conforme rfc1006 [ROS 87];
- serviço de apresentação e sessão;
- compilador ASN.1;
- os elementos de serviço de aplicação ACSE, ROSE e RTSE;

- ftam e um gateway ftam-ftp;
- terminal virtual (vt);
- serviço de diretórios OSI, QUIPU;
- agente SNMP versão 1.

5.2 PP

O PP [KIL 91] é um ATM que suporta uma variedade de protocolos de transferência de mensagens, incluindo: X.400 (1984) P1, X.400 (1988) ISO(10021) P1; e outros protocolos baseados na rfc822 [CRO 82].

O PP é fruto do desenvolvimento conjunto de diversas instituições e seu desenvolvimento foi baseado em experiências prévias com ATMs, e tem as seguintes características:

- trabalha com grande volume de mensagens;
- características de gerência;
- facilidades para conversão de protocolos, particularmente para mapeamento entre protocolos baseados na rfc822 [CRO 82] e X.400 de acordo com rfc987 [KIL 86] e rfc1148 [KIL 90].
- conversão de formatos do corpo das mensagens;
- suporte para o desenvolvimento de agentes do usuário, particularmente para aqueles que desejam usar X.400 e/ou capacidades multi-mídia.

O PP é portável para sistemas UNIX e sistemas compatíveis com UNIX. Para comunicação interna o PP precisa do OSI/ROS, que é implementado pelo

ISODE, bem como os níveis superiores da pilha OSI. As ferramentas ASN.1 do ISODE também são necessárias para a instalação do PP.

A documentação do PP versão 6.0 especifica o uso do ISODE versão 7.0. Mas neste trabalho foi utilizado o ISODE versão 8.0.

O PP consiste de uma série de procedimentos, chamados canais, que executam as tarefas de um ATM [KIL 91]:

- canais obrigatórios - executam funções especiais como submissão de mensagens e gerenciamento da fila do ATM;
- canais opcionais - programas que implementam diversos protocolos, os quais são suportados pelo PP, conversores de conteúdo de mensagens, e *gateways* entre os protocolos.

Os canais estão subordinados ao processo *QMGR*, que gerencia o ATM. Os canais apropriados para o tratamento das mensagens são ativados pelo *QMGR*. O *QMGR* primeiro chama o processo *pptsad* passando como parâmetro o nome do canal a ser ativado. O *pptsapd* executa uma operação *fork* e executa o canal apropriado passando o controle da execução ao *QMGR*.

O *QMGR* então inicializa o canal, ao qual informa que mensagem deve ser processada e o conjunto de destinatários. O canal então passa a execução do trabalho e após executá-lo atualiza a fila com o resultado da operação.

O funcionamento de todos os elementos está baseado em tabelas de configuração. Após a instalação do software deve ser feita a configuração do mesmo. Esta configuração consiste na edição do arquivo *tailor* e especificação de dados como: o endereço do ATM, o endereço do *postamaster* local, e informações sobre os canais ativos neste ATM e seus parâmentros.

Além do arquivo *tailor*, o PP apresenta uma série de tabelas que contém dados específicos desde "apelidos" para endereços reais, tipo de mensagens que o usuário recebe (rfc822, X.400), informações de roteamento e normalização para endereços X.400 e endereços baseados em rfc822, até informações necessárias para a execução das funções de *gateway* entre protocolos de correio eletrônico.

No momento de submissão de uma mensagem ao ATM, figura 5.1, o processo *submit* é ativado. Este processo estabelece uma associação com o *QMGR* e a mensagem é submetida ao ATM. Os demais processos envolvidos, como conversões e a própia entrega ou transferência da mensagem, estão subordinados ao processo *QMGR*.

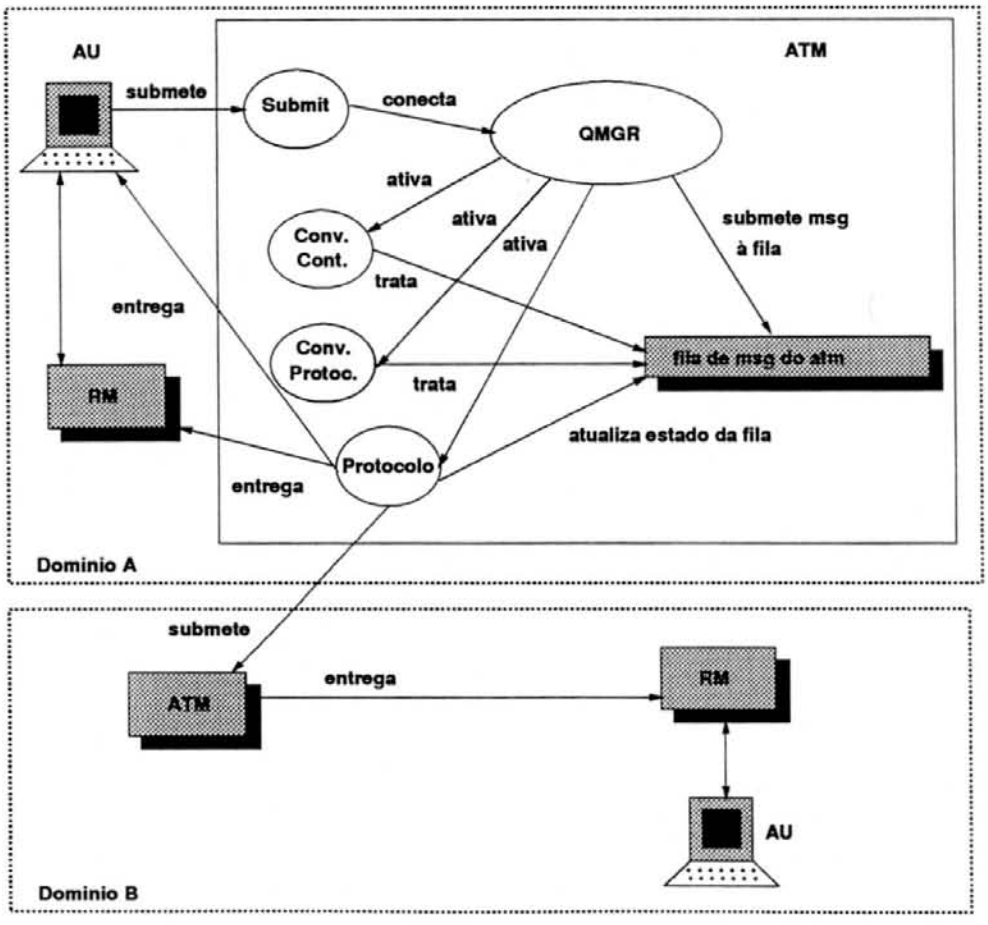

Figura 5.1: Processo de Submissão de Mensagens

Uma mensagem em formato rfc822 pode ser enviada para um usuário X.400. O processo *QMGR* encarrega-se de ativar os procedimentos necessários para a

conversão da mensagem para o formato X.400, da mesma forma que uma mensagem X.400 pode ser enviada para sistemas baseados na rfc822. Neste último caso a conversão das mensagens está sujeita a perda de informações se for composta por tipos de dados que não são convertidos para caracteres ASCII.

5.3 SunNet Manager

O SunNet Manager [SUN 89] é uma plataforma de gerência de redes heterogêneas, sobre a qual o desenvolvimento de procedimentos de gerência é dissociado da complexidade dos ambientes heterogêneos.

O projeto do SunNet Manager é baseado no modelo gerente/agente do modelo de gerência OSI. O gerente é um processo inicializado pelo usuário. O agente é um processo que coleta os dados dos objetos gerenciados e reporta-os ao gerente. A figura 5.2, demonstra as relações entre gerentes e agentes do SunNet Manager.

A comunicação entre agentes e gerentes SunNet Manager é feita através de *Remote Procedure Call* (RPC). Para suportar a gerência de equipamentos que não suportam RPC, o SunNet Manager oferece um agente procurador SNMP para interação com os agentes SNMP.

Além das facilidades gráficas e outras ferramentas como *browser*, o SunNet Manager oferece uma *Application Program Interface* (API) para o desenvolvimento de novos processos de gerência. A API oferece uma biblioteca de serviços de gerentes/agentes que provê a infra-estrutura de gerência e trata os serviços de comunicação. O agente e gerente não necessitam conhecer a complexidade subjacente envolvida na comunicação.

