

Universidade Federal de Santa Catarina - UFSC
Centro Tecnológico
Departamento de Informática e Estatística
Pós-graduação em Ciência da Computação.

Serviços de Atendimento ao Usuário do Sistema de Telecomunicações
Um modelo de informações para sua gerência

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

Por
Alexandre Peixoto Ferreira

Prof. Dr. João Bosco da Mota Alves
Orientador

Profa. Elizabeth S. Specialski
Co-orientadora

Florianópolis, Maio de 1998

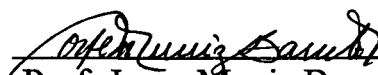
Serviços de Atendimento ao Usuário do Sistema de Telecomunicações
Um modelo de informações para sua gerência

Alexandre Peixoto Ferreira

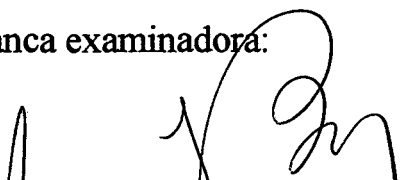
Esta dissertação foi julgada adequada para obtenção do título de


Mestre em Ciências da Computação

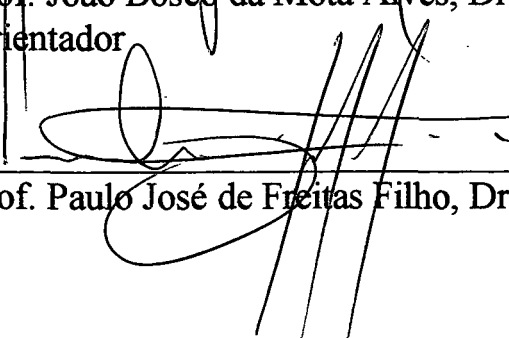
Especialidade em Redes de Computadores e aprovada em sua forma final pelo
programa de Pós-graduação em Ciência da Computação


Prof. Jorge Muniz Barreto, Dr.
Coordenador do Curso

Banca examinadora:


Prof. João Bosco da Mota Alves, Dr.
Orientador


Prof. Elizabeth S. Specialski, M. Sc.
Co-orientadora


Prof. Paulo José de Freitas Filho, Dr.


Prof. Alexandre Ramos, M. Comp.

Dedico este trabalho à minha esposa Adriana que sempre soube compreender e ajudar .

Agradecimentos

Agradeço a todos que direta ou indiretamente colaboraram para que este trabalho pudesse ser realizado.

Em especial gostaria de agradecer a professora e grande mestre Elizabeth Specialski pelo ensinamentos, apoio, incentivo, perseverança e principalmente orientação que permitiram a existência deste trabalho.

Ao professor Dr. João Bosco da Mota Alves sem o qual este trabalho não existiria.

Aos colegas da Flug Telemática e Automação, pela colaboração, apoio e principalmente pelo espírito de equipe que foi muito importante.

E ao grande amigo Gastão Eduardo de Figueiredo Júnior, pelo incentivo e amizade sempre constantes.

E a Adriana Beal, minha esposa, o mais importante agradecimento pela paciência e palavras de apoio sempre presentes que foram e serão sempre muito importantes.

Sumário

1. INTRODUÇÃO	1
2. GERÊNCIA DE REDES DE TELECOMUNICAÇÕES E TMN	5
2.1. A Arquitetura TMN	5
2.2. Modelo de Informação TMN	12
3. SERVIÇOS DE ATENDIMENTO AO USUÁRIO	15
3.1. Introdução	15
3.2. SIPT	18
3.3. A Plataforma de Serviços Dígitro	23
3.3.1. Estrutura física da plataforma de serviços	24
3.3.2. Um serviço na plataforma de serviços AXS/20	26
4. MODELO DE INFORMAÇÃO	29
4.1. Metodologia de Especificação de uma interface Q3	29
4.1.1. Tarefa 0: Gerar diretrizes	34
4.1.2. TIB 0: Diretrizes	34
4.1.3. Tarefa 1: Descrever os serviços de gerência TMN e seus objetivos pela perspectiva do usuário	34
4.1.4. TIB A: Objetivos e serviços de gerência TMN	34
4.1.5. Tarefa 2: Descrever o contexto da gerência TMN	36
4.1.6. TIB B: Papéis da gerência TMN, recursos de telecomunicações e funções de gerência TMN	37
4.1.7. TIB X: Modelos de informação genéricos e específicos de uma tecnologia	38
4.1.8. Tarefa 3: Modelagem de Objetos	38
4.1.9. TIB C: Biblioteca de informações de gerência	39
4.1.10. TIB D: Diagrama de entidade-relacionamento	39
4.1.11. Tarefa 4: Consolidar as informações disponíveis	39
4.1.12. Tarefa 5: Definir a organização da informação de gerência	40
4.1.13. TIB E: Organização da Informação de gerência	41
4.1.14. Tarefa 6: Definir os requisitos da comunicação	41
4.1.15. TIB F: Requisitos de comunicação	41
4.1.16. Tarefa 7: Preparar o documento para as tarefas de especificação de protocolos	41
4.1.17. TIB G: Perfis funcionais da TMN	42
4.2. Serviços de Gerência	42
4.3. Funções de Gerência	44
5. APLICAÇÃO DA METODOLOGIA	45
5.1. Tarefa 0: Gerar diretrizes	45
5.2. TIB 0: Diretrizes	45
5.3. Tarefa 1: Descrever os serviços de gerência TMN e seus objetivos pela perspectiva do usuário	46
5.4. TIB A: Objetivos e serviços de gerência TMN	46
5.5. Tarefa 2: Descrever o contexto da gerência TMN	46

5.6. TIB B: Papéis da gerência TMN, recursos de telecomunicações e funções de gerência TMN.....	47
5.6.1. Contexto de gerência:	47
5.7. TIB X: Modelos de informação genéricos e específicos de uma tecnologia	50
5.8. Tarefa 3: Modelagem de Objetos.....	50
5.9. TIB C: Biblioteca de informações de gerência.....	50
5.10. TIB D: Diagrama de entidade-relacionamento	52
5.11. Tarefa 4: Consolidar as informações disponíveis.....	55
5.12. Tarefa 5: Definir a organização da informação de gerência	57
5.13. TIB E: Organização da Informação de gerência.....	58
5.14. Tarefa 6: Definir os requisitos da comunicação.....	58
5.15. TIB F: Requisitos de comunicação	58
5.16. Tarefa 7: Preparar o documento para as tarefas de especificação de protocolos	59
5.17. TIB G: Perfis funcionais da TMN.....	59
5.18. Implementação e Teste do Modelo de Informação.....	60
6. CONCLUSÕES	61
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	64
ANEXO A: DESCRIÇÃO DO MODELO DE INFORMAÇÃO GDMO	68
ANEXO B: DESCRIÇÃO DO MODELO DE INFORMAÇÃO ASN.1.....	93

Índice de Figuras

FIGURA A: RELACIONAMENTO ENTRE A TMN E A REDE DE TELECOMUNICAÇÕES.....	6
FIGURA B: EXEMPLO DE UM AMBIENTE DE GERÊNCIA TMN.....	7
FIGURA C: PONTOS DE INTERFACE DA ARQUITETURA FÍSICA DA TMN.....	11
FIGURA D: CAMADAS DE GERENCIAMENTO TMN.....	12
FIGURA E: RELACIONAMENTO ENTRE OBJETOS GERENCIADOS E RECURSOS FÍSICOS.....	13
FIGURA F: ESTRUTURA DE UM SISTEMA SIPT.....	18
FIGURA G: ESTRUTURA DA CENTRAL AXS/20 EM UM SISTEMA SIPT.....	24
FIGURA H: ESTRUTURA DE UM SERVIÇO NA PLATAFORMA AXS/20.....	26
FIGURA I: RELACIONAMENTO ENTRE CONCEITOS DE ESPECIFICAÇÃO TMN.....	29
FIGURA J: RELACIONAMENTO ENTRE SERVIÇOS E FUNÇÕES DE GERÊNCIA TMN.....	31
FIGURA K: METODOLOGIA DE ESPECIFICAÇÃO DE INTERFACES TMN.....	33
FIGURA L: DIAGRAMA DE SERVIÇOS DE GERÊNCIA POR LLA E ÁREA FUNCIONAL.....	43
FIGURA M: DIAGRAMA ER AXS (ALARMES).....	52
FIGURA N: DIAGRAMA ER AXS (ESTRUTURA FÍSICA).....	53
FIGURA O: DIAGRAMA ER AXS (SERVIÇO).....	53
FIGURA P: DIAGRAMA ER DESCREVENDO A RELAÇÃO DE HERANÇA.....	54
FIGURA Q: DIAGRAMA ER DESCREVENDO A RELAÇÃO DE HERANÇA.....	55

Índice de Tabelas

TABELA 1 SERVIÇOS DE GERÊNCIA TMN.....	42
TABELA 2 FUNÇÕES DE GERÊNCIA E CLASSES QUE A SUPOORTAM.....	56
TABELA 3 CARACTERÍSTICAS DA INTERFACE PARA A GERÊNCIA DE CONFIGURAÇÃO.....	58
TABELA 4 CARACTERÍSTICAS DA INTERFACE PARA A GERÊNCIA DE FALHAS	59

Lista de Abreviaturas

API	Application Program Interface
ASN.1	Abstract Syntax Notation One
CCITT	Consultative Committee for International Telegraph and Telephone
CMIS	Common Management Information Service
CPA-T	Central de Programa Armazenado temporal
DC	Direct Current
DCN	Data Communications Network
DER	Diagrama Entidade-Relacionamento
DN	Distinguished name
GDMF	Guidelines for the Definition of Management Functions
GDMO	Guidelines for the Definition of Managed Objects
GDMS	Guidelines for the Definition of Management Services
GNIM	Generic Network Information Model
ISO	International Standards Organization
ITU-T	International Telecommunications Union - Telecommunications Standartization Section
LLA	Logical Layered Architecture
MD	Mediation Device
MFC	Multi-frequencial Compelida
MIS-User	Management Information Service User
MIT	Management Information Tree
MO	Managed Object
MOCS	Managed Object Conformance Statement
NE	Network Element
NMF	Network Management Forum
OAM	Operação, Administração, Manutenção
OAM&P	Operação, administração, manutenção e provisionamento
OS	Operations System
OSI	Open Systems Interconnection
PA	Posição de atendimento
QA	Q – Adaptor
RDN	Relative Distinguished Name
RTPC	Rede de Telefonia Pública Comutada
SS#7	Sistema de Sinalização por canal comum número 7
SIPT	Sistema Integrado de Posições de Tráfego
SMI	Structure of Management Information
SMF	Systems Management Functions
SMK	Shared Management Knowledge
SPT	Sistema Público de Telecomunicações
TIB	Task Information Base
TMN	Telecommunications Management Network
WS	Workstation

XMP
XOM

X/Open Management Protocols
X/Open Abstract Data Manipulation

Resumo

Atualmente, a TMN tem se tornado um recurso de grande importância dentro do sistema de telecomunicações. O aumento da eficiência da operação, administração e manutenção da rede em um ambiente competitivo é vital para as empresas operadoras e seus fornecedores.

Este trabalho trata sobre o desenvolvimento de um modelo de informação de gerência para os serviços de atendimento ao usuário do sistema de telecomunicações, seguindo a metodologia de especificação de interfaces Q3 definida pelo ITU-T na Recomendação M.3020.

Este modelo de informação visa permitir que os serviços de atendimento sejam gerenciados por um ambiente TMN através de uma interface de gerência padronizada, obtendo as vantagens associadas a esse ambiente de gerência.

Abstract

Nowadays, the TMN has become a tool of great importance for the telecommunications systems. The increase of efficiency of the operations, administration and maintenance of the network in a competitive environment is vital for the operators and it's suppliers.

This work is about the development of a management information model of the customer-oriented services of the telecommunications systems, using the specification methodology of Q3 interfaces defined by ITU-T in the recommendation M.3020.

This management information model has the objective to allow this customer oriented services to be managed by a TMN using a standard management interface, thus getting the advantages associated with this management environment.

1. INTRODUÇÃO

O sistema de telecomunicações tem se tornado um dos serviços mais importantes prestados à comunidade. As facilidades de comunicação hoje disponíveis aos usuários superam em muito aquelas existentes há poucos anos atrás. Hoje é possível realizar-se chamadas telefônicas automaticamente em praticamente todos os países do mundo. Com os telefones celulares, o serviço de telecomunicações acompanha seus usuários no seu dia-a-dia. Com o aparecimento e popularização dos telefones via satélite, o serviço os acompanha pelo mundo.

Quanto à transmissão de dados, estamos vendo recordes de velocidades sendo quebrados diariamente. Já temos a capacidade de enviar dezenas de gigabits por segundo em uma única fibra de vidro e dezenas de megabits por segundo por um simples par telefônico. Gigabits por segundo era, há pouco tempo, praticamente a capacidade agregada de transmissão das redes de dados de países inteiros. Estas velocidades estão cada vez mais disponíveis ao usuário em geral, e com o aparecimento da Internet e das redes locais, as empresas e usuários estão fazendo bom uso delas. As necessidades de transmissão de dados das empresas e usuários em geral estão aumentando de forma exponencial, exigindo maiores velocidades e menores custos.

As empresas atualmente utilizam maciçamente os serviços de telecomunicações para a realização de seus negócios e realização de suas atividades, e portanto estes serviços são cada vez mais fundamentais à sua sobrevivência em um mercado competitivo. Estamos na era da informação e, se a informação é o sangue da sociedade, a rede de telecomunicações é o seu sistema cardiovascular.

O sistema de telecomunicações tem hoje um grande desafio, que é prestar uma diversidade de serviços a um número de consumidores cada vez maior, a exigir custos menores e qualidade. A infra-estrutura necessária para prestar estes serviços é cada vez mais complexa e diversificada, o que praticamente impede que um único fabricante tenha todos os componentes necessários, levando a uma quase obrigatoriedade de se ter uma gama de fornecedores.

A digitalização da rede de telecomunicações possibilitou um enorme ganho de escala. As centrais de comutação tiveram a sua capacidade ampliada e seu tamanho físico reduzido. Na transmissão, circuitos que permitiam a existência de uns poucos canais de voz hoje atingem centenas a milhares de canais. Mas a disponibilidade dos serviços foi afetada por essa mudança tecnológica. Os sistemas analógicos, devido a sua natureza distribuída, apresentam uma degradação dos serviços quando existe uma falha; já os sistemas digitais, apesar de bem mais confiáveis, apresentam falhas que podem provocar a parada completa do sistema, tornando os serviços que utilizam estes recursos indisponíveis. Este fenômeno exige uma observação mais constante dos equipamentos, porque se as falhas são mais raras, tendem a ser mais catastróficas.

O mercado de telecomunicações tem exigido das empresas operadoras ações para reduções de custo e aumento da qualidade do serviço prestado. A automação da supervisão dos equipamentos e gerência dos serviços foi uma das formas utilizadas para racionalizar o uso do pessoal na operação, administração e manutenção da rede. Este processo começou na década de 60, com os sistemas de telesupervisão, destinados a permitir a observação remota dos alarmes existentes nos equipamentos desatendidos.

Com as centrais digitais, que possuem maiores capacidades de supervisão e configuração, a existência de sistemas centralizados de supervisão e controle tornou-se possível, mas com a restrição de serem orientados a um fabricante e normalmente a um tipo de equipamento. Isso reduziu custos, mas implicava ainda a necessidade de um número elevado de sistemas centralizados e a correspondente numerosa equipe treinada no uso destes sistemas.