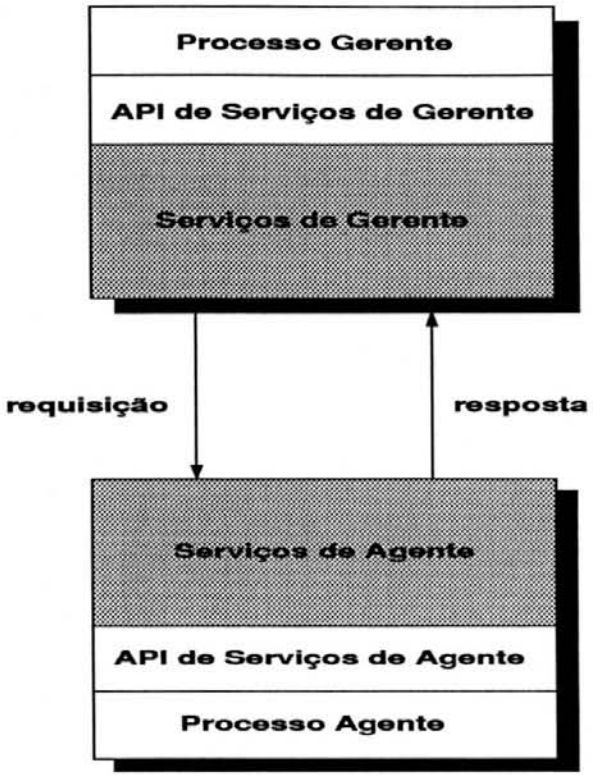

Figura 5.2: Modelo Agente/Gerente

5.4 Arquitetura do Protótipo

O protótipo, figura 5.3, utiliza os softwares apresentados anteriormente.

O SunNet Manager oferece ao ambiente ferramentas e outros requisitos especificados na proposta de gerência. Entre outros, este ambiente, oferece interface gráfica para interação com o usuário, bem como para a visualização do *lay-out* do sistema, ferramenta gráfica para visualização dos dados resultantes da gerência, *browser* para percorrer a MIB, métodos de geração de avisos, capacidade de estabelecer limites de valores para objetos gerenciados.

O SunNet Manager oferece três tipos de gerentes para interação com o agente. O "QUICK DUMP" que executa uma única consulta ao agente, como diz o nome, consulta rápida. O "EVENT REPORT" monitora os dados em intervalos de tempo determinados pelo usuário e gera alarmes quando as condições estabelecidas

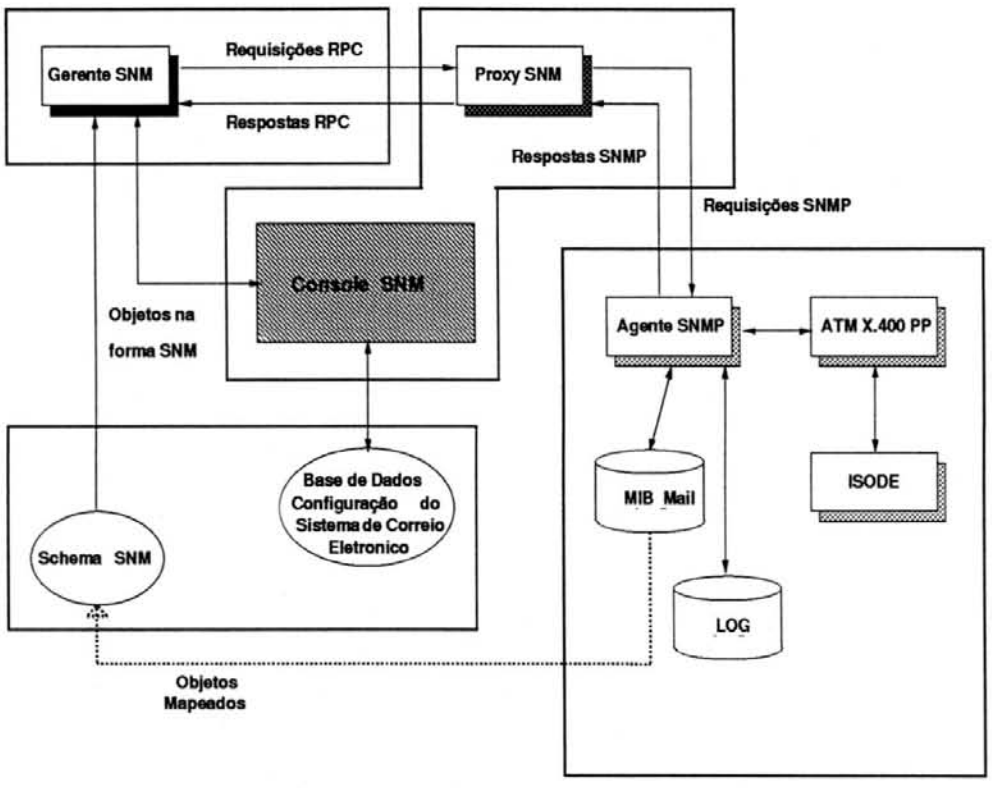

Figura 5.3: Arquitetura do Protótipo

pelos usuários forem alcançadas. Os alarmes podem ser gráficos, sonoros, sonoros e gráficos ou o gerente pode notificar o administrador através de correio eletrônico.

O "DATA REPORT" monitora dados em intervalos de tempo pré estabelecidos pelo gerente, podendo monstrá-los graficamente, através da ferramenta para geração de gráficos.

Estes gerentes são ativados pelo usuário através da interface gráfica do SunNet Manager, figura 5.4. O ATM é representado pela máquina em que está instalado. Escolhida a máquina, escolhe-se no menu qual gerente será executado. Escolhido o gerente aparece um novo menu com os agentes que estão configurados naquela máquina. A escolha do agente implica na ativação de um terceiro menu que mostra um grupo de objetos que o agente monitora.

Os demais softwares envolvidos, o ISODE e o PP, são respectivamente a implementação da pilha OSI sobre TCP/IP necessária para suportar as aplicações

OSI, e a aplicação de correio eletrônico. O ISODE além de implementar os níveis superiores OSI oferece um agente SNMP versão 1.0, que está sendo utilizado no protótipo.

Como observa-se na figura 5.3 os dados de gerência são obtidos pelo agente SNMP através da monitoração da aplicação de correio eletrônico. Estes dados são passados ao gerente que os torna visíveis ao usuário. Os dados estão constantemente sendo monitorados e atualizados, uma vez que o agente seja inicializado. Mas estes dados são repassados ao gerente somente sob requisição deste.

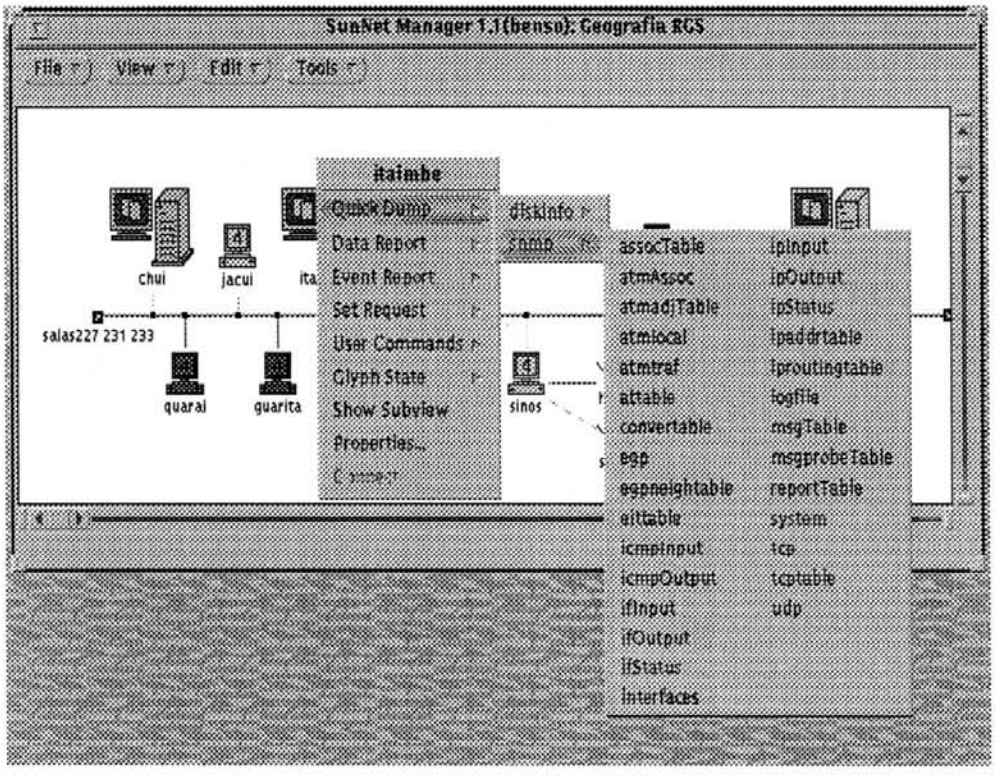

Figura 5.4: Interface Gráfica do SunNet Manager

Os objetos gerenciados pelo agente SNMP são mantidos em uma MIB no formato definido pela SMI da arquitetura de gerência Internet. A MIB foi facilmente convertida para o formato de *schema* do SunNet Manager pelo utilitário *mib2schema*. Este mapeamento é necessário, pois o gerente SunNet Manager necessita conhecer os objetos para poder manipulá-los.