O ITU-T (*International Telecommunications Union, Telecommunications Standardization Section*) iniciou então um trabalho de definição de uma arquitetura de gerência que tivesse como objetivos o planejamento, provisionamento, instalação, operação e administração e manutenção integrada dos serviços e redes de telecomunicações. Esta arquitetura de gerência foi baseada nos conceitos de gerência de redes de computadores em desenvolvimento pela ISO (*International Standards Organization*). Desse trabalho surgiu um conjunto de recomendações [M.3000] que definiram a TMN (*Telecommunications Management Network*) ou Rede de Gerência de Telecomunicações.

Um dos passos que permite que uma TMN gerencie um sistema ou um serviço é a criação de um modelo de informação que o descreva. Essa descrição deve seguir os padrões definidos pelas recomendações do ITU-T [X.720, X.722]. O objetivo do presente trabalho é a obtenção de um modelo de informação para os serviços de atendimento ao usuário seguindo a metodologia preconizada pelo ITU-T, este modelo deve prover funcionalidades para a gerência de falhas e de configuração, mas ter características genéricas permitindo que com poucas alterações obtenha-se um modelo que atenda as necessidades de outras gerências.

Este modelo de informações fornece uma visão padronizada para os serviços de atendimento, e esta padronização simplifica os conhecimentos necessários para a sua gerência. Isto gera benefícios, tais como: permitir a centralização da gerência, porque não exige que os operadores conheçam as especificidades de cada equipamento ou serviço; facilita a construção de aplicações que integrem a gerência dos serviços de atendimento com outras funcionalidades ou sistemas; etc.

Os sistemas de gerência normalmente são implantados visando atender as necessidades mais urgentes da empresa. A primeira necessidade que a empresa percebe é obter uma visão global da situação dos equipamentos e sistemas constituintes da rede. A gerência de falhas fornece esta visão, porque permite aos elementos gerenciados enviar as falhas identificadas para um ponto de observação centralizado. Estas informações são concentradas e correlacionadas dando aos usuários uma visão mais completa da situação operacional da sua rede.

O próximo passo é permitir aos usuários do sistema de gerência configurar a rede de modo a satisfazer as necessidades de gerência (desativação de canais E1, mudança na tabela de roteamento, etc) ou necessidades administrativas (desconectar um assinante, redimensionar os operadores em uma plataforma de serviços, etc). Neste caso as necessidades básicas são a redução de custos e a diminuição do tempo para que estas alterações sejam feitas.

O capítulo 2 descreve os conceitos de gerência integrada de redes e serviços e TMN que serão utilizados como base no desenvolvimento deste trabalho.

O capítulo 3 descreve os serviços de telecomunicações e os recursos físicos e lógicos, que são os elementos a serem gerenciados pelo modelo de informação proposto.

No capítulo 4 é descrita a metodologia a ser adotada neste trabalho, que seguirá principalmente o recomendado nas recomendações da série [M.3000] do ITU-T. O capítulo 5 descreve a aplicação da metodologia.

As conclusões e indicações para futura evolução do trabalho são apresentadas no capítulo 6.

2. GERÊNCIA DE REDES DE TELECOMUNICAÇÕES E TMN

2.1. A Arquitetura TMN

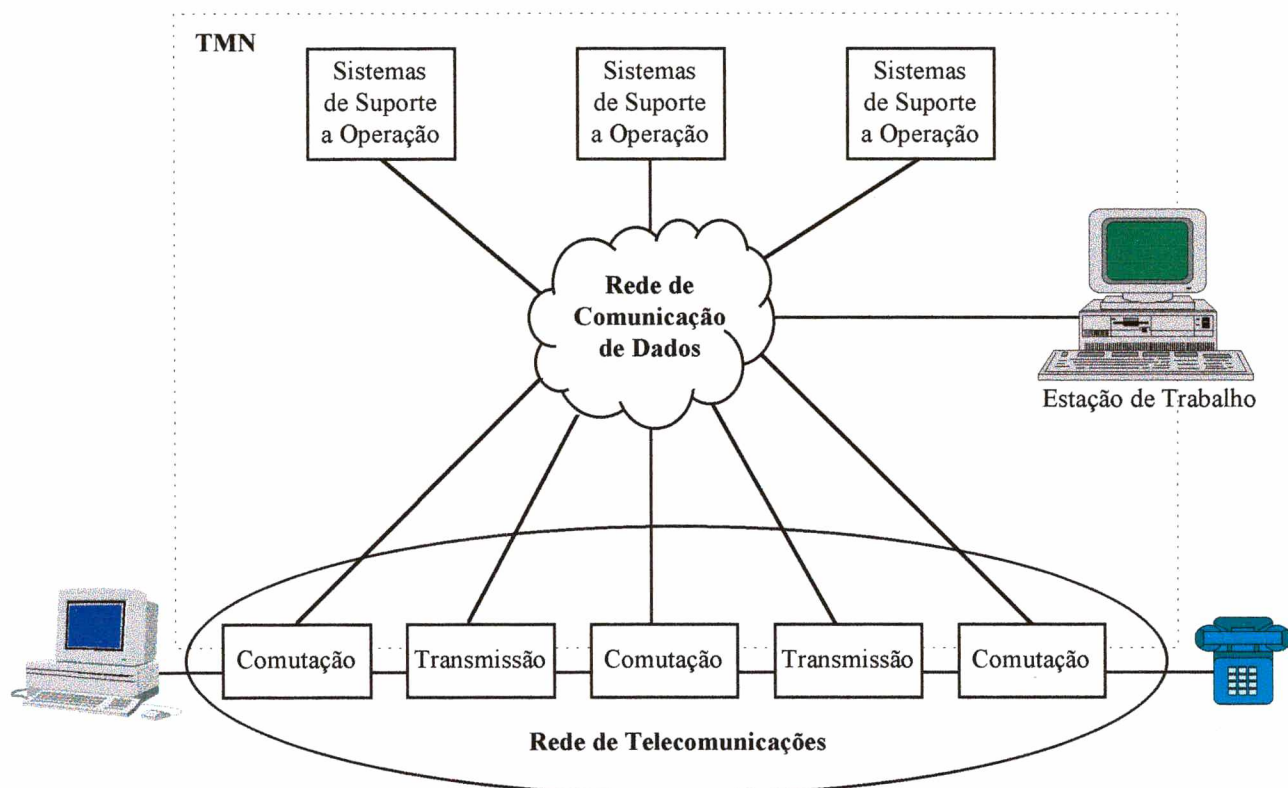
A arquitetura TMN serve como um modelo genérico para a implementação de sistemas abertos voltados à operação, administração, manutenção e provisionamento (OAM&P) das redes e serviços de telecomunicações.

Pode ser conceituada como "uma arquitetura organizada para interconectar vários tipos de Sistemas de Suporte à Operação - OS (*Operations System*) e equipamentos de telecomunicações para a troca de informações de gerência utilizando interfaces padronizadas que incluem a definição de protocolos e mensagens" [BRISA]. Oferece tanto para a administração da rede de telecomunicações quanto para os fabricantes um conjunto de padrões a serem utilizados no desenvolvimento, compra de equipamentos de telecomunicações, assim como no projeto da rede de gerência.

O ITU-T, por meio da recomendação [M.3010], define a arquitetura da rede de gerência, conceitualmente separada da rede de telecomunicações, mas com ela intercomunicando-se em pontos específicos para coleta de informação e controle. Partes da rede de telecomunicações podem ser utilizadas para realização de funções de gerência.

A figura 1 mostra o relacionamento entre a TMN e a rede de telecomunicações

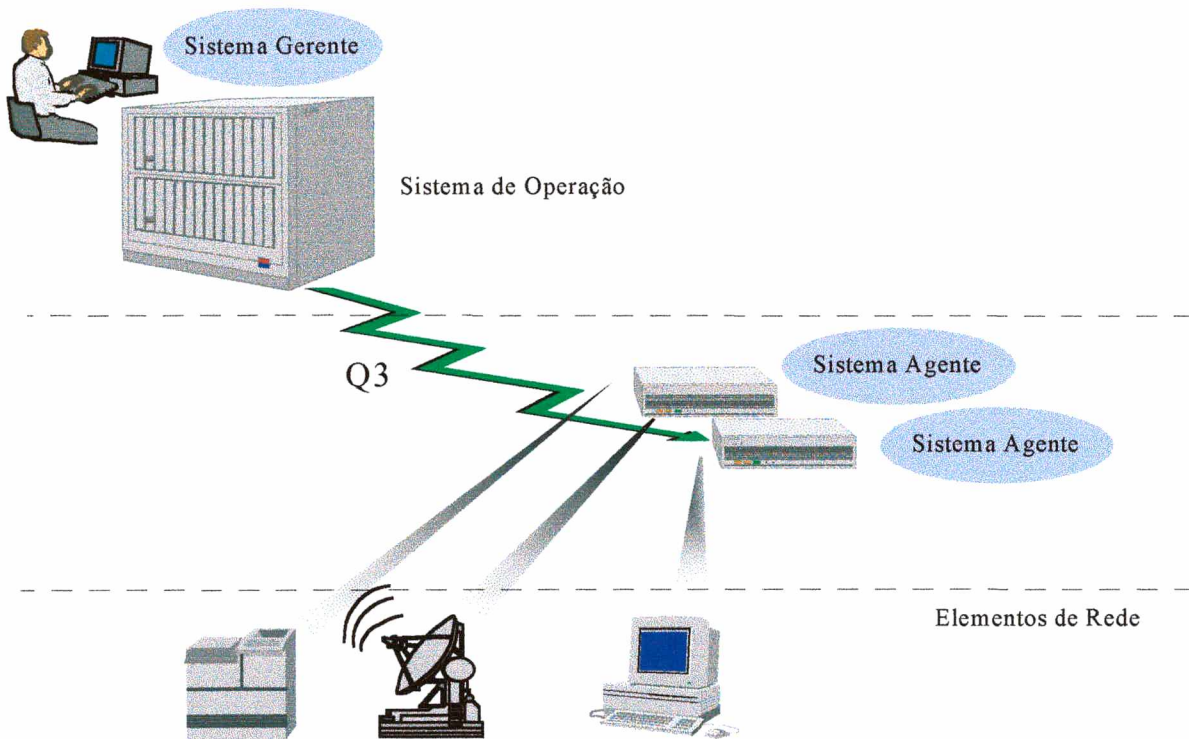
Figura A: Relacionamento entre a TMN e a rede de telecomunicações



A rede de telecomunicações é composta de uma diversidade de equipamentos distribuídos geograficamente, tais como: centrais de comutação, computadores, rádios, fibras ópticas, roteadores, multiplexadores, etc. A rede de telecomunicações é um ambiente distribuído; logo, sua gerência é intrinsecamente uma aplicação distribuída, envolvendo a troca de informações entre processos de gerência.

A figura 2 [Schweitzer] ilustra um ambiente de gerência TMN.

Figura B: Exemplo de um ambiente de gerência TMN



Uma TMN pode variar da simplicidade de uma interconexão entre um OS e um elemento da rede de telecomunicações à uma complexa rede interconectando vários OSs a muitos tipos de equipamentos de telecomunicações.

A arquitetura funcional da TMN é baseada em blocos funcionais que provêm as funções gerais que permitem a execução das funções de gerência. As funções que estão diretamente envolvidas na gerência pertencem a TMN e são objetos de padronização. Na arquitetura funcional são definidos os seguintes blocos funcionais:

- OSF** (*Operations Systems Function*). O OSF tem a finalidade de processar as informações relacionadas a gerência de telecomunicações para controlar, monitorar e coordenar as funções de telecomunicações inclusive a própria TMN.
- WSF** (*Workstation Function*). O WSF provê os meios para que o usuário possa interpretar as informações da TMN e vice-versa.
- NEF** (*Network Element Function*). O NEF é um bloco funcional que se comunica com a TMN para ser controlado e gerenciado. O NEF provê as funções de telecomunicações para a rede de telecomunicações sendo gerenciada.
- QAF** (*Q Adaptor Function*). O QAF é utilizada para conectar a TMN entidades que são funcionalmente equivalentes a NEF e OSF mas possuem interfaces proprietárias.
- MF** (*Mediation Function*). O MF atua sobre as informações que são trocadas entre o OSF e NEF (QAF) para assegurar que as informações estejam de acordo com o esperado pelos blocos funcionais.

A interconexão entre blocos funcionais é realizada através dos pontos de referência. Os pontos de referência definem a divisão dos serviços entre os blocos funcionais. Três classes de pontos de referência são definidos:

- q** entre OSF e OSF, OSF e NEF(QAF), OSF e MF, MF e MF, MF e NEF(QAF).
- f** entre WSF e OSF e WSF e MF.
- x** entre OSF e OSF (de TMN diferentes).

A TMN se utiliza dos conceitos de gerência desenvolvidos pela ISO para a gerência OSI (*Open Systems Interconnection*). Na gerência OSI a função gerente é definida como sendo a parte da aplicação distribuída que emite operações de gerência e recebe notificações, e a função agente como a parte da aplicação que recebe as operações de gerência, atua sobre os recursos gerenciados e responde a estas operações e envia notificações.

Para que exista a interoperação dos sistemas, o conhecimento compartilhado de gerência (SMK - *Shared Management Knowledge*) é um conceito fundamental, fornecendo um contexto comum aos sistemas agentes e gerentes sobre seguintes aspectos:

- Capacidades de protocolo suportadas;
- Funções de gerência suportadas;
- Classes de objetos gerenciados suportadas;
- Instâncias de objetos gerenciáveis disponíveis;
- Capacidades autorizadas;
- Relações hierárquicas (*containment*) entre os objetos (*name bindings*).

Uma TMN pode ser implementada em uma variedade de configurações físicas. Os blocos construtivos da TMN são nomeados de acordo com o conjunto de blocos funcionais que cada um pode conter. Para cada bloco construtivo temos um bloco funcional que lhe é característico e o nomeia. Existem os seguintes blocos construtivos definidos:

OS (*Operations System*). O OS é o sistema que executa o OSF.

MD (*Mediation Device*). O MD é o equipamento que realiza a MF.

QA (*Q Adaptor*). O QA é o equipamento que conecta OS e MF a equipamentos da rede de telecomunicações que não tem interface TMN.

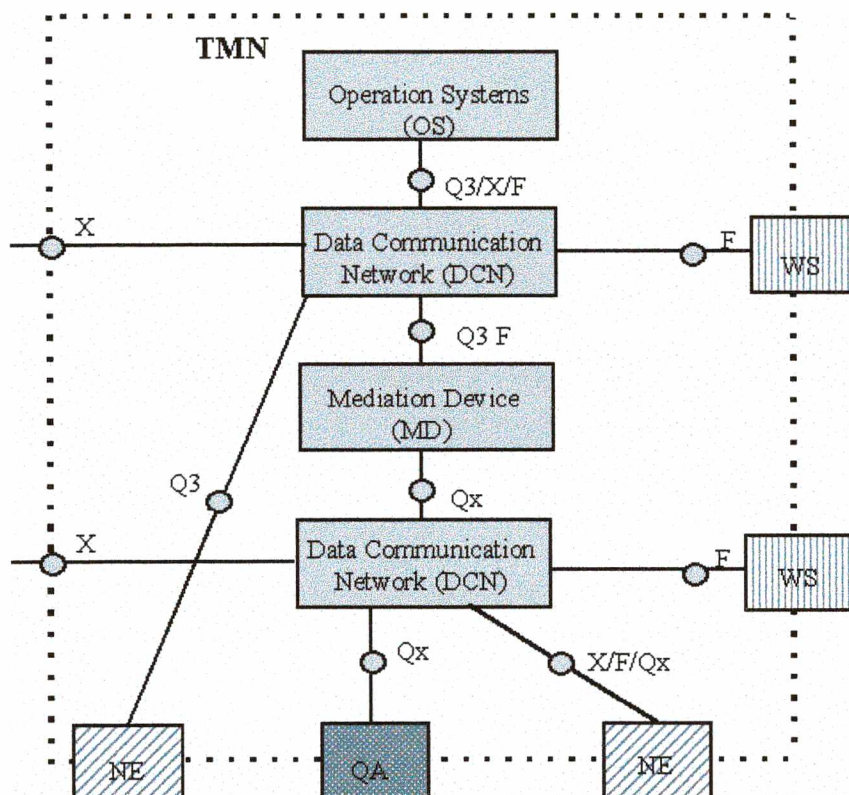
- NE** (*Network Element*). O NE implementa a NEF, sendo composto de equipamentos de telecomunicações e as função de interface com a TMN.
- WS** (*Workstation*). WS é um sistema que implementa a WSF. A WS permite que o usuário se comunique com a TMN.

As interligações entre os blocos construtivos são realizadas através de interfaces padronizadas que são aplicadas nos pontos de referência existentes entre os blocos funcionais de cada bloco construtivo. As seguintes interfaces são definidas para uma TMN:

- Q** Interface que é aplicada ao ponto de referência q. A TMN criou duas interfaces Q: Q_x e Q_3 . A interface Q_3 é caracterizada pela porção do modelo de informação compartilhado entre o OS e os outros elementos da TMN e a interface Q_x é caracterizada pela porção do modelo de informação compartilhado entre o MD e o NE(QA). A interface Q_x é prevista como uma interface simplificada para os elementos de rede que não possam suportar uma interface Q_3 . Uma interface Q_3 consiste de um modelo de informações de gerência representado conforme os padrões aplicáveis [M.3010, M.3100, X.701, X.720, X.721, X.722], protocolos das camadas 1 a 6 do modelo de referência OSI [X.200, Q.810, Q.811, TELE059, TELE060] e o serviço de gerência da camada de aplicação CMIS (*Common Management Information Service*) [X.710, X.711].
- F** Interface que é aplicada ao ponto de referência f. Não é padronizada pela TMN.
- X** Interface que é aplicada ao ponto de referência x. Esta interface é semelhante a interface Q porém com os aspectos de segurança mais reforçados.

A figura 3 a seguir ilustra a estrutura de uma TMN.

Figura C: Pontos de interface da arquitetura física da TMN



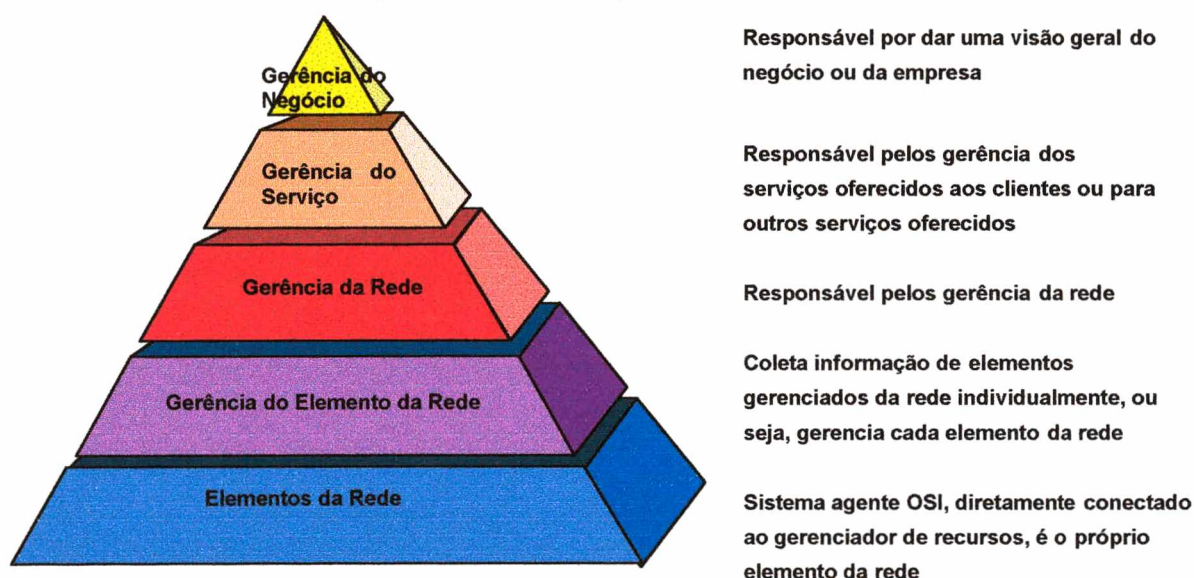
A complexidade das redes de telecomunicações cria a necessidade de subdividir a gerência em domínios. Cada domínio restringe o escopo das atividades de gerência e encapsula as operações para outros domínios. Esta divisão é realizada através da Arquitetura Lógica em Camadas (LLA - *Logical Layered Architecture*).

A arquitetura lógica em camadas é um conceito de desenvolvimento segundo o qual, baseadas em uma hierarquia, as funções de gerência podem ser agrupadas. O escopo de gerência de cada camada é mais amplo do que o da camada inferior.

O conceito da LLA é usado recursivamente como ferramenta para decomposição de uma atividade particular de gerência em uma série de domínios funcionais aninhados. Cada domínio pode conter outros, ou funções para a representação dos objetos gerenciados nele contidos.

A figura 4 exemplifica a divisão funcional típica em cinco domínios (camadas) de gerência.

Figura D: Camadas de gerenciamento TMN



2.2. Modelo de Informação TMN

A TMN usa o Modelo de Gerência OSI para descrever os elementos de rede, utilizando uma modelagem orientada a objetos [Booch91, Yourdon91] para representar abstrações de recursos lógicos e físicos a serem gerenciados.

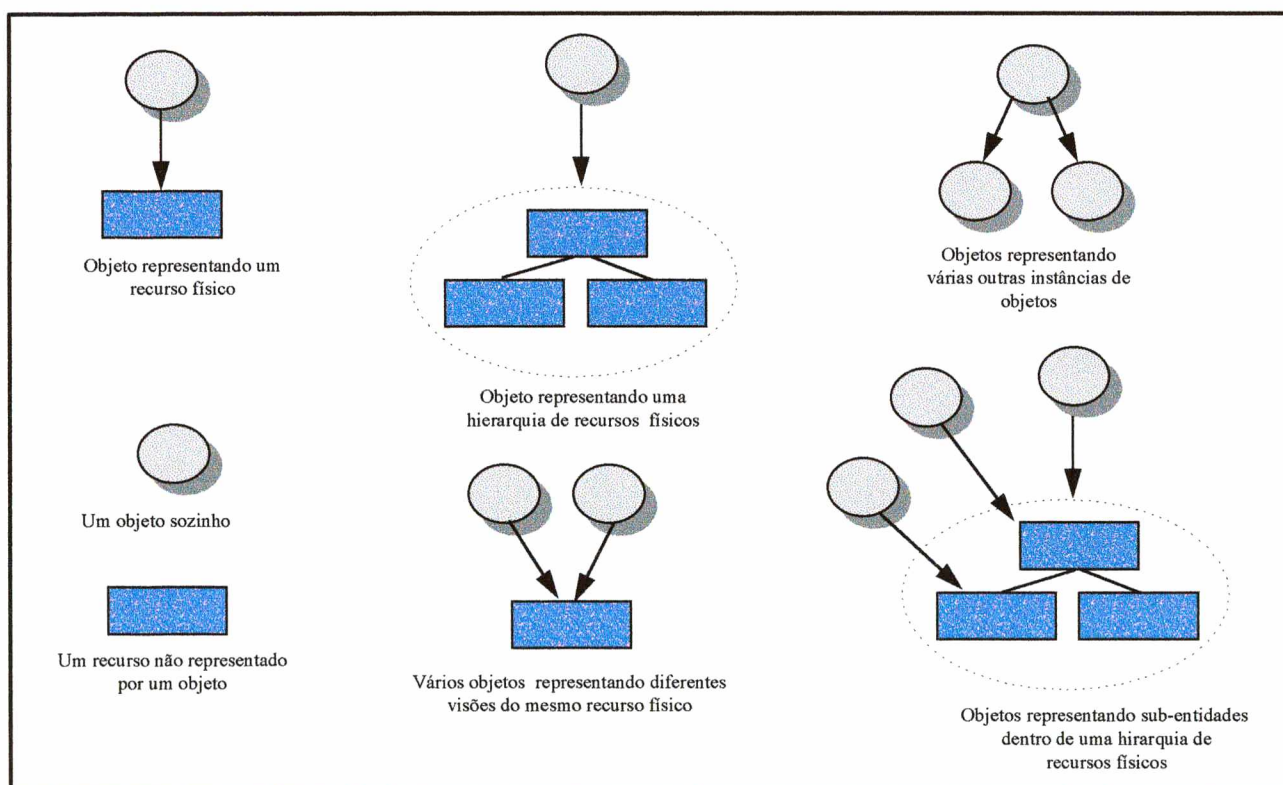
Na definição de objetos gerenciados deve-se considerar que :

- ⇒ não existe necessariamente uma correspondência um-para-um entre os objetos gerenciados e os recursos reais;

- ⇒ um recurso pode ser representado por um ou mais objetos gerenciados. Quando um recurso é representado por múltiplos objetos gerenciados, cada um deles representa uma visão distinta daquele recurso;
- ⇒ podem existir objetos gerenciados representando recursos lógicos da TMN;
- ⇒ se um recurso não é representado por um objeto gerenciado, ele é invisível aos sistemas de gerência;
- ⇒ um objeto gerenciado pode prover um visão abstrata de recursos que são representados por outros objetos gerenciados;
- ⇒ um objeto gerenciado pode estar contido em outro objeto.

A figura 5 é uma ilustração do relacionamento entre objetos gerenciados e recursos físicos.

Figura E: Relacionamento entre objetos gerenciados e recursos físicos



Os objetos gerenciados se diferenciam através de suas propriedades associadas. Estas definem os atributos, ações, operações, notificações e comportamentos que os mesmos podem assumir. As propriedades são especificadas através de uma estrutura de classes, agrupando objetos com características comuns. Um objeto gerenciado é uma instância de uma classe, e compartilha as mesmas propriedades com os outros membros desta mesma classe.

A documentação de classes de objetos gerenciados é feita segundo as diretrizes para definição de objetos gerenciados (GDMO - *Guidelines for the Definition of Managed Objects*) [X.722] através de estruturas notacionais pré-definidas chamadas *templates*, que utilizam a sintaxe conhecida como ASN.1 (*Abstract Syntax Notation One*) [X.208] para descrever os objetos e suas características.

Na especificação de um modelo de informação, devem também ser documentados os relacionamentos existentes entre objetos. Estes podem ser realizados através de Diagramas Entidade-Relacionamento (DERs) e explicitados através das hierarquias de *containment* e herança.

A relação que identifica as classes possíveis para objetos superiores usados na construção do nome de um objeto gerenciado é conhecida como *name-binding*. A hierarquia de *containment* e os *name-bindings* determinam como será referido um objeto individual dentro da árvore de informações de gerência (MIT - *Management Information Tree*) [X.720], explicitando modelos das relações do tipo todo-parte (objeto A contém objeto B).

Quando gerenciado, cada equipamento é chamado de Elemento de Rede - NE (*Network Element*).

3. SERVIÇOS DE ATENDIMENTO AO USUÁRIO

3.1. Introdução

As empresas de telecomunicações prestam uma série de serviços aos seus usuários utilizando o próprio sistema de telecomunicações como forma de acesso a estes serviços. Como exemplo destes serviços, podemos citar: auxílio a lista (102), solicitação de consertos (103), solicitação de serviços (104), teledespertador (139) , hora certa (130), interurbano via telefonista (101), etc.

Os serviços são realizados através de uma ou mais chamadas telefônicas entre o usuário e o sistema, podendo a chamada ser iniciada pelo próprio usuário (auxílio a lista), pelo sistema (teledespertador) ou ambas. Utiliza-se como interface entre o usuário e o serviço: operadores (telefonistas), o teclado do telefone ou mais recentemente a própria voz com o uso de sistemas de reconhecimento automático de voz.

Atualmente a maior parte destes serviços ainda utiliza operadores devido à deficiência das interfaces disponíveis ao usuário. O teclado telefônico é uma interface muito pobre porque disponibiliza apenas informações numéricas e o Brasil ainda possui uma grande quantidade de telefones decádicos (por pulso), que devido ao tipo de sinalização adotada (sinalização DC) [NEC], dificulta a detecção remota dos números digitados, gerando um alto percentual de informações incorretas. Da mesma forma, o reconhecimento de voz ainda está “engatinhando” em termos de desenvolvimento, estando sujeito ainda a uma grande taxa de erros de interpretação, o que inviabiliza o seu uso em serviços mais sofisticados.

Um dos serviços mais utilizados pelos usuários das redes de telecomunicações no Brasil e no mundo é o serviço de auxílio à lista, que permite a um usuário localizar o número do telefone de um outro assinante.

O sistema de telecomunicações, devido as características técnicas dos equipamentos utilizados, associa a cada usuário um número que identifica seu aparelho dentre todos os existentes na planta. Nos primórdios na telefonia, quando se utilizava a comutação manual, o operador (telefonista) podia utilizar um nome ou outra informação (ex.: tipo do negócio) para identificar qual o telefone a chamar. Na comutação automática essa capacidade de comutação inteligente não é mais disponível, o que exigiu que os operadores disponibilizassem listas contendo o nome do usuário e seu número telefônico. Normalmente as empresas operadoras fornecem as listas impressas aos seus usuários, disponibilizam o serviço de auxílio a lista e até a Internet hoje pode acessar esta lista.

O sistema de telefonia fixa tem dificuldades técnicas para manter o número telefônico em uma mudança de endereço. Uma central telefônica abrange uma área física determinada, e com a constante mobilidade hoje existente em grandes cidades, existe um grande número de mudanças que não se refletiram ainda em listas em papel. Isto exige que a empresa disponibilize esta informação de uma forma fácil para seus usuários, permitindo a comunicação entre esses assinantes.

Outra situação comum é a necessidade de se localizar o telefone de empresas que prestam algum tipo de serviço, ou por seu endereço. A lista telefônica funciona também como uma das formas para se localizar fornecedores, prestadores de serviços tais como: encanadores, eletricitas, mecânicos, etc.

O serviço de auxílio a lista é hoje um serviço essencial para as empresas operadoras, principalmente as que operam as redes de grandes cidades.

O acesso do usuário ao serviço é feito ligando-se para um número específico, no caso do sistema Telebrás (Telecomunicações Brasileiras S.A) padronizado como 102, é atendido por um operador, fornecendo as informações que dispõe para localizar o telefone. O operador então consulta a base de dados de assinantes e informa ao usuário o telefone localizado (se existir).

O serviço de auxílio a lista utiliza operadores porque o tipo de informações que o usuário deve prover para que o sistema possa localizar a informação desejada é muito complexo (pesquisa em base de dados de forma genérica), e a capacidade do operador em processar informações também é utilizada como interface entre o usuário e o sistema de pesquisa.