Os objetos da MIB mail foram adicionados ao arquivo *schema* do agente *proxy* **snmp**. A simples adição dos objetos ao *schema* e a recompilação deste arquivo pelo SunNet Manager faz com que o *proxy* reconheça os objetos.

5.4.1 LOG

O agente SNMP além de alimentar a MIB mantém arquivos de *log* das atividade do ATM, com o intuito de formar um histórico do sistema que permitirá futuras avaliações do ATM.

O registro das falhas permite a avaliação das falhas ocorridas no ATM sob vários aspectos:

- avaliação da sucetibilidade do ATM à falhas;
- causas mais freqüentes das falhas;
- períodos mais sucetíveis a falhas;
- prever futuras falhas baseadas nos registros do *log*;
- identificar possíveis soluções;
- quem pode solucionar as falhas.

A obtenção de informações sobre falhas irrecuperáveis são obtidas pelo *agente ping*. Este agente foi instalado em outra máquina da rede do ATM para registrar falhas na máquina onde está instalado o ATM. A função deste agente verificar se a máquina está ativa através do comando "ping".

Outro agente utilizado para monitorar falhas do ATM é o *agente processo*. Este agente, assim como o *agente ping*, foi desenvolvido com base em experiências prévias no desenvolvimento de agentes SunNet Manager [SIL 92]. O agente monitora

a tabela de processos da máquina, verificando possíveis problemas com os processos que implementam o ATM. As falhas detectadas pelos agentes são gravadas no arquivo de *log*.

As falhas referentes ao funcionamento do ATM são monitoradas na própria aplicação que implementa o ATM.

Conforme especificado no Modelo Funcional foi implementado o *log* de falhas. Este arquivo foi implementado como um arquivo ASCII com a seguinte estrutura em linguagem C:

```
struct log{
char *data;              /* data */
char *hora;              /* hora */
char *problema;          /* descricao do problema */
char *solucao;           /* descricao da solucao adotada */
char status[n][2];       /* variaveis associadas */
char *responsavel;       /* responsavel pela solucao */
} logs;
```

O agente SNMP acessa os dados do arquivo de *log* através de objetos da MIB especialmente definidos para esta função. Dessa forma o gerente pode requisitar a recuperação de registros de falhas através do agente SNMP, figura 5.5.

Os dois agentes, *ping* e *processo*, monitoram falhas de uma forma mais abrangente, ou seja, monitoram falhas que podem ocorrer com a aplicação X.400, mas que poderiam ocorrer com qualquer outra aplicação.

As falhas específicas do processo de aplicação ATM monitoradas neste trabalho e já especificadas anteriormente são:

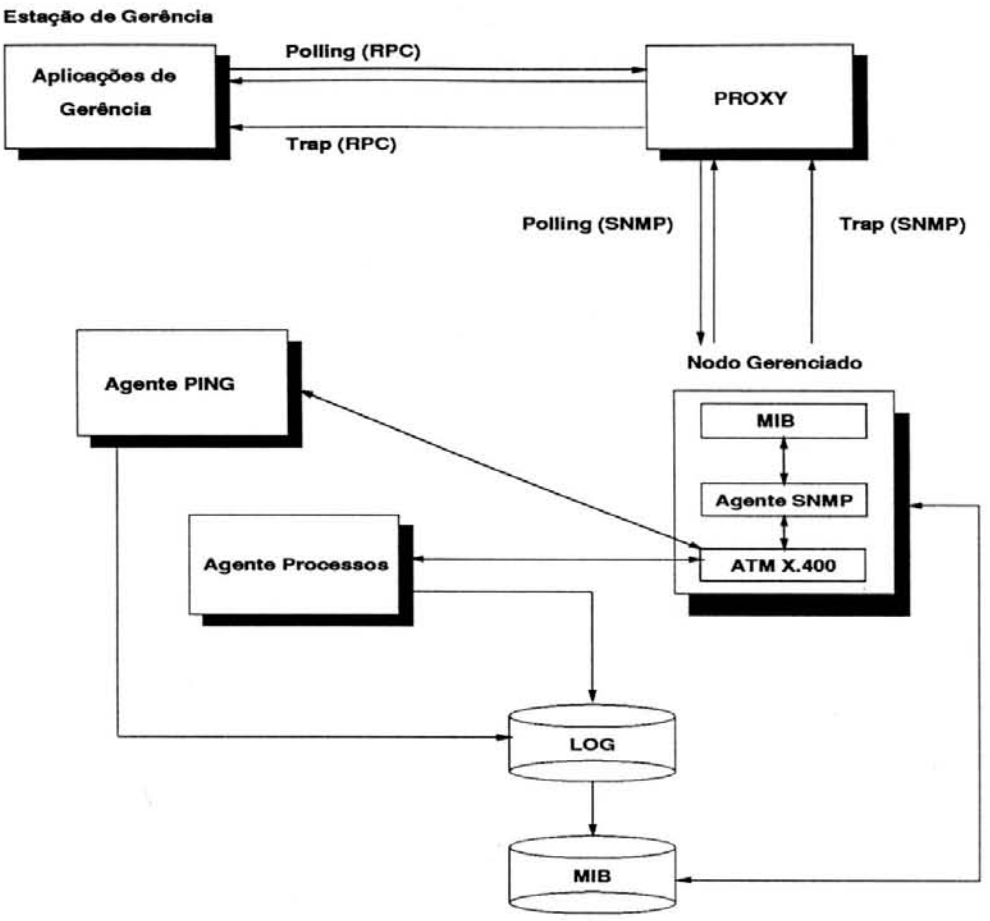

Figura 5.5: Interação do Agente SNMP com o *log* através da MIB

- Congestionamento;
- *Loops* de redirecionamento e roteamento;
- Conversões;
- Replicação de mensagens;
- Resolução de nomes;
- Problemas na submissão, entrega e transferência das mensagens.

Estas falhas são monitoradas no próprio código da aplicação, o PP, como os demais objetos gerenciados.

O SunManager possui um agente que executa as funções do comando **ping**. Porém como o objetivo do sistema é gerar registros sobre falhas que podem ser acessados pelo agente SNMP, optou-se pela implementação de um novo agente que automaticamente gera informações para o arquivo de *log* adequado.

5.4.2 Agente SNMP

O agente SNMP tem como função monitorar o objeto da gerência para fornecer ao gerente as informações sobre o conjunto de objetos gerenciados.

O agente, em linhas gerais, recebe uma PDU (*Protocol Data Unit*), decodifica-a, identifica o tipo de operação (*get*, *get-next*, *set*), recupera os dados, codifica-os em uma PDU para transmissão na rede e os envia em resposta à requisição.

A recuperação dos dados é feita por um módulo dedicado, isto é, para cada grupo da MIB-II existe um módulo específico para a obtenção dos dados. Da mesma forma a recuperação dos dados da MIB do correio eletrônico é implementada por um módulo específico. No anexo A-2 encontra-se a descrição do agente SNMP em SDL [CCITT 85].

O módulo *mail.c* contém os procedimentos que foram implementados para suportar a MIB de correio eletrônico. Existe dois tipos de procedimentos para recuperação de dados da MIB, um para recuperação de dados não tabulares e outro para recuperação de dados tabulares (tabelas). Há ainda outros dois procedimentos para execução de operações de *set*, um para dados não tabulares e outro para dados tabulares. Os algorítmos seguido por estes procedimentos, em linhas gerais, assemelham-se, havendo diferença apenas na identificação da instância dos objetos.

O agente SNMP recebe o pacote e executa as funções anteriormente descritas e então ativa o procedimento apropriado para o tratamento de dados tabulares

ou não tabulares, conforme a operação *get*, *get-next* ou *set*, passando a executar os seguintes passos:

- identificar a instância do objeto;
- recuperar (*get*) ou alterar (*set*) o valor do objeto;
- retornar o valor ao módulo principal para o encapsulamento na PDU de resposta.

A operação de *set* não foi implementada, pois os objetos da MIB não permitem operações de escrita, apenas leitura.

O valor dos objetos gerenciados é alterado sempre que algum evento ocorre no objeto de gerência, ou seja, os valores são constantemente atualizados e não apenas consultados mediante uma requisição do gerente. Por exemplo, a submissão de mensagens ao ATM causa a automática alteração dos valores dos objetos que refletem volume de mensagens, número de mensagens e outros relacionados.

A alteração automática ocorre porque o código para monitoração da aplicação foi inserido na aplicação. O PP foi recompilado, bem como o agente SNMP do ISODE, para suportar as alterações decorrentes da implementação do protótipo. O agente tem acesso aos objetos gerenciados por meio de memória compartilhada [STE 90].

Os procedimentos alterados no PP para a captura dos dados de gerência foram:

- canal *submit*: para obter informações sobre a submissão das mensagens, volume de mensagens e problemas na submissão;
- o canal qmgr: para obter informações sobre a fila do ATM, ativação de procedimentos e detecção de falhas;

- os canais X400 *in* e *out*: para obter informações sobre as mensagens propriamente ditas, sobre a geração de relatórios, sobre as mensagens de teste e sobre a transferênica e entrega das menagens;
- os procedimentos de associação: para gerenciar as associações.