Outra característica importante do serviço de auxílio a lista é a sua tarifação. O sistema Telebrás padronizou que serviço será gratuito ou cobrado, dependendo da informação fornecida. Se a informação desejada constar da lista impressa, a ligação é cobrada, caso contrário esta é gratuita. Portanto a tarifação deve ser feita pelo sistema que implementa o serviço, gerando um bilhete de tarifação que identifique o assinante chamador e informe se a chamada deve ser cobrada ou não.

Um outro serviço também muito comum no Brasil é o teledespertador, em que o telefone é utilizado como despertador do usuário. Este usuário recebe uma chamada telefônica em um horário predeterminado, com uma mensagem gravada indicando a chegada do horário programado. Este serviço é programado através de uma chamada telefônica para um número (no sistema Telebrás 134) e informando-se qual o horário ou horários em que se deseja a chamada.

Este serviço hoje pode e é normalmente prestado de forma totalmente automática. Isto é facilitado pelo tipo de dado que o usuário deve informar (horário), eminentemente numérico, pelo tipo das operações que o usuário pode executar (programar um horário, desprogramar) ser simples e pela execução do serviço consistir apenas em realizar uma chamada em um horário determinado e colocar uma mensagem gravada. Este serviço é tarifado pelas empresas operadoras e em alguns casos até prestado por terceiros.

Os sistemas utilizados para prover estes serviços são constituídos de uma interface com a rede de telecomunicações, uma interface com os operadores (no caso de um serviço que os utilize) e um sistema computacional, que realiza os procedimentos necessários para a execução do serviço.

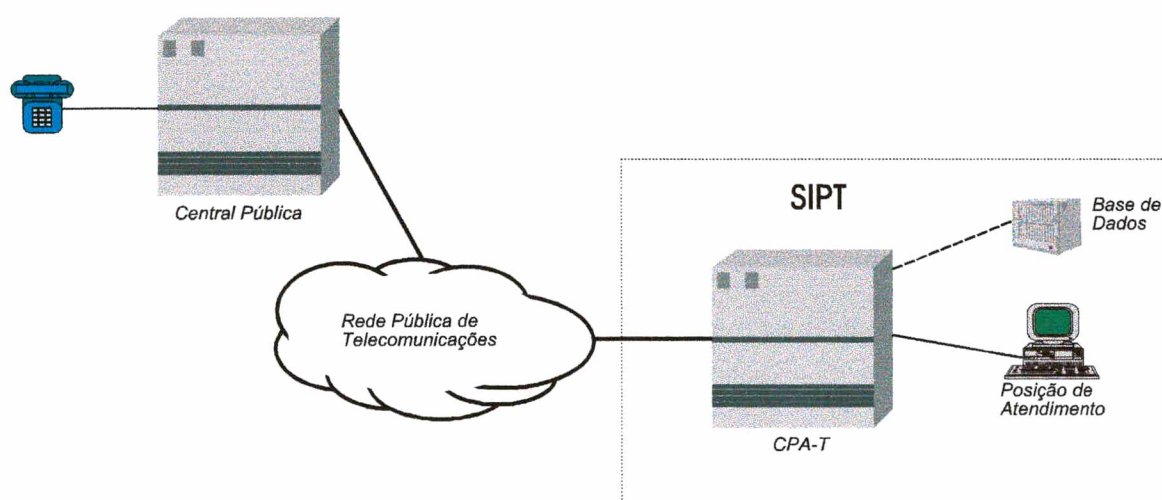
Com a proximidade cada vez maior entre a informática e as telecomunicações, a quantidade de novos serviços que se tornam possíveis técnica e economicamente é muito grande. Por exemplo: serviço teleconta, onde o usuário pode saber qual o valor da sua conta telefônica. No caso de um sistema desregulamentado, estes serviços podem se tornar uma vantagem competitiva.

3.2. SIPT

A Telebrás, desejando possuir um padrão para os sistemas que sejam utilizados para suportar os serviços de atendimento ao usuário, definiu o SIPT (Sistema Integrado de Posições de Tráfego) [TELE704], como um sistema que tem por objetivo o atendimento centralizado/integrado dos serviços que utilizam operadores, por meio de interfaces padronizadas. Este sistema é projetado de modo a permitir que o mesmo equipamento possa prestar uma gama de serviços, utilizando os mesmos recursos físicos e lógicos, tais como: operadores, equipamentos de comutação, canais de interconexão, equipamentos de informática, base de dados, etc.

A figura 6 mostra a estrutura de um SIPT

Figura F: Estrutura de um sistema SIPT



Este sistema é composto de uma central de comutação CPA (Central de Programa Armazenado) que se interliga a rede de telecomunicações, um grupo de posições de atendimento, e um banco de dados que serve como repositório das informações prestadas ou colhidas pelos serviços.

A central de comutação é responsável pela execução dos procedimentos necessários para que as chamadas telefônicas sejam realizadas. Uma central CPA se caracteriza por suas funções estarem sob o controle de um programa armazenado no equipamento, o que permite que sua funcionalidade seja alterada sem que a sua estrutura física seja modificada.

A posição de atendimento é a interface do operador com o sistema, compondo-se de um teclado associado a um terminal de vídeo e uma interface de áudio, ou apenas da interface de áudio, caso o serviço a ser prestado não exija uma interface mais complexa. Nesse equipamento, o operador interage com o sistema telefônico e com o sistema de informática para executar o serviço desejado.

O banco de dados é o repositório lógico das informações a serem utilizadas pelo sistema. Como exemplos, temos a base de dados de assinantes para o serviço de auxílio a lista, a lista de chamados para conserto, a tabela de programação de chamadas para o teledespertador, etc...

O SIPT pode ser distribuído geograficamente, possibilitando que se tenha um melhor aproveitamento dos recursos. O uso de posições de atendimento remotas e de base de dados localizadas em outros sistemas distantes geograficamente são situações comuns, que exigem que o sistema aceite que estes componentes sejam remotos.

A Telebrás definiu o SIPT como sendo a plataforma padrão para implementar os serviços de atendimento ao usuário na rede telefônica. Podem existir sistemas com apenas um serviço em utilização, caso de um serviço que demande muitos recursos, e.g. o auxílio a lista (102) de uma grande cidade ou estado; ou vários serviços compartilhando a mesmo sistema, normalmente visando redução de custos.

Os serviços podem ser visualizados como uma função do programa que está implementado em um SIPT. Devido as características de funcionamento, configuração e operação dos serviços pode-se definir um serviço genérico do qual se possa derivar os serviços individuais.

Como qualquer sistema existente em uma rede de telecomunicações, a observação do funcionamento e mensuração dos parâmetros de operação são fundamentais para que o serviço prestado obedeça os critérios de desempenho exigidos. A determinação imediata de ações corretivas em caso de falhas e reconfigurações em caso de problemas de qualidade de serviço; congestionamento, falta de operadores, etc. garante que estes serviços estarão disponíveis ao usuário para sua utilização.

A Telebrás [TELE704] define uma série de requisitos de operação e gerência para esses sistemas, principalmente para o uso da área de operação e planejamento das empresas de telecomunicações. Esses requisitos permitem definir um modelo básico de gerência que atenda as necessidades dessas áreas.

O sistema que implementa o SIPT deve atender um conjunto de requisitos funcionais referentes as suas interfaces e funções.

Relacionado a interface com os operadores os requisitos principais são:

- ⇒ Atendimento de critérios de ergonomia, visando o conforto, principalmente visual e auditivo.
- ⇒ Homogeneidade entre os sistemas dos operadores e supervisores.
- ⇒ Possibilidade de criação de hierarquias entre as posições de operador, pelo uso de comandos homem-máquina.
- ⇒ Capacidade do operador e/ou supervisor colocar sua posição em estado inativo.
- ⇒ Uso de ícones orientativos para o operador indicando estados, configurações, etc.

- ⇒ Poder expedir automaticamente chamadas para o assinante chamador ou chamado, sob comando da operadora.

Requisitos funcionais relativo a operação:

- ⇒ SIPT deverá oferecer a possibilidade do operador interagir com formulários.
- ⇒ Um tipo particular de formulário é o bilhete que terá seus campos preenchidos pelo operador ou pelo sistema, permitindo o encaminhamento e tarifação da chamada/serviço.
- ⇒ Os bilhetes deverão ser armazenados para posterior recuperação. Esta recuperação poderá ser feita por comando do operador, ou após decorrido algum tempo.
- ⇒ SIPT deverá colocar no bilhete a data e hora de início e a duração da chamada correspondente.
- ⇒ Os bilhetes deverão possuir as informações adequadas a atender a cada um dos tipos de chamadas a serem processadas.
- ⇒ SIPT deverá ter a capacidade de apresentar mensagens, relatórios estatísticos, alarmes, etc.
- ⇒ SIPT deverá ter a facilidade de periodicamente compilar as informações de tarifação, contidas nos bilhetes armazenando estas informações de forma compatível com o processo de faturamento.

Requisitos funcionais relativos ao encaminhamento da chamada:

- ⇒ SIPT deverá ter a capacidade de constituir grupos de posições de operador para o atendimento de determinados tipos de chamadas ou tipos de serviços. A constituição e reconfiguração desses grupos poderá ser feita dinâmica sem interromper a operação do sistema. Uma supervisora habilitada por senha terá comandos homem-máquina para executar essas operações.

- ⇒ encaminhamento das chamadas ao SIPT deverá ser feito ao grupo de posições apto a fazer o seu atendimento. A seleção da posição dentro do grupo deve seguir um critério que balanceie o tráfego dentro do grupo.
- ⇒ No caso de não atendimento de uma chamada por uma posição dentro de um tempo predeterminado por um comando, entre 0 e 30 segundos, a chamada será sinalizada na posição da supervisora e oferecida para outra posição dentro do grupo selecionado. Não sendo atendida novamente, será oferecida a supervisora.
- ⇒ Para chamadas em fila, deverá existir a possibilidade de enviar música ou mensagens orientadoras automaticamente ou por comando.
- ⇒ As chamadas em fila devem seguir o critério de primeiro na fila primeiro a ser atendido.

Requisitos básicos referente ao processamento das chamadas:

- ⇒ operador deverá ser informado através de um sinal audível e pela tela que existe uma chamada sendo oferecida para atendimento. O operador poderá atender esta chamada através de uma tecla de atendimento. Opcionalmente o atendimento será automático após o envio do sinal audível e o recebimento dos dados referentes a chamada na tela.
- ⇒ operador poderá transferir a chamada atendida para o supervisor, ou para outra posição livre de operador, pela digitação do número do operador antes do código de acesso. Também deverá ser possível o acesso de um telefone com categoria de teste a uma determinada posição de operador.
- ⇒ Uma posição de operador após atender a chamada poderá efetuar:
 - ⇒ estacionamento da chamada, permitindo o atendimento de uma nova chamada, retornando a chamada estacionada quando a nova terminar.
 - ⇒ reter a chamada, permitindo o operador intervir na chamada a qualquer instante.

- ⇒ separar a chamada, permitindo o operador conversar com um dos interlocutores em particular.
- ⇒ monitorar a chamada escutando a conversação para avaliar a qualidade de transmissão por um curto espaço de tempo.

Requisitos referentes ao acesso as bases de dados:

- ⇒ SIPT deverá permitir a interação com base de dados externas (regionais, nacionais e internacionais), via protocolos X.25 e SS#7.
- ⇒ A interação com a base de dados deverá ser feita através da própria posição de operador.
- ⇒ SIPT também deverá permitir a interação com uma base de dados interna.
- ⇒ As interações com as bases serão consultas, ou registro de informações.

Requisitos referentes a operação e manutenção:

- ⇒ As facilidade de OAM das posições de operador deverão estar integradas com a CPA-T responsável pela sistema.
- ⇒ As falhas de operação do SIPT quando detectadas, devem ter seu grau de severidade informado através de um sistema de alarme. Os alarmes devem conter informações adicionais que permitam o diagnóstico destas falhas.

3.3. A Plataforma de Serviços Dígitro

A Dígitro, como empresa fornecedora do sistema Telebrás, desenvolveu uma plataforma de serviços que atendem aos requisitos do SIPT. Esta plataforma é baseada na central de comutação AXS/20 como CPA-T.

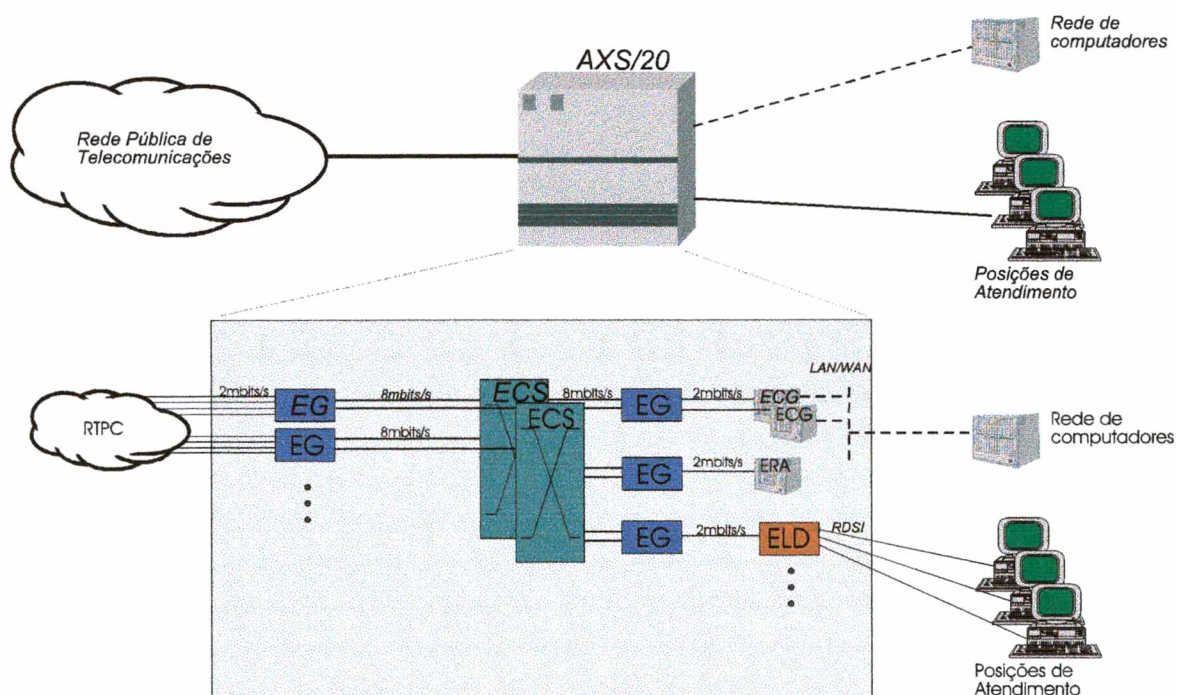
A central AXS/20 é uma central de comutação digital de grande porte e de última geração. Suas principais características são:

- Plataforma computacional aberta, baseada em sistema operacional UNIX;
- Processamento distribuído, baseada na arquitetura cliente-servidor;
- Grande conectividade, suportando uma grande variedade de interfaces físicas (ethernet, fast ethernet, serial, etc.) e protocolos (TCP/IP, IPX, X.25);
- Suporta os seguintes protocolos de sinalização: MFC-5C, MFC-5S, DTMF, decádico, SS#7 ;
- Sistema tolerante a falhas.

3.3.1. Estrutura física da plataforma de serviços

A central AXS/20 utiliza uma arquitetura física modular, com cada módulo possuindo uma finalidade específica. A figura abaixo ilustra os principais componentes da central.

Figura G: Estrutura da central AXS/20 em um sistema SIPT



ECG - Estágio de Coordenação e Gerência.

Unidade principal de processamento da central. Pode ser visto como o cérebro da central, é baseado em microcomputadores padrão IBM-PC utilizando o sistema operacional Solaris (Sun Microsystems) e é responsável pelas operações de alto nível da central tais como: controle da comutação, execução dos serviços, etc.

ECS – Estágio de Comutação e Sincronismo

Unidade de comutação digital da central. Pode ser visto como o coração da central. Apresenta uma matriz de comutação com capacidade de 16000 conexões de 64kbits/s, suportando até 50000 assinantes.

EG – Estágio de Grupo

Interconecta a matriz de comutação (ECS) aos outros estágios da central. Possui capacidade de processamento local sendo responsável pela filtragem, identificação de sinais, troca de protocolos MFC, etc.

ELD – Estágio de Linha Digital

Responsável pela interligação entre as linhas digitais de assinantes e os outros módulos. Implementa as interfaces físicas da RDSI.

ERA – Estágio de Resposta Audível

Consiste de um microcomputador padrão IBM-PC com sistema operacional Solaris responsável pelo armazenamento e reprodução de áudio digitalizado, é utilizado tanto para mensagens de sistema, quanto para armazenar as caixas de voz dos usuário na função de correio de voz e outras finalidades.

Posição de atendimento digital

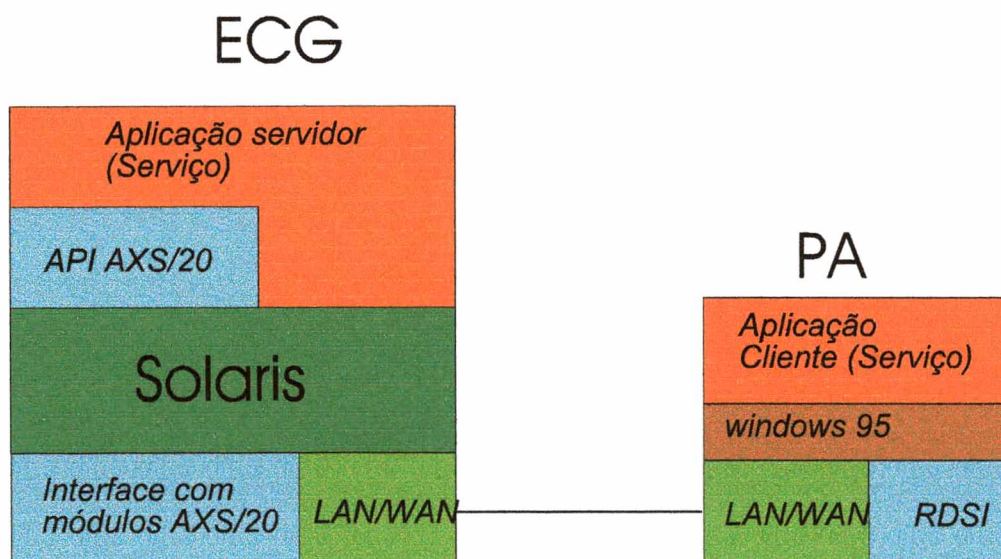
Consiste de um microcomputador padrão IBM-PC com sistema operacional windows 95 e interface RDSI. Executa as aplicações com as quais os operadores e supervisoras interagem. Utiliza o protocolo TCP/IP como protocolo de troca de informações entre as aplicações da (PA) e os servidores.

A central AXS/20 em conjunto com as posições de atendimento implementa um SIPT atendendo a todos os requisitos exigidos pela Telebrás para essa plataforma.

3.3.2. Um serviço na plataforma de serviços AXS/20

Uma aplicação na plataforma de serviços Dígitro é construída segundo uma estrutura cliente-servidor, onde a parte cliente é executada na PA e a parte servidor é executada no ECG. Esta estrutura é mostrada na figura abaixo.

Figura H: Estrutura de um serviço na plataforma AXS/20



A cada serviço a ser implementado na plataforma, é desenvolvido um conjunto de programas (aplicação) que realizam as operações de alto nível do serviço utilizando as funcionalidades básicas presentes no sistema. A plataforma AXS/20 já provê as funcionalidades de comutação, troca de sinalização (MFC, R2D, SS7, etc), tratamento de erros, interface com os sistemas de gerências, alocação de recursos, gravação e reprodução digital de áudio, etc.

A aplicação, que implementa o serviço no SIPT Dígitro, tem como responsabilidade configurar a troca de sinalização, encaminhar a chamada, comandar o envio ou gravação de mensagens digitalizadas, interagir com o operador, trocar informações com os sistemas externos, etc.

Para a operação do serviço temos os seguintes conceitos definidos:

Operador

Representa o atendente, supervisor, etc. A cada operador é associada uma identidade (nome, senha) e privilégios de acesso.

Grupo

O grupo é responsável pelas filas de atendimento destinadas a um serviço. Um operador pode estar associada a um ou mais grupos. Os grupos podem ser normais ou de transbordo, os de transbordo são aqueles que recebem a chamada caso o grupo normal não consiga atendê-la a tempo ou a fila está grande demais.

Serviço

Representa uma aplicação da plataforma SIPT, executando um serviço de atendimento específico, tal como: 134, 102, etc.

A um serviço é associado um grupo de operadores para atendimento das chamadas.

A Dígitro tem em operação várias plataformas AXS/20 operando como SIPT em empresas do sistema Telebrás, executando vários dos serviços mencionados, tal como: 134, 102 e outros.

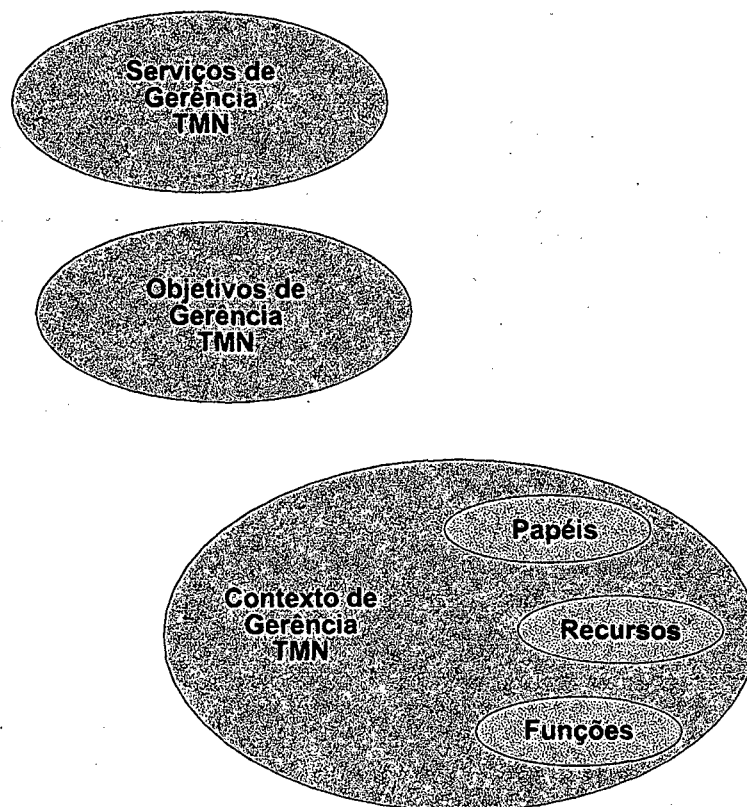
4. MODELO DE INFORMAÇÃO

4.1. Metodologia de Especificação de uma interface Q3

O ITU-T, através da recomendação [M.3020], definiu uma metodologia para a especificação de interfaces para uma TMN. Esta metodologia descreve os processos que transformarão os requisitos dos usuários da TMN em uma especificação de interface. Os requisitos de usuário são escritos na forma de serviços de gerência.

A figura abaixo ilustra os conceitos da metodologia de especificação TMN [M.3020].

Figura I: Relacionamento entre conceitos de especificação TMN



Um serviço de gerência TMN, de acordo com a definição do ITU-T [M.3020], referencia as informações pertinentes de gerência servindo a um objetivo de gerência específico.

Um contexto de gerência TMN define o ambiente em que os serviços de gerência TMN são executados, incluindo a descrição de quem gerencia a rede, o que é gerenciado e como é gerenciado. O contexto é descrito usando os componentes ortogonais: papel na gerência, recursos da rede de telecomunicações e as funções de gerência.

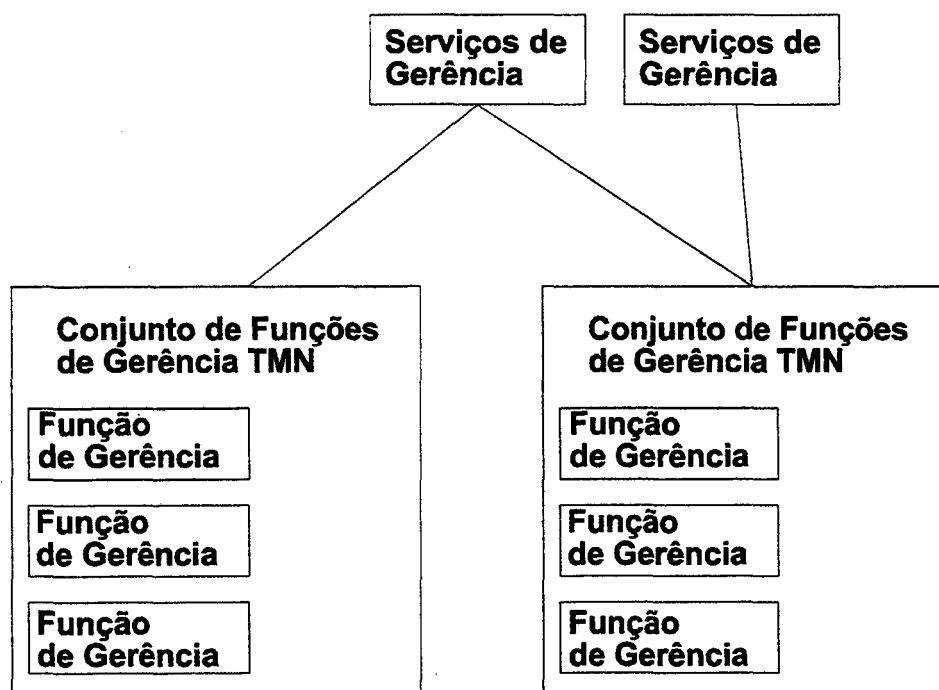
Uma função de gerência é uma interação cooperativa entre as aplicações de gerência (gerente e agente) para a gerência dos recursos da rede (físicos e lógicos). Isto normalmente corresponde a uma ou algumas operações/notificações CMIS. Tipicamente, uma função de gerência é a menor parte desta interação.

Um conjunto de funções de gerência é um agrupamento de funções de gerência que sejam correlacionadas. Um conjunto é a menor parte reutilizável de uma especificação funcional e deve ser considerado na sua totalidade. Ele é similar aos requisitos da OSI SMF (Systems Management Functions).

Os conjuntos de funções de gerência e suas funções individuais são objeto de padronização pelo ITU-T na recomendação [M.3400], e podem ser reutilizados por mais de um serviço de gerência. Somente se nenhum conjunto de funções já especificadas atender aos requisitos de um serviço de gerência é que um novo conjunto deve ser definido e incorporado ao padrão vigente. O ITU-T encoraja a reutilização dos conjuntos de funções de gerência com o intuito de reduzir o esforço de especificação.

A figura abaixo mostra a relação entre os serviços de gerência e as funções de gerência que os compõem.

Figura J: Relacionamento entre serviços e funções de gerência TMN



O ITU definiu que a metodologia [M.3020] se aplica a três diferentes tipos de usuários:

- ⇒ Usuário responsável pela coordenação, definindo os aspectos gerais da TMN.
- ⇒ Usuário responsável pela especificação de aplicações, definindo os aspectos específicos da TMN referentes a uma função ou equipamento particular.
- ⇒ Usuário responsável pela especificação de protocolos, definindo os aspectos específicos dos protocolos relacionados a TMN.

Este trabalho está relacionado ao segundo tipo de usuário devido ao interesse de se definir a gerência do serviço atendimento ao usuário de telecomunicações. A metodologia [M.3020] define um guia para ser utilizado pelo usuário, definindo os passos a serem seguidos, as informações necessárias a cada etapa e os resultados obtidos.

O primeiro passo da metodologia é definir qual o sujeito - recurso de telecomunicações a ser gerenciado. Esse recurso pode ser um equipamento, um serviço, um segmento da rede ou mesmo toda a rede de telecomunicações. Essa definição deve ser precisa e clara, porque dela dependem todos os outros passos da metodologia. No presente caso, o sujeito é o conjunto de serviços de atendimento ao usuário de telecomunicações que sejam prestados através da rede de telecomunicações. Desta forma, outros serviços prestados ao usuário, como conserto, pagamento de contas feitos de forma direta não estarão sendo considerados neste trabalho.

A execução da metodologia no recurso de telecomunicações escolhido resulta em um modelo de informação. Este modelo de informação é especificado em termos das classes dos objetos gerenciados com os respectivos atributos, ações e notificações. As classes representam todos os aspectos da gerência do recurso escolhido.

O modelo de informação define precisamente as mensagens que podem ser utilizadas para a gerência remota de um recurso de telecomunicações (sintaxe) e o significado (semântica) destas mensagens. A metodologia prevê que o CMIP é o protocolo mais apropriado para o transporte destas mensagens de gerência, porque o mapeamento das mensagens para os serviços CMIS é direto.

A metodologia está dividida em sete tarefas de especificação:

Tarefa 1: Descrever os serviços de gerência e seus objetivos através da perspectiva do usuário.

Tarefa 2: Descrever o contexto da gerência TMN.

Tarefa 3: Modelar os objetos.

Tarefa 4: Consolidar as informações disponíveis.

Tarefa 5: Definir a organização da informação de gerência.

Tarefa 6: Definir os requisitos da comunicação.

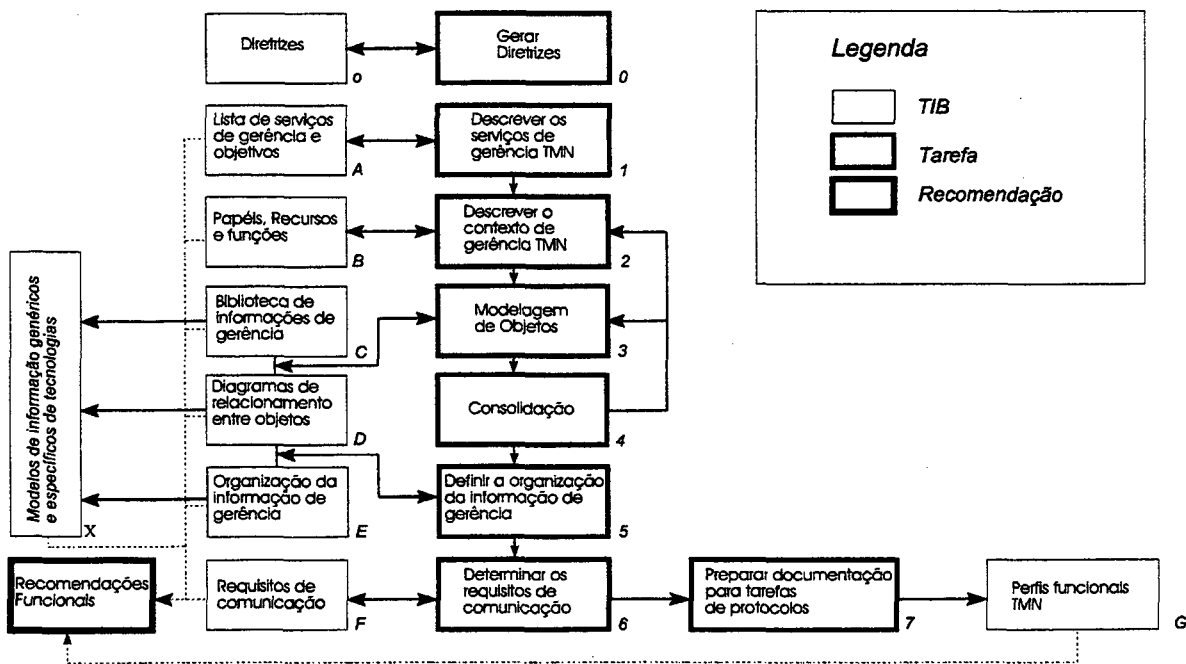
Tarefa 7: Preparar o documento para as tarefas de especificação de protocolos.