A integração do agente SNMP com o SunNet Manager foi bastante fácil. O agente *proxy* do SunNet Manager aceita a adição de novos objetos ao seu *schema* sem necessidade de alterações em seu código e recompilação. Para que os objetos sejam reconhecidos pelos gerentes e pelo *proxy* deve-se somente reinicializar o SunNet Manager usando a opção **-i**, após adicionar os novos objetos no arquivo */usr/snm/agents/snmp.schema*. Esta opção provoca a leitura dos arquivos *schema* onde são definidos os objetos gerenciados por cada agente.

6 CONCLUSÃO

A gerência de um objeto, seja este componente físico da rede ou uma aplicação, passa pelas mesmas fases de avaliação e levantamento dos requisitos relevantes à gerência e implementação dos objetos gerenciados.

Para propor um conjunto de objetos que formam a MIB específica para um objeto de gerência é aconselhável que se faça estudos sobre:

- o ambiente em que está inserido o objeto;
- qual a função do objeto;
- quais as operações críticas para o ambiente;
- quais as informações necessárias para gerenciar o objeto em alto nível;
- o objeto detalhadamente;
- a especificação dos objetos gerenciados;
- definição dos processos de gerência, e
- a implementação dos objetos de gerência.

Com base nestas informações é possível formar consciência sobre os pontos importantes que devem ser representados na MIB através de objetos gerenciados.

Definida a MIB, deseja-se ter um meio de alimentar esta base de dados. Isto concretiza-se através da monitoração do objeto da gerência pela implementação de um agente dentro do paradigma de gerência de redes de computadores.

A fase seguinte à definição da MIB e anterior à implementação do agente concentra-se na decisão sobre as ferramentas a serem utilizadas para o desenvolvimento dos processos de gerência.

Estas foram as fases pelas quais passou o desenvolvimento deste trabalho. O passo mais difícil e demorado durante o processo de definição conceitual foi o levantamento dos requisitos relevantes à gerência do ATM, e por conseguinte, a definição dos objetos gerenciados que formaram a MIB.

A interoperabilidade entre os softwares não acarretou problemas, uma vez que são oferecidas ferramentas para a execução de tarefas trabalhosas como a tradução da MIB para o formato *schema* do SunNet Manager.

Ainda referente a interoperabilidade, o agente SNMP pode fornecer os dados gerenciados a qualquer outro gerente, de qualquer outro ambiente, desde que este fale SNMP e obedeça os critérios para autenticação. Dessa forma o agente não está subjugado somente a plataformas que utilizam SunNet Manager.

O agente SNMP do ISODE não oferece uma API amigável e não tem documentação. O conhecimento sobre o agente é adquirido através da bibliografia disponível sobre SNMP e através do estudo do código fonte do agente. A escolha do agente SNMP do ISODE para o modelo deve-se ao fato de já estar sendo empregado o ISODE no modelo, e à falta de espaço em disco para a adição de um novo pacote que implemente o agente.

A implementação ou inserção do código de gerência nos fontes do PP foi a parte mais trabalhosa da implementação. O PP é um software bastante complexo e grande. A documentação é um tanto confusa, e a falta de recursos computacionais dificultou a interação com a aplicação.

A inserção do código de gerência no fonte do PP e o uso de memória compartilhada foi a solução adotada, tanto por motivo de desempenho da aplicação quanto do agente. Outras técnicas poderiam ser empregadas, como por exemplo a monitoração dos pacotes X.400 na rede. Mas isso acarretaria a queda de desempenho do agente, uma vez que este deveria examinar todos os pacotes da rede para selecionar pacotes X.400.

Enfim, este trabalho é um passo na direção da gerência da aplicação de correio eletrônico, além do aperfeiçoamento dos conhecimentos sobre as entidades envolvidas e aplicação de conceitos teóricos.

BIBLIOGRAFIA

[BOR 92] BORENSTEIN, N.; FREED, N. In: MIME (Multipurpose Internet Mail Extensions). **Request For Comments 1341**. Jun. 1992.

[CAR 94] CARRILHO, Jóse A.; MADEIRA, Edmundo R. M. Um Esquema para o Gerenciamento do Protocolo FTP Baseado em Domínios. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 12., 1994, Curitiba, Paraná. **Anais...** Paraná: Larc, 1994. 765p.

[CAR 93] CARVALHO, Tereza C. M. B. et all. **Gerenciamento de Redes de Computadores - Uma abordagem de Sistemas Abertos.** São Paulo: Makron Books, 1993. 364p.

[CCITT 85] CCITT. **Functional Specification and Description Language (SDL)**. Geneva: Recommendations , Facsimile VI.11, Red Book, 1985.

[CCITT 89] CCITT. **Data Communication Networks:** Message Handling System. Geneva: ITU, 1989. 628p.

[CCITT 89a] CCITT. **Data Comminucation Networks:** Open System Interconnection (OSI) - Model and Notation, Service Definition. Geneva: ITU, 1989. 502p.

[COM 91] COMER, D. E. **Internetworking with TCP/IP:** Principles, Protocols, and Architeture. 2nd. Englewood Cliffs: Prentice-Hall, 1991. v.1, 347p.

[CRO 82] CROCKER, David H. In: Standard for the Format of ARPA Internet Text Messages. **Request for Comments 822**. Aug. 1982.

[KIL 86] KILLE, Steve. Mapping between X.400 and RFC 822. **Request For Commments 987**. Jun. 1986.

[KIL 90] KILLE, Steve. Mapping between X.400(1988) / ISO 10021 and RFC 822. **Request For Comments 1148**. Mar. 1990.

[KIL 91] KILLE, Steve; ONION, Julian. **The PP Manual**. [s.l.:s.n.], 1991. 206p.

[KIL 94] KILLE, Steve; FREED, N. Mail Monitoring MIB. **Request For Comments 1566**. Jan. 1994.

[KOC 90] KOCHAN, Stephen G.; WOOD, Patrick H. **Unix Networking**. Carmel: Hayden Books Unix System Library, 1990. chap. 3, p.49-91.

[McC 90] McCLOGHRIE, K.; ROSE, M. Structure and Identification of Management Information for TCP/IP-based Internets. **Request For Comments 1155**. May. 1990.

[McC 91] McCLOGHRIE, K.; ROSE, M. Management Information Base for Network Management of TCP/IP-based internets: MIB-II. **Request For Comments 1213**. Mar. 1991.

[McC 93] McCOY, Emily; FREIWIRTH, Ray. Email Management Requiremnets. **Working Draft**. 1993. 40p.

[POS 82] POSTEL, Jonathan B. Simple Mail Transfer Protocol. **Request For Comments 821**. Aug. 1982.

[REI 93] REINHARDT, Andy. Smarter E-Mail Is Comming. **Byte**. New York. International Edition. v. 18, n. 3, p.90-108, Mar. 1993.

[ROS 87] ROSE, Marshall T.; CASS, D. ISO Transport Service on top of the TCP. **Request for Comments 1006**. 1987.

[ROS 90] ROSE, Marshall T. **The Open Book:** A Practical Perspective on OSI. Englewood Cliffs: Prentice-Hall, 1990. 651p.

[ROS 91] ROSE, Marshall T. **The Simple Book**. Englewood Cliffs: Prentice-Hall, 1991. 347p.

[SCH 90] SCHOFFSTALL, M. Fredor, J., Davin, J. Case. A Simple Network Management Protocol (SNMP). **Request For Comments 1157**. May. 1990.

[SIL 92] SILVA, Ana C. Benso da; WESTPHALL, Carlos B. Implementação de Novos Agentes para Gerência de Redes de Computadores. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 11., 1992, Campinas, São Paulo. **Anais...** São Paulo: Larc, 1992. 756p. p.306-320.

[STE 90] STEVENS, W. Richard. In: **UNIX Networking Programming**. Englewood Cliffs: Prentice-Hall, 1990. 772p. p. 87-170.

[SUN 89] SunMicrosystem. **SunNet Manager - Installation and User's Guide**. Mountain View: Sun Microsystem Inc, 1988. 99p.

[SUN 89a] Sun Microsystem Inc. **Network Programming Guide**. Mountain View: Sun Microsystem Inc, 1989. 353p.

[WES 91] WESTPHALL, Carlos Becker **Conception et développement de l'architecture d'administration d'un réseau métropolitain**.Toulouse, 16 Juillet 1991. (Thèse de Doctorat nouveau régime. Université Paul Sabatier)

ANEXO A-1 MIB NO FORMATO DO SNMP

```
-- Mail-MIB Primeira experiencia


MAIL-MIB DEFINITIONS ::= BEGIN

IMPORTS
        enterprises, OBJECT-TYPE
                FROM RFC1155-SMI
        DisplayString
                FROM RFC1158-MIB;

-- Mail especific MIB

mail          OBJECT IDENTIFIER ::= { enterprises 100 }

-- the AdjacAtm group

adjacAtm OBJECT IDENTIFIER ::= { mail 1 }

atmadjTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF AtmadjEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { adjacAtm 1 }

atmadjEntry OBJECT-TYPE
        SYNTAX  AtmadjEntry
```

```
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { atmadjTable 1 }

AtmadjEntry ::=
        SEQUENCE {
            atmadjname
                DisplayString,
            atmadjaddr
                IpAddress,
            atmadjdomain
                DisplayString,
            atmadjstatus
                INTEGER
        }

atmadjname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmadjEntry 1 }

atmadjaddr  OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmadjEntry 2 }

atmadjdomain OBJECT-TYPE
        SYNTAX  DisplayString
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmadjEntry 3 }

atmadjstatus   OBJECT-TYPE
        SYNTAX  INTEGER {
                 busy(1),
                 down(2),
                 up(3)
        }
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmadjEntry 8 }

-- the atm local group

atmlocal OBJECT IDENTIFIER ::= { mail 2 }

atmlocname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 1 }

atmlocaddr  OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 2 }
```

```
atmlocdomain OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 3 }

applocname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 4 }

protlocname OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 5 }

protlocver  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 6 }

atmlocstatus OBJECT-TYPE
        SYNTAX  INTEGER {
                busy(1),
                down(2),
                up(3)
        }
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 7 }

atmqueueIns  OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 9 }

atmqueueOuts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 10 }

atmdateStart OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 11 }

atmlastfault OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 12 }

atmfaultTotals OBJECT-TYPE
        SYNTAX  Counter
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 13 }

atmactivetime   OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 14}

atminactivetime OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 15}

submitedMsgtots OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 16 }

reporttotals    OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 17 }

probetotals     OBJECT-TYPE
        SYNTAX  Counter
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 18 }

trasfMsgtots OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 19 }

storedMsgtots OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 20 }

rejectMsgtots OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 21 }

submitedMsgVols OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 22 }

transfMsgVols OBJECT-TYPE
        SYNTAX  Counter
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 23 }

storedMsgVols OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmlocal 23 }


-- Message group

atmMesg OBJECT IDENTIFIER ::= { mail 3 }

msgTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF MsgEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { atmMesg 1 }

msgEntry  OBJECT-TYPE
        SYNTAX  MsgEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { msgTable 1 }

MsgEntry  ::=
        SEQUENCE {
            msgId
```

```
    INTEGER,
msgsubtime
    DisplayString,
msgdeltime
    DisplayString,
atmFromAdd
    IpAddress,
atmToAdd
    IpAddress,
msgsize
    INTEGER,
originorname
    DisplayString,
reciporname
    DisplayString,
priority
    INTEGER,
implconverprohib
    INTEGER,
convlossprohib
    INTEGER,
defdeltime
    DisplayString,
latestdeltime
    INTEGER,
physforwarprohib
    INTEGER,
origreporreq
    INTEGER,
origencodtype
```

```
                DisplayString,
            submissionstat
                INTEGER,
            deliverystat
                INTEGER,
            transfstat
                INTEGER,
            reasonreject
                INTEGER,
            nondeliveryreason
                INTEGER,
            nondeldiagnostic
                INTEGER,
            tracemsg
                DisplayString,
            content
                DisplayString
}

msgId  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 1 }

msgsubtime  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 2 }
```

```
msgdeltime  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 3 }

atmFromAdd  OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 6 }

atmToAdd  OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 7 }

msgsize  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 8 }

originorname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 9 }
```

```
reciporname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 10 }

priority  OBJECT-TYPE
        SYNTAX  INTEGER {
                urgent(1),
                non-urgent(2),
                normal(3)
        }
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 11 }

implconverprohib  OBJECT-TYPE
        SYNTAX  INTEGER {
                prohibited(1),
                allowed(2)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 12 }

convlossprohib  OBJECT-TYPE
        SYNTAX  INTEGER {
                prohibited(1),
                allowed(2)
```

```
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 13 }

defdeltime  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 14 }

latestdeltime  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 15 }

physforwarprohib  OBJECT-TYPE
        SYNTAX  INTEGER {
                prohibited(1),
                allowed(2)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 16 }

origreporreq  OBJECT-TYPE
        SYNTAX  INTEGER {
                no-report(1),
                non-delivery-report(2),
```

```
                report(3)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 17 }

origencodtype  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 18 }

submissionstat  OBJECT-TYPE
        SYNTAX  INTEGER {
                success(1),
                not-success(2)
        }
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 19 }

deliverystat  OBJECT-TYPE
        SYNTAX  INTEGER {
                success(1),
                not-success(2)

        }
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 20 }
```

```
transfstat  OBJECT-TYPE
        SYNTAX  INTEGER {
                success(1),
                not-success(2)
        }
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 21 }

reasonreject  OBJECT-TYPE
        SYNTAX  INTEGER {
                submission-ctl-violated(1),
                element-not-subscried(2),
                originator-invalid(3),
                recipient-imp-specified(4),
                incosistent-request(5),
                security-error(6),
                unsupported-crtic-function(7),
                remote-bind-error(8)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 22 }

nondeliveryreason OBJECT-TYPE
        SYNTAX  INTEGER {
                tranfer-failure(1),
                unable-to-transfer(2),
                conversion-not-performed(3),
```

```
                physical-redent-nperformed(4),
                physical-del-nperformed(5),
                restrict-delivery(6),
                directory-operation-unsucc(7)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 23 }

nondeldiagnostic OBJECT-TYPE
        SYNTAX  INTEGER {
                unrecognized-or-name(1),
                ambiguous-or-name(2),
                mts-congestion(3),
                lopp-detected(4),
                recipient-unavailable(5),
                maximum-time-expired(6),
                encoded-information-type-unsupp(7),
                content-too-long(8),
                conversion-impractical(9),
                implicit-conversion-prohibited(10),
                implicit-conversion-nsubscribed(11),
                invalid-arguments(12),
                content-syntax-error(13),
                size-constraint-violation(14),
                protocol-violation(15),
                content-type-nsupoorted(16),
                too-many-recipients(17),
                no-bilateral-agreement(18),
                unsupported-critical-function(19),
```

```
                conversion-with-loss-prohibited(20),
                line-too-long(21),
                page-split(22),
                pictorial-symbol-loss(23),
                alphabetic-character-loss(24),
                multiple-information-loss(25),
                recipient-reassignament-prohibited(26),
                redirection-loop-detected(27)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 24 }

tracemsg  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 25 }

content  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { msgEntry 26 }


-- Probe Group
msgprobegrp OBJECT IDENTIFIER ::= { mail 4 }

msgprobeTable OBJECT-TYPE
```

```
        SYNTAX  SEQUENCE OF MsgprobeEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { msgprobegrp 1 }

probeEntry OBJECT-TYPE
        SYNTAX  MsgprobeEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { msgprobeTable 1 }

MsgprobeEntry ::=
        SEQUENCE {
           origprobname
                DisplayString,
           recipprobname
                DisplaySrting,
           probid
                INTEGER,
           submissiontime
                INTEGER,
           repreasreject
                DisplayString
}

origprobname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { probeEntry 1 }
```

```
recipprobname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { probeEntry 2 }

probid  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { probeEntry 3 }

submissiontime  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { probeEntry 4 }

submissionstatus  OBJECT-TYPE
        SYNTAX  INTEGER {
                success(1),
                not-success(2)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { probeEntry 5 }

repreasreject  OBJECT-TYPE
        SYNTAX  INTEGER {
```

```
                submission-ctl-violated(1),
                element-not-subscried(2),
                originator-invalid(3),
                recipient-imp-specified(4),
                incosistent-request(5),
                security-error(6),
                unsupported-crtic-function(7),
                remote-bind-error(8)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { probeEntry 6 }

-- Report group

reportgrp OBJECT IDENTIFIER ::= { mail 5 }

reportTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF ReportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { reportgrp 1 }

reportEntry OBJECT-TYPE
        SYNTAX  ReportEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { reportTable 1 }

ReportEntry ::=
```

```
        SEQUENCE {
            reportid
                INTEGER,
            reprecipername
                DisplayString,
            result
                INTEGER
}

reportid  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { reportEntry 1 }

reprecipername  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { reportEntry 2 }


result  OBJECT-TYPE
        SYNTAX  INTEGER {
                success(1),
                not-success(2)
}
        ACCESS  read-only
        STATUS  mandatory
        ::= { reportEntry 5 }
```

```
-- Associations group

atmAssoc OBJECT IDENTIFIER ::= { mail 6 }

inAssocTotals  OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 1 }

outAssocTotals  OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 2 }

lastInAssocs  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 3 }

lastOutAssocs  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 4 }
```

```
rejectInAssocTotals  OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 5 }

rejectOutAssocTotals  OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 6 }

abortedAssocTotals  OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 7 }

p1maxinassoc    OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 8 }

p1maxoutassoc    OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { atmAssoc 9 }
```

```
assocTable   OBJECT-TYPE
        SYNTAX  SEQUENCE OF AssocEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { atmAssoc 8 }

assocEntry  OBJECT-TYPE
        SYNTAX  AssocEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { assocTable 1 }

AssocEntry  ::=
        SEQUENCE {
            associd
                INTEGER,
            initAdd
                IpAddress,
            acceptorAdd
                IpAddress,
            assocTime
                DisplayString,
            transfmsgVolume
                INTEGER,
            assocStatus
                INTEGER,
            assocRejReason
                DisplayString
}
```

```
associd  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { assocEntry 1 }

initAdd  OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { assocEntry 2 }

acceptorAdd  OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { assocEntry 3 }

assocTime  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { assocEntry 4 }

transfmsgVolume  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { assocEntry 5 }
```

```
assocStatus  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { assocEntry 6 }

assocRejReason  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { assocEntry 7 }


-- The conversion group

conversion OBJECT IDENTIFIER ::={ mail 7 }

convertable OBJECT-TYPE
        SYNTAX  SEQUENCE OF ConvEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { conversion 1 }

converEntry     OBJECT-TYPE
        SYNTAX  ConvEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { convertable 1 }

ConvEntry ::=
```

```
        SEQUENCE {
                convtype
                   DisplayString
        }

convtype        OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { conventry 1 }

-- encoded information types  group
-- encoded information types  group

eits    OBJECT IDENTIFIER ::= { mail 8 }

eittable        OBJECT-TYPE
        SYNTAX  SEQUENCE OF EitEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { eits 1 }

eitEntry     OBJECT-TYPE
        SYNTAX  EitEntry
        ACCESS  not-accessible
        STATUS  mandatory
        ::= { eittable 1 }

EitEntry ::=
        SEQUENCE {
```

```
                eittype
                  DisplayString
        }

eittype         OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { eitEntry 1 }


-- the log group

logfile  OBJECT IDENTIFIER ::= { mail 9 }

data     OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { logfile 1 }

atmname  OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { logfile 2 }

problema OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
```

```
        STATUS  mandatory
        ::= { logfile 3 }

solucao OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { logfile 4 }

responsavel OBJECT-TYPE
        SYNTAX  DisplayString
        ACCESS  read-only
        STATUS  mandatory
        ::= { logfile 5 }

END
```