Esta numeração não implica uma seqüência definida das tarefas. As tarefas ou combinações de tarefas são realizadas de uma forma interativa em passos. O objetivo de cada passo é a especificação das funções de gerência, modelo de objetos e protocolos para uma comunicação específica entre blocos construtivos da TMN (NE, QA, MD, WS e OS). Para cada bloco, seu tipo e seu papel (gerente ou agente OSI) precisa ser especificado.

Cada tarefa possui uma base de informações associada denominada TIB (Task Information Base). Cada TIB contém os resultados das interações anteriores da metodologia. Em geral, uma tarefa tem como entradas a sua TIB e a TIB da tarefa precedente e os seus resultados se refletem na sua TIB.

A figura abaixo mostra um fluxograma das tarefas e as TIBs da metodologia aplicada ao modelo de informação, a parte do diagrama associada às tarefas de protocolo foi suprimida por não fazer parte do escopo deste trabalho.

Figura K: Metodologia de especificação de interfaces TMN



A seguir será apresentada uma descrição detalhada das tarefas individuais e suas TIBs.

4.1.1. Tarefa 0: Gerar diretrizes

Consiste na compilação de uma lista dos documentos relevantes para o entendimento da metodologia, particularmente na área de modelagem de objetos e sistemas de gerência.

Esta tarefa está fora da porção interativa da metodologia, e é realizada independentemente das tarefas remanescentes.

4.1.2. TIB 0: Diretrizes

Referências a outros documentos tais como:

- Modelagem orientada a objetos
- Ambiente de gerência OSI
- Diretrizes para a definição de objetos gerenciados (GDMO)
- Conjunto de instruções para os grupos executando as tarefas
- Princípios para a coordenação das tarefas

4.1.3. Tarefa 1: Descrever os serviços de gerência TMN e seus objetivos pela perspectiva do usuário

Identificar cada área de atividades de gerência que será suportada pela TMN em uma lista de serviços de gerência TMN. Para cada serviço, identificar os objetivos da gerência TMN e exemplos de benefícios que os usuários da TMN obtêm destes objetivos. A metodologia define diretrizes para a definição dos serviços denominadas GDMS (Guidelines for Definitions of Management Services).

4.1.4. TIB A: Objetivos e serviços de gerência TMN

Esta TIB é composta de uma lista completa dos serviços de gerência TMN com uma descrição curta de cada um dos serviços, complementada por uma lista dos objetivos.

Os serviços de gerência TMN já identificados pelo ITU-T estão listados na recomendação [M.3200].

Esta TIB tem seu conteúdo e estrutura definida pela GMDS, que especifica que os serviços de gerência devem ser definidos de acordo com um padrão (*template*). O padrão permite que os requerimentos da TMN sejam descritos de forma e completa e uniforme, facilitando a reutilização dos modelos de informação e os conjuntos de funções de gerência já existentes, ou definidos para este serviço.

A seguir temos a descrição do padrão determinado pela GDMS:

1) Descrição dos serviços de gerência

Descrição em prosa dos serviços, utilizando a recomendação [M.3200] como base.

2) Objetivos de gerência

Deve fornecer uma descrição clara dos benefícios do usuário e o motivo para uso deste serviço de gerência.

3) Descrição do contexto de gerência

Deve capturar de forma uniforme as informações relevantes para a gerência de uma específica área de telecomunicações. O contexto de gerência pode ser descrito utilizando-se três componentes ortogonais: papéis, recursos e funções. As informações referentes ao contexto de gerência fazem parte da TIB B.

3.1) Papéis de gerência

Descrição detalhada em prosa dos papéis identificados para este contexto de gerência, para permitir que as funções de suporte sejam determinadas.

3.2) Recursos de telecomunicações

Descrição dos recursos lógicos e físicos que devem ser modelados por uma ou mais objetos gerenciados. Devem ser claramente definidos e armazenados na TIB B.

3.3) Funções de gerência TMN

Descrição das funções de gerência a serem utilizadas para atingir os objetivos da gerência.

4) Cenários de gerência

Descrever exemplos de interação de gerência usando a definição da informação de gerência e os serviços e mensagens de gerência TMN.

5) Arquitetura

Os itens seguintes precisam ser alocados na arquitetura física e funcional TMN:

- Conjunto de funções de gerência ([M.3200])
- Funções de gerência ([M.3400])
- Pontos de referência aplicáveis (q, x, f)
- Interfaces aplicáveis (Q, F X)

4.1.5. Tarefa 2: Descrever o contexto da gerência TMN

Listar os papéis, recursos e funções de gerência TMN associados a um serviço de gerência TMN. As relações entre estes componentes devem ser descritas na forma de cenários quando possível, de acordo com o definido no GDMS no sub-item referente ao contexto de gerência.

Na descrição das funções de gerência e conjuntos de funções devem ser utilizadas as diretrizes para descrição de serviços de gerência GDMF (Guidelines for Definition of Management Functions) detalhadas na TIB B.

4.1.6. TIB B: Papéis da gerência TMN, recursos de telecomunicações e funções de gerência TMN

De acordo com o descrito na TIB A, aqui são listadas as informações referentes ao contexto de gerência juntamente com os cenários de uso contendo a relação entre estes componentes.

A GDMF apresenta de forma semelhante ao GDMS um padrão para a definição de funções de gerência, tendo por objetivo descrever de forma uniforme as funções e conjuntos de funções permitindo a sua reutilização.

O menor componente que permite reutilização é um conjunto de funções de gerência, que é um agrupamento de funções que contextualmente estejam relacionadas. Um conjunto precisa suportar os requisitos funcionais deste grupo de funções.

Abaixo temos o padrão a ser seguido de acordo com GDMF.

1) Descrição do conjunto de funções de gerencia TMN

Define o nome e descreve em prosa o conjunto.

2) Requisitos de gerência

Uma descrição dos objetivos da gerência e os conceitos empregados no conjunto sendo descrito.

3) Modelo funcional

Composto de:

- breve descrição do conjunto de funções de gerência TMN.
- funcionalidade associada ao conjunto.
- identificação dos recursos afetados pelo conjunto.

4) Funções de gerência TMN

Para cada função de gerência deve ser providenciado:

- Uma descrição sumária.
- Uma descrição detalhada da informação de gerência trocada entre o sistema gerenciado e o sistema gerente.

4.1.7. TIB X: Modelos de informação genéricos e específicos de uma tecnologia

A TIB X contém modelo de informação genérico da rede e modelos específicos a tecnologias. Estes modelos são definidos pela análise das arquiteturas da rede de telecomunicações (rede pública de comutação, SDH, rede de comutação de pacotes). O modelo e a hierarquia de classes devem conter as classes que são necessárias para a especificação e também as classes e objetos de suporte utilizados na definição de interfaces.

Os formulários de conformidade dos objetos gerenciados - MOCS (Managed Object Conformance Statement) devem ser especificados para as classes instanciáveis existentes na TIB X. Estes formulários devem seguir a recomendação [X.724].

A TIB X recebe informações das tarefas 3, 4 e 5 através das TIBs C, D e E.

4.1.8. Tarefa 3: Modelagem de Objetos

Utilizando o modelo de informação genérico de rede - GNIM [M.3100], identificar as classes existentes e novas classes que sejam necessárias para suportar as funções de gerência.

O processo é executado da seguinte forma:

- 1) Com o uso das TIBs C e D, são analisados os modelos genéricos e específicos, incluindo a hierarquia de classes, para determinar se alguma classe preexistente pode satisfazer aos novos requerimentos funcionais.
- 2) Se necessário, criar-se-á novas classes para satisfazer aos requerimentos não atendidos.

3) Modificar o modelo para incorporar as novas classes.

O modelo de objetos deve considerar extensões opcionais e específicas de fabricantes, atualizando o conteúdo das TIBs C, D e X.

4.1.9. TIB C: Biblioteca de informações de gerência.

A hierarquia de classes de objetos devem ser especificadas de acordo com a estrutura de informação de gerência - SMI, definida na recomendação [X.721] e as diretrizes de definição de objetos gerenciados - GDMO da recomendação [X.722]. Deve ser observado que a nomeação dos objetos não faz parte desta TIB, apenas a herança é definida o que determina a características das classes e objetos.

As classes de objetos criadas devem ser registradas de acordo com a recomendação [M.3020], com o uso de um identificador de objeto ASN.1. Porém devido às características deste trabalho de não estar associado a uma norma do ITU-T foi utilizada uma hierarquia própria da Flug Telemática e Automação para uso de seus modelos.

4.1.10. TIB D: Diagrama de entidade-relacionamento

O GNIM também é descrito através de um diagrama E-R (entidade-relacionamento) entre classes. Os objetos gerenciados aplicáveis e seus relacionamentos devem ser utilizados na tarefa 4 para especificar a organização da informação de gerência.

4.1.11. Tarefa 4: Consolidar as informações disponíveis

Nesta etapa é feita uma verificação entre as funções de gerência e as classes que devem suportar estas funções. Deve-se verificar se:

- A parte de monitoração da função pode obter todas as informações necessárias dos objetos e/ou;
- A parte de controle da função tem a influência necessária nos objetos.

- A parte de monitoração envolve recuperar valores de atributos e receber notificações. A parte de controle envolve criar e remover objetos, alterar valores e executar ações. Todas as restrições impostas pelas funções de gerência precisam ser atendidas.

Se uma função não é atendida por uma ou mais das classes existentes, então a tarefa 3 precisa ser refeita, novas classes podem ser criadas ou as existentes pode ser estendidas por especialização criando uma subclasse. Algumas classes podem existir apenas para serem utilizadas como superclasses de outras, e, portanto, não são diretamente relacionadas com uma função de gerência.

Para cada classe na TIB C, deve-se verificar se esta classe provê alguma razão para adicionar alguma função de gerência TMN adicional, ou se modifica alguma função já existente. Se alguma modificação ou adição de funções de gerência existe, a tarefa 2 deve ser novamente executada.

4.1.12. Tarefa 5: Definir a organização da informação de gerência

Determinar a organização da informação de gerência para cada tipo de sistema gerenciado como visto por uma aplicação ou sistema de gerência particular para o serviço de gerência escolhido. Verificar a organização pelo ponto de vista do sistema gerenciado.

Esta tarefa define a estrutura de nomeação dos objetos gerenciados (RDNs – Relative Distinguished Name e DN – Distinguished name).

4.1.13. TIB E: Organização da Informação de gerência

A organização da informação de gerência especifica o modelo de informação de um sistema gerenciado como visto por uma interface específica por uma aplicação ou sistema de gerência. Esse modelo de informação contém todos as classes de objetos que podem e serão providas por um sistema gerenciado para o sistema gerente ou aplicação. Em particular, define a estrutura de nomeação para os objetos dentro do sistema gerenciado. A organização da informação de gerência define todas as possíveis trocas de informação entre as aplicações de gerência e o sistema gerenciado. Para permitir melhor entendimento da organização, é necessário o uso de diagramas.

4.1.14. Tarefa 6: Definir os requisitos da comunicação

Criar conjuntos de requisitos de comunicação para os mais prováveis cenários de comunicação. Podem existir requisitos para transações simples, transferência de arquivos, acesso a arquivos, ou combinações entre estes. Outros requisitos tais como: capacidade de transmissão, confiabilidade, atrasos ou regras de nomeação podem vir a ser necessários.

4.1.15. TIB F: Requisitos de comunicação

Conjuntos de requisitos de comunicação contendo:

- Natureza da comunicação;
- Frequência, requisitos de serviço da camada de aplicação (OSI), atraso, etc.

4.1.16. Tarefa 7: Preparar o documento para as tarefas de especificação de protocolos

Os resultados das tarefas anteriores devem ser examinados para preparar a documentação necessária para a execução das tarefas de protocolo. Devido a Telebrás [TELE060] já ter definido os perfis funcionais que podem ser utilizados, as tarefas de protocolo não serão escopo deste trabalho, apenas deve-se escolher um ou mais dos perfis que se adequem ao problema.

4.1.17. TIB G: Perfis funcionais da TMN

Os perfis funcionais da TMN provêm toda a informação necessária para executar as tarefas associadas com a escolha e definição dos protocolos para uso nas interfaces TMN. Incluem-se nos perfis funcionais os resultados obtidos nas tarefas 0 a 7 que espera-se sejam úteis para a seleção.

4.2. Serviços de Gerência

Um serviço de gerência, como descrito no item 4.1, é percebido como uma área de atividades de gerência que provê suporte para um aspecto de OAM (Operação, Administração e Manutenção). Um serviço é sempre descrito pela percepção do usuário dos requisitos de OAM.

O ITU-T lista na recomendação [M.3200] um conjunto de serviços de gerência já identificados, mostrados na tabela abaixo:

Tabela 1 Serviços de Gerência TMN

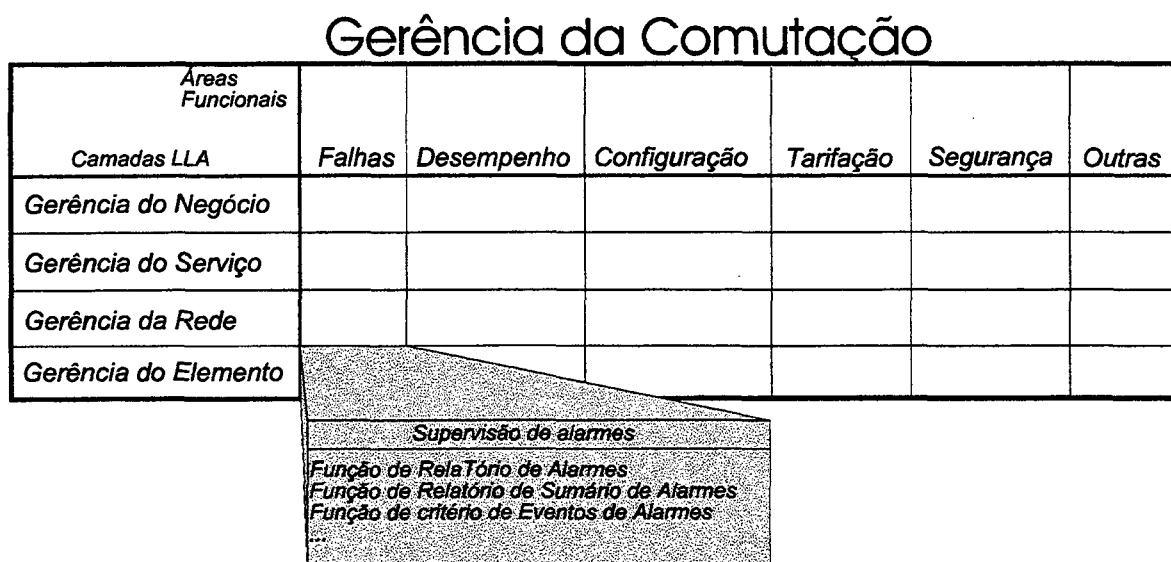
Serviços de Gerência TMN definidos na recomendação [M.3200]	
1	Administração de cliente
2	Administração de encaminhamento e análise de dígitos
3	Administração de medidas e análise de tráfego
4	Administração de tarifas e faturamento
5	Gerência da segurança da TMN
6	Gerência de tráfego
7	Gerência do acesso do cliente
8	Gerência da rede de transporte
9	Gerência da comutação
10	Gerência dos equipamentos em instalações dos clientes
11	Administração de instalação de sistemas
12	Administração da qualidade do serviço e do desempenho da rede
13	Gerência de serviço controlado pelo cliente
14	Gerência do sistema de sinalização por canal comum
15	Gerência de redes inteligentes
16	Restauração e recuperação
17	Gerência de materiais
18	Gerência da força de trabalho
19	Gerência da TMN

Os serviços de gerência não serão objeto de padronização pelo ITU-T, mas esta lista serve como subsídio para o processo de padronização das funções de gerência, classes de objetos gerenciados e mensagens. A lista não é exaustiva, podendo ser ampliada quando novos serviços forem identificados.