ANEXO A-2 ESPECIFICAÇÃO DOS AGENTES EM SDL

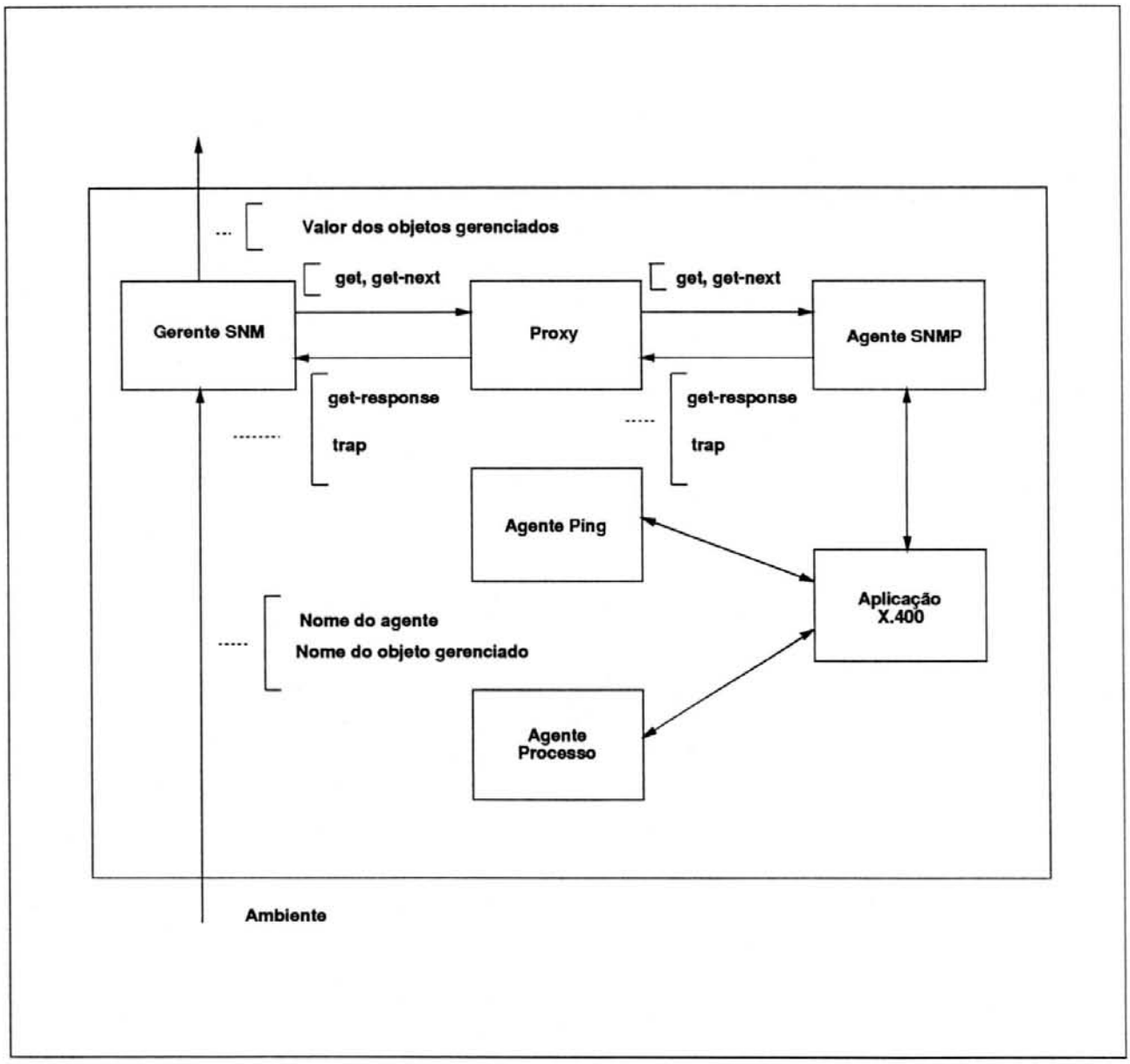

Figura A-2.1: Visão geral dos processos

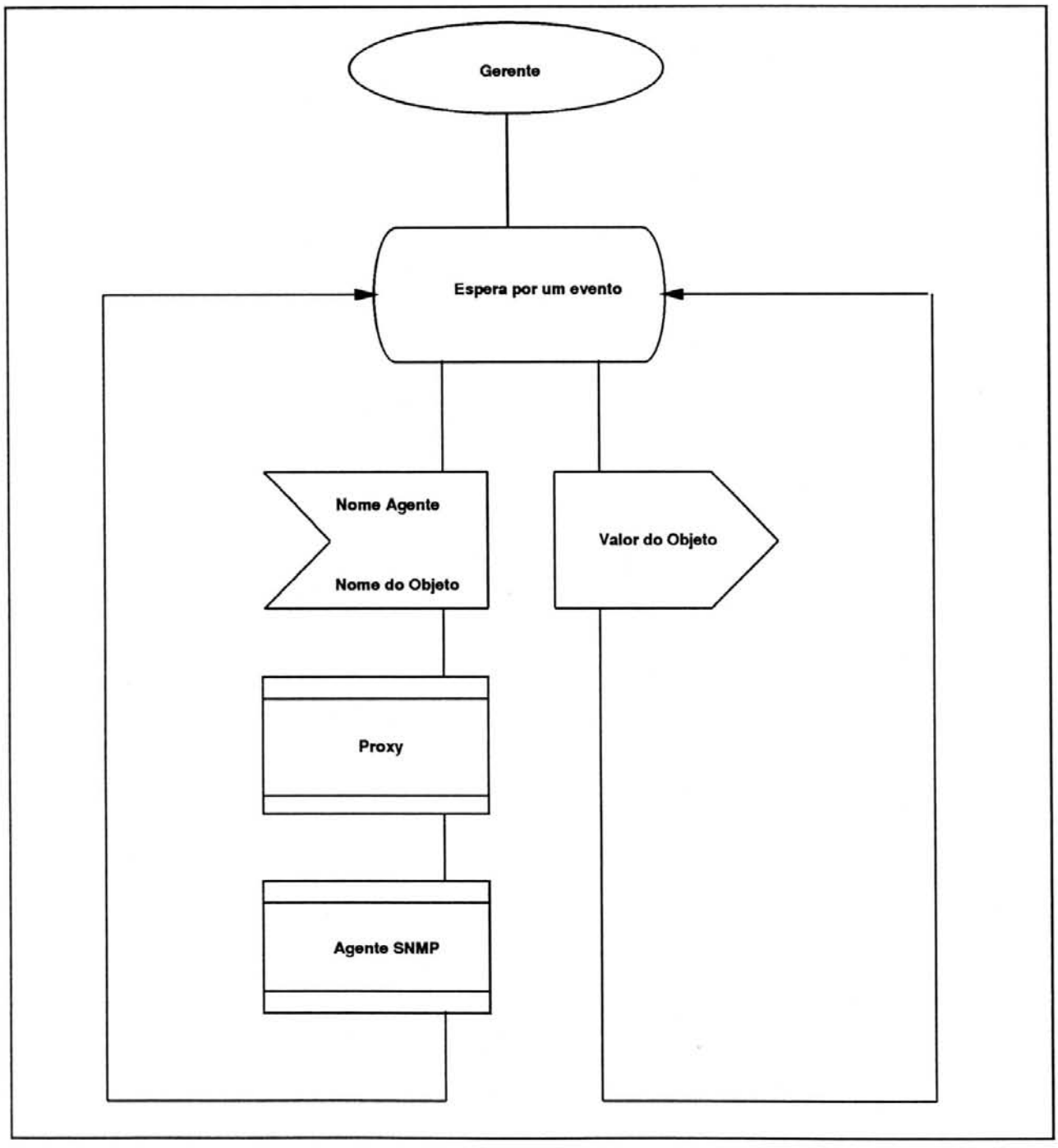

Figura A-2.2: Descrição do gerente

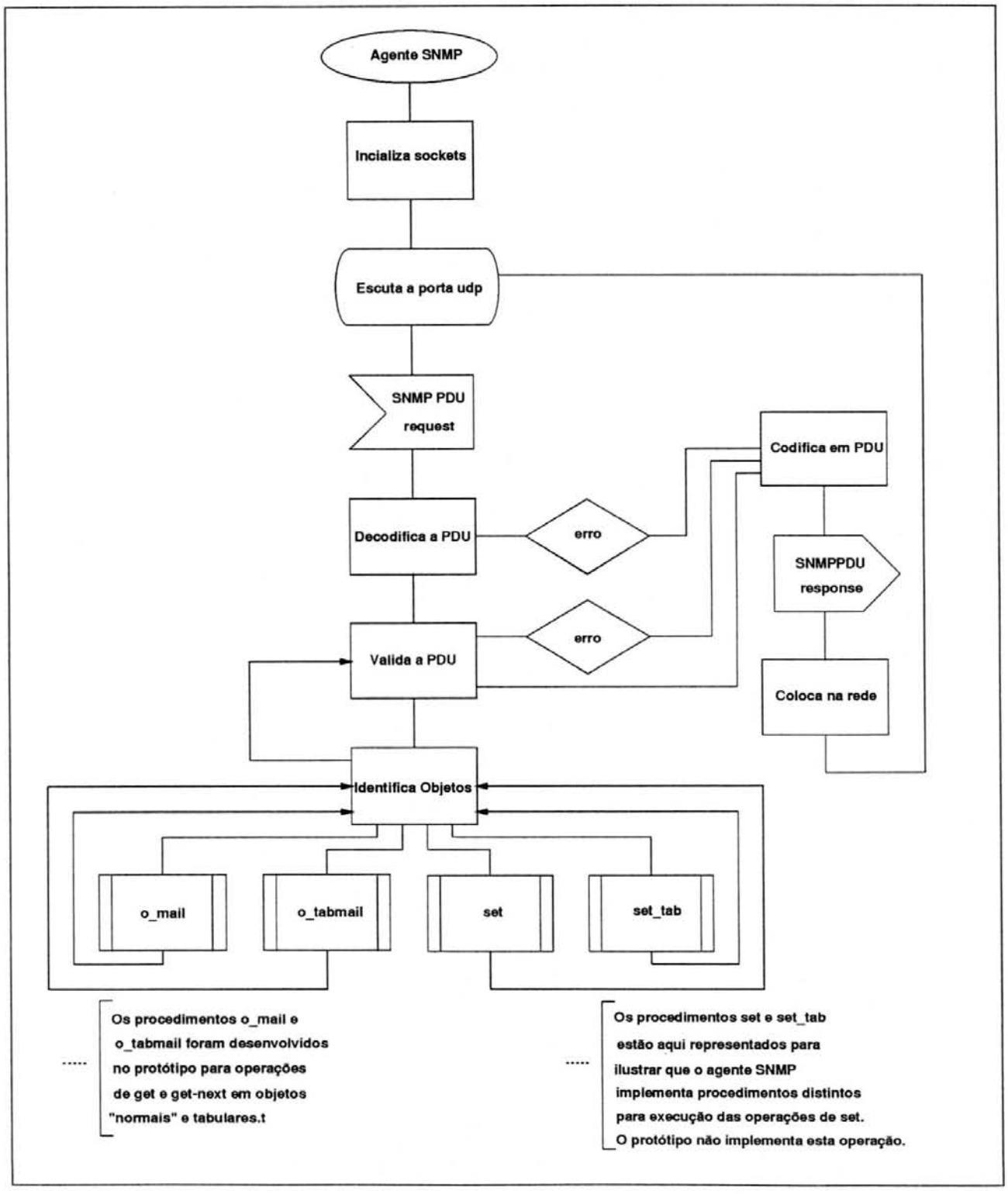

Figura A-2.3: Descrição do agente SNMP

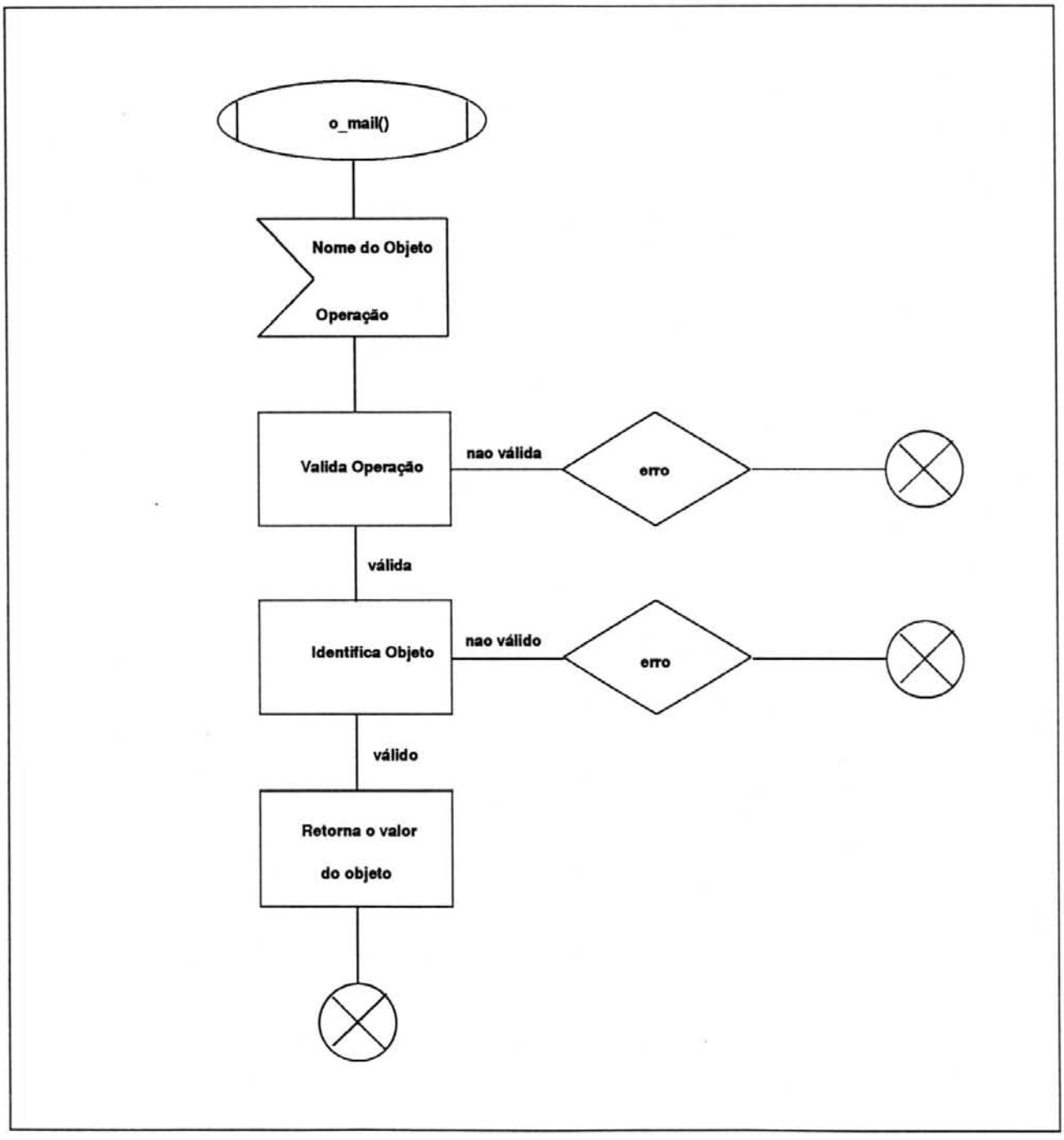

Figura A-2.4: Descrição do procedimento o_mail()