Para alguns destes serviços, os grupos de trabalho do ITU-T já identificaram e definiram seus requisitos, objetivos e contextos de gerência como definido na metodologia [M.3020] TIBs A e B, enquanto os outros ainda estão em estudo.[M.3200]

Um novo serviço de gerência deve ser definido utilizando-se o GDMS descrito no capítulo anterior, onde pode-se utilizar o diagrama abaixo para representar graficamente o relacionamento entre o serviço de gerência, os conceitos de LLA e as áreas funcionais de gerência. Cada uma das células do diagrama deve ser preenchida com um grupo de funções de gerência TMN para atender a camada LLA e a área funcional.

Figura L: Diagrama de serviços de gerência por LLA e área funcional



No diagrama mostrado, o serviço de gerência - gerência da comutação tem um componente que atende a área funcional de falhas - supervisão de alarmes, definida com as funções de gerência que devem ser utilizadas na camada gerência de elemento.

4.3. Funções de Gerência

Uma função de gerência TMN é a menor parte de um serviço de gerência como percebido pelo usuário do serviço, e consistirá geralmente em uma seqüência de ações em um ou mais objetos gerenciados definidos. As funções de gerência são agrupadas em conjuntos de funções de gerência que é a menor parte reutilizável.

O ITU-T na recomendação [M.3400] já definiu um conjunto de funções de gerência para uso dos serviços de gerência listados na recomendação [M.3200]. As funções são classificadas pela área funcional de gerência que atendem.

A partir destas funções de gerência um modelo genérico de rede (GNIM) foi proposto na recomendação [M.3100] definindo um conjunto de objetos gerenciados que procura atender às necessidades das funções já definidas. Neste modelo estão descritas classes que modelam os recursos mais genéricos encontrados na rede, sem no entanto se especializar quanto a aspectos de tecnologia, fabricantes ou características especiais.

As funções de gerência apesar de definidas para atender os serviços da recomendação [M.3200], procuram refletir os aspectos gerais de OAM&P da rede de telecomunicações.

Os conjuntos de funções de gerência são ainda acrescidos das funções de gerência de sistemas OSI (OSI SMFs), tais como: controle de logs, função de gerência de objetos, etc.

5. APLICAÇÃO DA METODOLOGIA

A metodologia mostrada anteriormente será aplicada passo a passo, visando ao final do processo se definir um modelo de informação que permita a gerência do SIPT AXS/20. As ferramentas existentes para automação das tarefas da metodologia são limitadas, caras e não cobrem todo o espectro das tarefas.

As ferramentas mais comuns atendem a tarefa de modelagem dos objetos, permitindo ao analista ter uma interface visual com a hierarquia de classes, facilitando o trabalho de escolha das classes e criação das novas classes necessárias. Somente foi utilizado o compilador GDMO/ASN.1 da ferramenta Solstice Agent Toolkit 2.0 como forma de conferir se a sintaxe das classes e das definições dos atributos estavam corretas e completas.

5.1. Tarefa 0: Gerar diretrizes

Um processo de compilação foi executado visando obter-se as informações necessárias para o entendimento e referências para as outras tarefas.

5.2. TIB 0: Diretrizes

Referências utilizadas:

- Modelagem orientada a objetos: [Booch91] e [Yourdon91].
- Ambiente de gerência OSI: [M.3010],[X.701],[X.721] e [BRISA].
- Diretrizes para a definição de objetos gerenciados (GDMO) [X.722].

5.3. Tarefa 1: Descrever os serviços de gerência TMN e seus objetivos pela perspectiva do usuário

Um SIPT é composto de uma central de comutação e pertence a rede de comutação da rede de telecomunicações, analisando-se a lista de serviços catalogados pelo ITU-T vê-se que o serviço de gerência - gerência de comutação é o serviço responsável pela gerência das funcionalidades de comutação, portanto aplicável a gerência do SIPT.

A gerência da comutação tem funções de gerência atendendo as áreas funcionais de desempenho, falhas e configuração. Neste presente trabalho a área funcional de desempenho não será considerada para o modelo de informação, sendo dada prioridade às áreas funcionais de falhas e configuração.

O SIPT pode ser considerado um elemento de rede (NE) para efeito de gerência, por ser um elemento com funções de telecomunicações e dispor de uma interface de gerência Q₃.

5.4. TIB A: Objetivos e serviços de gerência TMN

Serviços de gerência TMN selecionados e definidos:

Gerência da Comutação

Este serviço está descrito na recomendação [M.3200].

A gerência da comutação cobre as funções de comutação incluídas em uma central de comutação. Este serviço contém funções para as seguintes áreas funcionais, desempenho, falhas e configuração. Os objetivos da gerência de comutação são atingir os requisitos de qualidade de serviço determinados.

5.5. Tarefa 2: Descrever o contexto da gerência TMN

Na TIB B está definido o contexto de gerência que é composto de: papéis de gerência, recursos e funções de gerência TMN associado ao serviço de gerência TMN descrito na TIB A.

5.6. TIB B: Papéis da gerência TMN, recursos de telecomunicações e funções de gerência TMN

5.6.1. Contexto de gerência:

5.6.1.1. Papéis de gerência:

O SIPT será considerado um elemento de rede (NE) para a TMN, logo sendo gerenciado através de uma interface Q₃. Os OSs que utilizam o serviço de gerência de comutação atuam diretamente sobre o SIPT, executando as funções de gerência TMN que compõem este serviço.

5.6.1.2. Recursos de telecomunicações:

O SIPT se comporta na rede de telecomunicações como uma central de comutação, e portanto será tratado genericamente como tal, com as suas particularidades sendo especializações do modelo.

5.6.1.3. Funções de gerência TMN:

A seguir serão listadas as funções de gerência utilizadas na definição do modelo de informação, divididas por área funcional.

- Gerência de falhas

Gerência de falhas é um conjunto de funções que permite a detecção, isolamento, e correção de operação anormal da rede de telecomunicações e seu ambiente.

- Supervisão de alarmes

Uma TMN provê a capacidade de observar as falhas de um NE em quase tempo real. Na ocorrência de uma falha a indicação é gerada e disponibilizada pelo NE, baseado nisto a TMN determina a natureza e severidade da falha.

- Conjunto de funções relatório de alarmes

Permite ao NE informar a ocorrência de uma alarme. Provê a TMN com facilidades para definir os destinos de uma informação de alarme, quais as características do alarme para que ele seja enviado, habilitar ou desabilitar o envio dos alarmes.

- Conjunto de funções sumário de alarmes

Permite a TMN agendar, receber, ler a agenda , habilitar, desabilitar o sumário dos alarmes existentes em um NE.

- Conjunto de funções critério de eventos de alarmes

Permite a TMN definir quais os critérios que serão utilizados qualificar uma situação como em falha.

- Conjunto de funções gerência de indicação de alarme

Permite a TMN habilitar ou não as indicações locais (no NE) de alarme.

- Conjunto de funções controle de log

Permite a TMN criar, destruir, habilitar , desabilitar ao NE ao registro dos alarmes.

- Gerência de Configuração

Gerência de configuração é um conjunto de funções que permite a TMN controlar, identificar, coletar dados e enviar dados a um NE.

- Provisionamento

Consiste de procedimentos que sejam necessários para colocar um equipamento em serviço, sem incluir a instalação. O estado da unidade, em serviço, fora de serviço, em espera, reservado, também pode ser controlado pela TMN.

- Configuração de NE

Permite a TMN registrar a configuração do NE, identificar novas entidades, identificar a retirada de unidades, alterar parâmetros, etc.

- Conjunto de funções administrativas de NE

Permite a TMN comandar as funções administrativas, tais como: ajustar o relógio, iniciar ou terminar cópias de segurança, configurar o acesso e funcionalidades do usuário.

- Conjunto de funções gerência de base de dados de NE

Permite a TMN realizar operações em bases de dados do NE, tais como: inicialização, reinicialização, alteração, consulta, cópia de segurança.

- Controle e estado NE

Consiste na capacidade de monitorar e controlar certos aspectos do NE. Exemplos incluem a consulta ou alteração do estado de serviço de um NE ou de um componente.

- Conjunto de funções controle e estado genérico NE

Permite a TMN consultar e alterar o estado do NE, verificar entrada em serviço, etc.

5.7. TIB X: Modelos de informação genéricos e específicos de uma tecnologia

Esta TIB contém as classes definidas na GNIM da recomendação [M.3100] acrescidas das bibliotecas de classes definidas pelo NMF Omnipoint volume 1 [NMFVOL1] e volume 4 [NMFVOL4]. Neste TIB também fazem parte os modelos presentes nas recomendações ITU-T aplicáveis [X.721], [X.725], [X.730], [X.731], [X.732], [X.733], [X.734] e [X.735].

5.8. Tarefa 3: Modelagem de Objetos

Na metodologia não existe um padrão a ser adotado quanto ao processo de modelagem, apenas deve-se atender aos requisitos funcionais dos serviços de gerência. O processo de criação de classes é de inteira discricção do analista.

O uso das informações do modelo pela TMN é facilitado quando se utiliza classes já existentes na GNIM, recomendações ITU-T ou de uso geral (NMF), portanto deve-se utilizá-las quando estas atenderem às necessidades, especializando-as se necessário. Para os casos em que não existem classes adequadas serão criadas classes novas (derivadas de top).

Desta forma a TIB X foi analisada, procurando as classes que poderiam ser utilizadas da forma original, ou como superclasses, as classes escolhidas estão apresentadas na TIB C mostrada abaixo.

5.9. TIB C: Biblioteca de informações de gerência.

O modelo criado está descrito em GDMO no anexo A, e as sintaxes de transferência em linguagem ASN.1 no anexo B.

Para a supervisão de alarmes utilizou-se diretamente as classes definidas nas recomendações ITU-T [X.721], [X.733], [X.734], [X.735] e [Q.821], sendo incluídas no modelo as seguintes classes:

- *alarmRecord*. Definida na recomendação [X.721].
- *eventForwardingDiscriminator*. Definida na recomendação [X.721].
- *eventLogRecord*. Definida na recomendação [X.721].
- *log*. Definida na recomendação [X.721].
- *logRecord*. Definida na recomendação [X.721].
- *alarmSeverityAssignmentProfile*. Definida na recomendação [Q.821].
- *currentAlarmSummaryControl*. Definida na recomendação [Q.821].
- *managementOperationsSchedule*. Definida na recomendação [Q.821].

O modelo definido para a configuração da central AXS, como central de comutação genérica, em [schweitzer] foi aproveitado como base deste modelo. As classes acrescentadas representam a funcionalidade SIPT e seus componentes físicos e lógicos.

As classes criadas neste modelo estão listadas abaixo.

axsService

Uma instância desta classe representa um serviço disponível na central AXS com as suas características específicas. Derivada da classe *service* [NMFVOL4].

pa

Uma instância desta classe representa uma posição de atendimento refletindo seu estado de serviço e suas configurações. Derivada da classe *opEquipment* [NMFVOL1].

paOperator

Uma instância desta classe representa um operador de PA (atendente, Supervisora, etc.). Descreve o operador e seus privilégios no sistema.. Derivada da classe *top* [X.721].

paOperatorGroup

Uma instância desta classe representa um grupo de operadores. Indica quais operadores estão associadas a este grupo e quais as características deste grupo. Derivada da classe *top* [X.721].

operatorPrivilege

Uma instância desta classe representa um conjunto de privilégios de atendimento, definindo permissões e características de atendimento. Cada operador está associado a um conjunto de privilégios. Derivada da classe *top* [X.721].

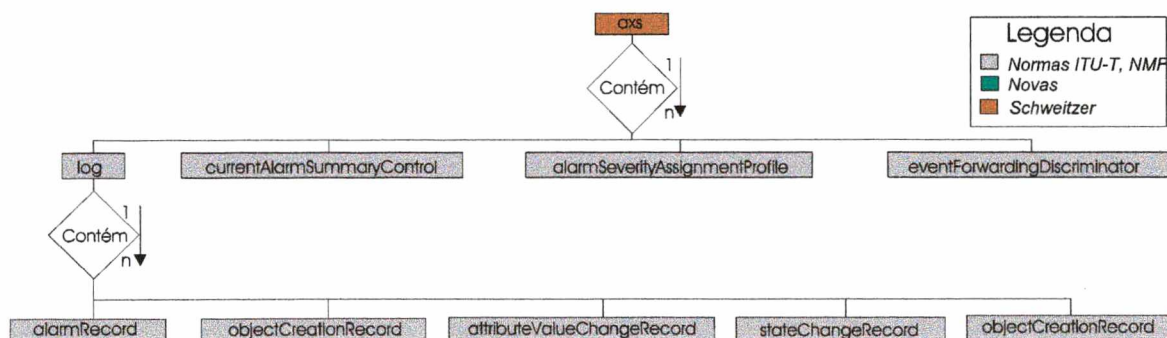
dacService

Uma instância desta classe representa um serviço DAC disponível na central AXS com as suas características específicas. Derivada da classe *axsService*.