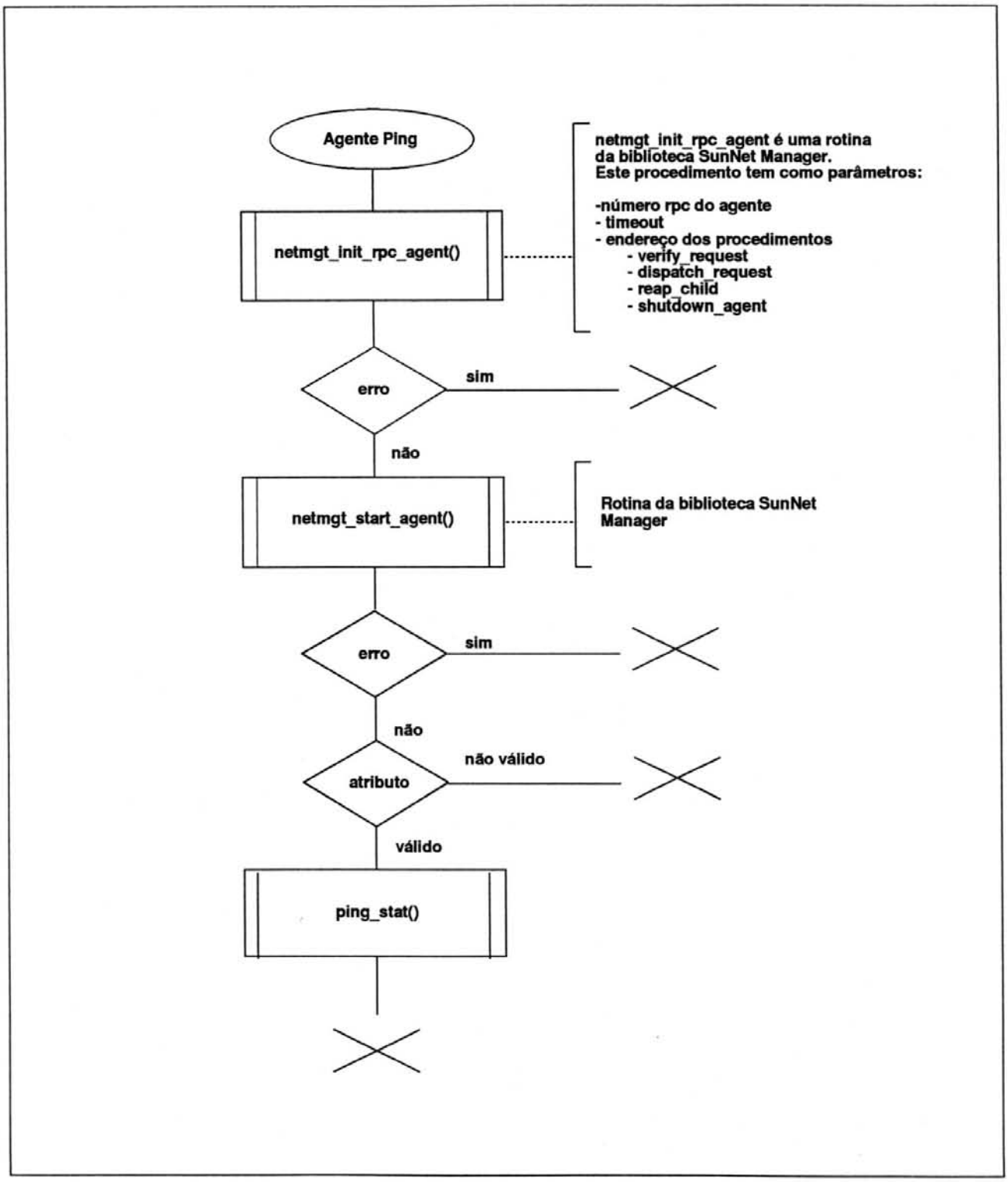

Figura A-2.5: Descrição do agente ping

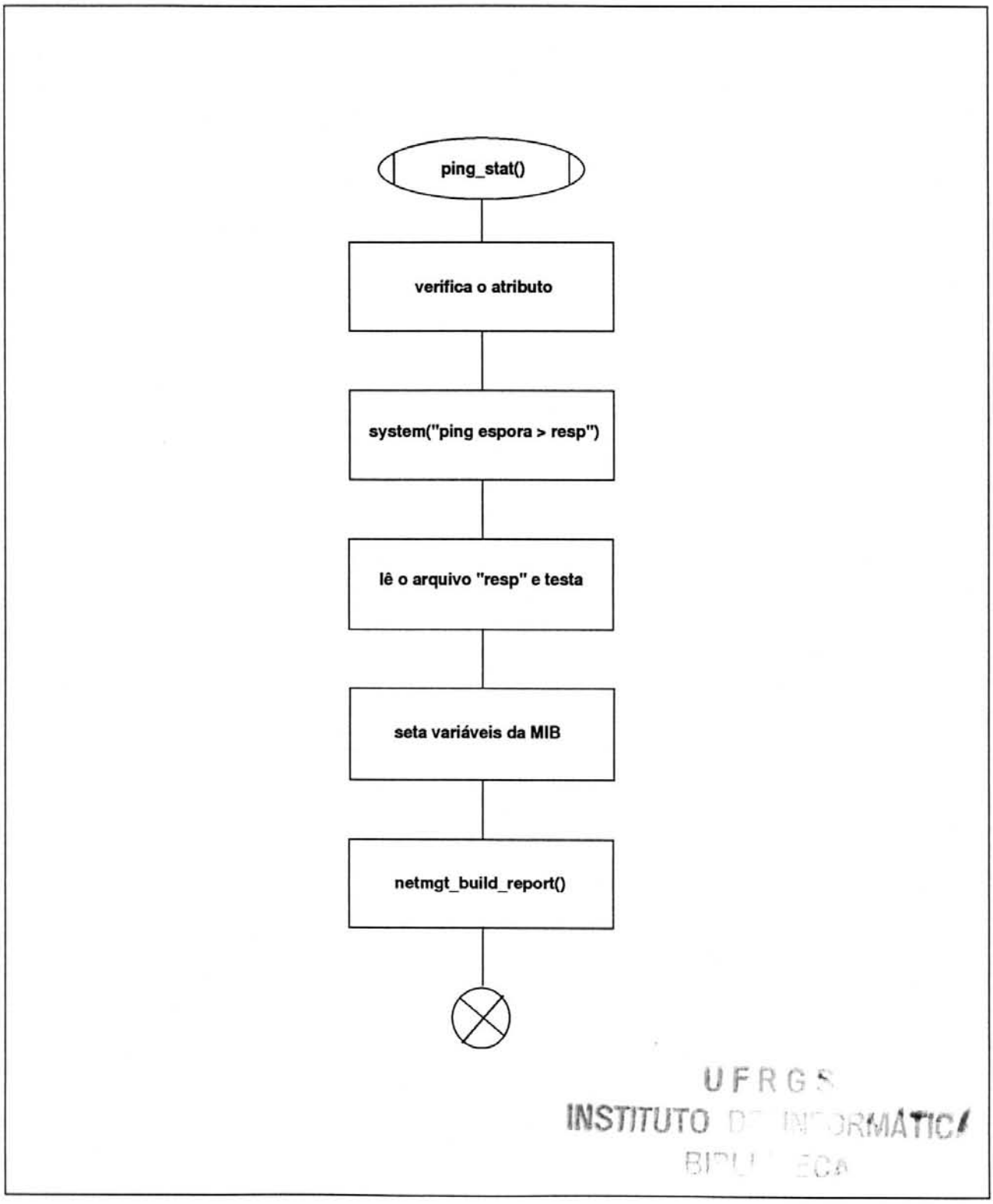

Figura A-2.6: Descrição do procedimento ping_stat()

Informática  UFRGS

Uma Proposta para Gerência de Correio Eletrônico.

Dissertação apresentada aos Senhores:

Prof. Dr. Carlos Alberto Heuser

Prof. Dr. Edson dos Santos Moreira (USP/São Carlos)

Prof. Dr. Raul Fernando Weber

Vista e permitida a impressão.
Porto Alegre, 15/09/95

Profa. Dra. Liane Margarida Rockenbach Tarouco,
Orientador.

Prof José Palazzo Moreira de Oliveira
Coordenador do Curso de Pós-Graduação
em Ciência da Computação - CPGCC
Instituto de Informática - UFRGS