5.10. TIB D: Diagrama de entidade-relacionamento

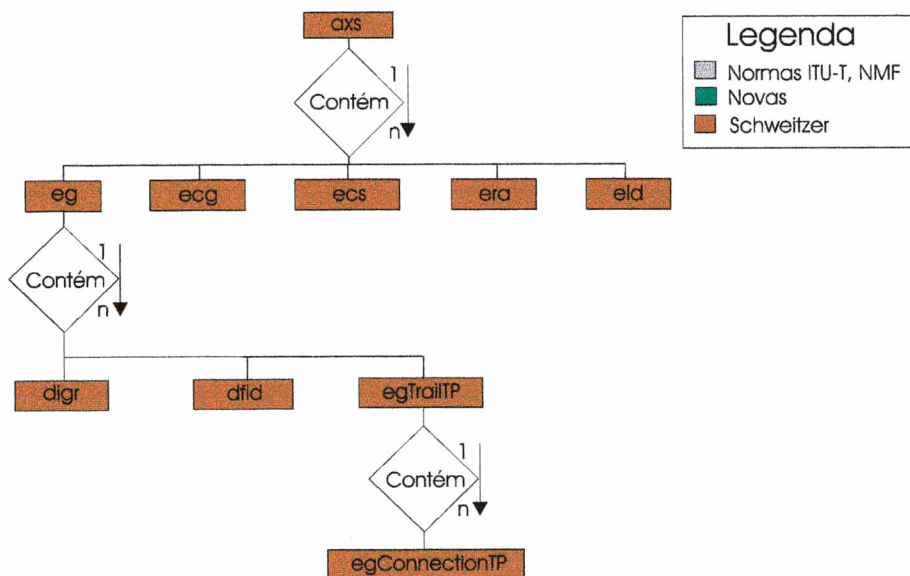
O diagrama ER abaixo mostra a relação entre a classe *axs* e as classes das recomendações do ITU-T relativas a alarmes.

Figura M: Diagrama ER *axs* (alarmes)



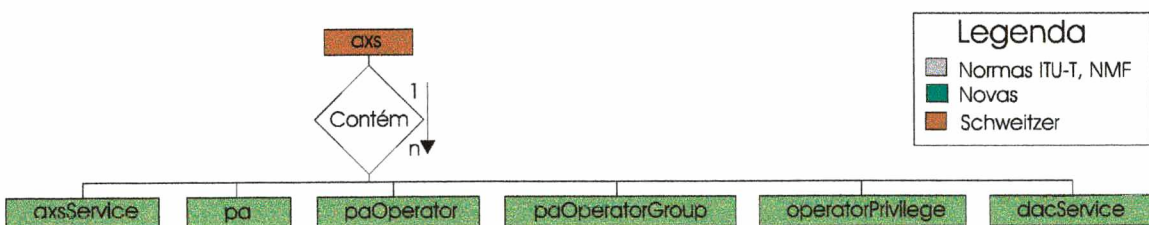
O diagrama ER abaixo mostra a relação entre os componentes que representam a estrutura física do axs.

Figura N: Diagrama ER axs (estrutura física)



O diagrama abaixo mostra a relação entre os componentes do serviço da plataforma AXS.

Figura O: Diagrama ER axs (Serviço)



O diagrama ER mostrado abaixo indica a relação de herança existente entre as classes presentes no modelo.

Figura P: Diagrama ER descrevendo a relação de herança

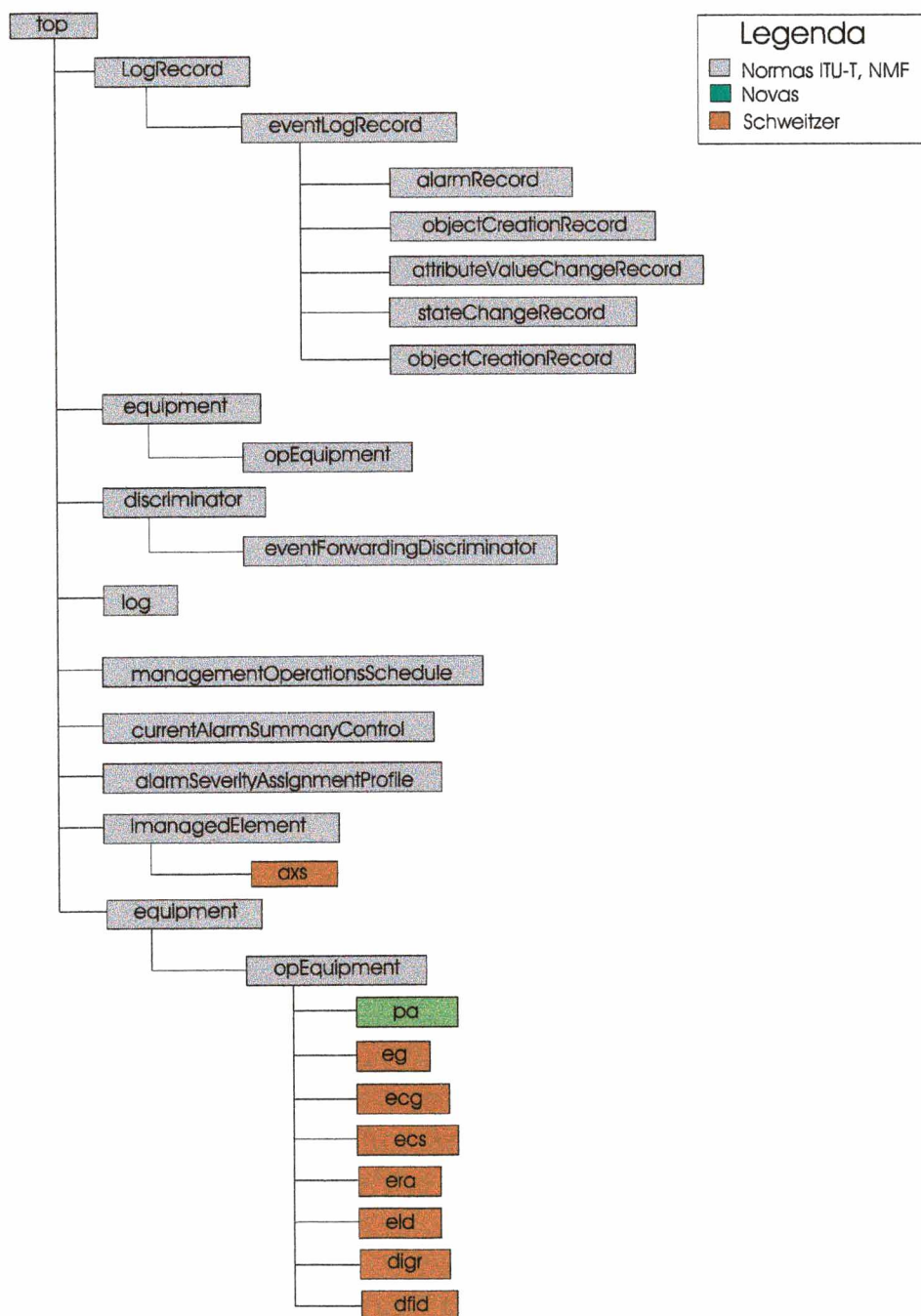
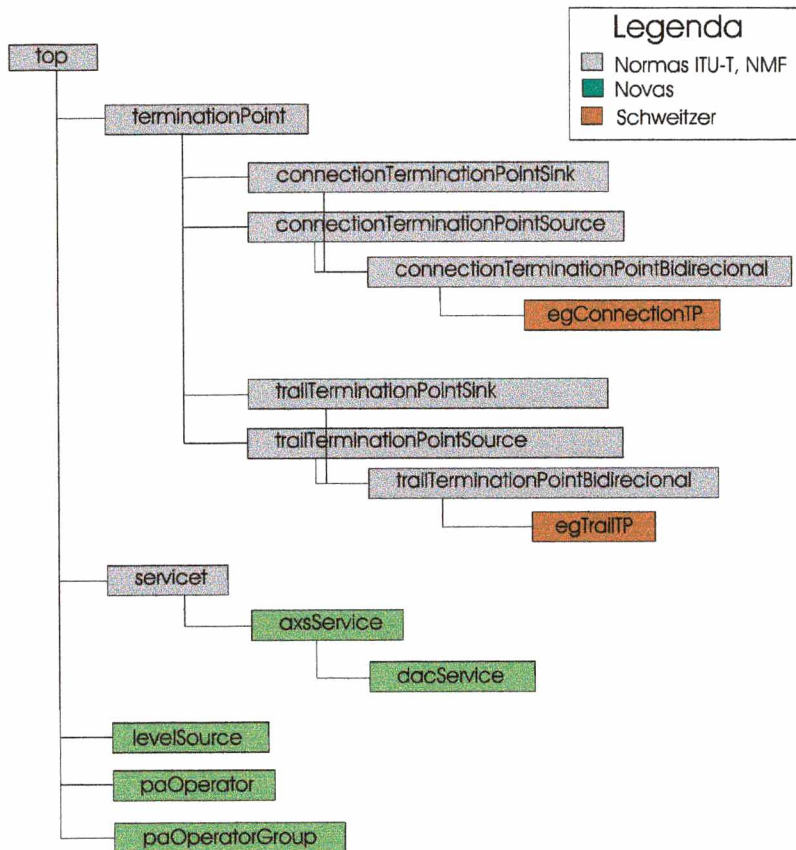


Figura Q: Diagrama ER descrevendo a relação de herança



5.11. Tarefa 4: Consolidar as informações disponíveis

A etapa de consolidação consiste na verificação se todas as funções de gerência contempladas no modelo estão sendo suportadas, e verificar se novas funções de gerência são geradas pelas classes criadas.

A tabela abaixo mostra as funções de gerência e as classes que a suportam.

Tabela 2 Funções de gerência e classes que a suportam

Conjunto de funções	Função de gerência	Classe
Relatório de alarmes	<i>Report alarm</i>	<i>managedElement</i>
		<i>acs</i>
	<i>Route Alarm Report</i>	<i>eventForwardingDiscriminator</i>
	<i>Request Alarm Report Route</i>	<i>eventForwardingDiscriminator</i>
	<i>Condition Alarm Report</i>	<i>eventForwardingDiscriminator</i>
	<i>Request Alarm Report Control condition</i>	<i>eventForwardingDiscriminator</i>
	<i>Allow/Inhibit Alarm Reporting</i>	<i>eventForwardingDiscriminator</i>
Sumário de alarmes	<i>Request Alarm History</i>	<i>log</i>
		<i>logRecord</i>
	<i>Report Current Alarm Summary</i>	<i>managementOperationsSchedule</i>
		<i>currentAlarmSummaryControl</i>
	<i>Route Current Alarm Summary</i>	<i>managementOperationsSchedule</i>
	<i>Request Current Alarm Summary Route</i>	<i>managementOperationsSchedule</i>
	<i>Schedule Current Alarm Summary Route</i>	<i>managementOperationsSchedule</i>
	<i>currentAlarmSummaryControl</i>	
Critérios de eventos de alarmes	<i>Request Current Alarm Summary Schedule</i>	<i>managementOperationsSchedule</i>
		<i>currentAlarmSummaryControl</i>
	<i>Allow/Inhibit Current Alarm Summary</i>	<i>managementOperationsSchedule</i>
Gerência de indicação de alarme	<i>Request Current Alarm Summary</i>	<i>managementOperationsSchedule</i>
		<i>currentAlarmSummaryControl</i>
Critérios de eventos de alarmes	<i>Condition Alarm Event Criteria</i>	<i>alarmSeverityAssignProfile</i>
	<i>Request Alarm Event Criteria</i>	<i>alarmSeverityAssignProfile</i>
Gerência de indicação de alarme	<i>Allow/Inhibit Audible/Visual Alarm Indications</i>	<i>acs</i>
	<i>Reset Audible/Visual Alarm</i>	<i>acs</i>
Controle de log	<i>Allow/Inhibit Logging</i>	<i>log</i>
	<i>Condition Logging</i>	<i>log</i>
	<i>Request Log Condition</i>	<i>log</i>
Configuração de NE	<i>Request Configuration</i>	<i>acs, acsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionTP, egTrailTP</i>
	<i>Configuration Report</i>	<i>acs, acsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionTP, egTrailTP</i>
	<i>Grow</i>	<i>acsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionTP, egTrailTP</i>
	<i>Prune</i>	<i>acsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionTP, egTrailTP</i>

	<i>Restore</i>	<i>axsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionIP, egTrailTP</i>
	<i>Assign</i>	<i>axsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionIP, egTrailTP</i>
	<i>Delete</i>	<i>axsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionIP, egTrailTP</i>
	<i>Set Service State</i>	<i>axsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionIP, egTrailTP</i>
	<i>Request Assignments</i>	<i>axsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionIP, egTrailTP</i>
	<i>Set Parameters</i>	<i>axs, axsService, pa, paOperator, paOperatorGroup, ecg, ecs, era, eg, dfid, dgir, egConnectionIP, egTrailTP</i>
	<i>Restart Request</i>	<i>axs, pa, ecg, ecs, era, eg, dfid, dgir</i>
	<i>Restart Report</i>	<i>axs, pa, ecg, ecs, era, eg, dfid, dgir</i>
Administração de NE	<i>Set Clock</i>	<i>axs</i>
	<i>Backup Copy</i>	<i>axs</i>
	<i>Terminate Procedure</i>	<i>axs</i>
	<i>Route Messages</i>	<i>eventForwardingDiscriminator</i>
	<i>Set Service Controls</i>	<i>axs</i>
Funções genéricas de gerência de estado e controle de NE	<i>Request Status</i>	<i>axs</i>
	<i>Status Report</i>	<i>axs</i>
	<i>Allow/Inhibit Automatic Restoration</i>	<i>axs, eg, ecg, ecs</i>

5.12. Tarefa 5: Definir a organização da informação de gerência

Nesta tarefa deve-se definir basicamente como os objetos instanciados serão nomeados e desta forma referenciados pela interface Q₃. Os *name-bindings* (padrão de nomeação para instâncias de uma classe) foram definidos para todas as classes criadas neste modelo de informação, em casos particulares a nomeação é implícita quando se especializa uma classe que permite a nomeação de suas subclasses.

Dentro da estrutura TMN, para um NE os dados referentes a aquele NE devem ficar referenciados a uma instância da classe *managedElement* ou subclasse desta, e portanto foi desta forma que a informação foi organizada.

5.13. TIB E: Organização da Informação de gerência

Os diagramas ER nas figuras: Figura M, Figura N e Figura O retratam a organização da informação. No caso dos diagramas mostrados, a relação “*de contém*” indica que existe uma relação de nomeação entre a instância que contém e a instância contida.

5.14. Tarefa 6: Definir os requisitos da comunicação

O sistema de atendimento se comportará como um NE para a TMN, logo contendo uma interface Q₃. Os requisitos de comunicação desta interface devem ser definidos para permitir a escolha dos protocolos das camadas OSI de 1 a 7 a serem utilizados.

5.15. TIB F: Requisitos de comunicação

De acordo com o disposto no apêndice I da recomendação [M.3010], deve-se examinar as características de desempenho da interface considerando: atraso, acuidade e disponibilidade; e as características de tráfego da interface considerando: quantidade, frequência e prioridade.

Para a interface de gerência com o modelo de informação do sistema de atendimento ao usuário apresentado pode-se considerar os seguintes parâmetros para a gerência de configuração:

Tabela 3 Características da interface para a gerência de configuração

Parâmetros		Requisito
Desempenho	Atraso	permite longo
	Acuidade	alta
	disponibilidade	alta
Tráfego	Quantidade	média
	Frequencia	esparsa
	Prioridade	média

Para aos dados da gerência de falha temos os seguintes dados:

Tabela 4 Características da interface para a gerência de falhas

Parâmetros		Requisito
Desempenho	Atraso	curto
	Acuidade	alta
	disponibilidade	alta
Tráfego	Quantidade	baixa
	Frequencia	esparsa
	Prioridade	alta

Como os requisitos não são excludentes, pode-se utilizar a mesma interface para a troca de informações tanto da gerência de configuração quanto da gerência de falhas.

5.16. Tarefa 7: Preparar o documento para as tarefas de especificação de protocolos

5.17. TIB G: Perfis funcionais da TMN

A TIB G é composta das outras TIBs mais as informações dos perfis funcionais previstos para as interfaces. O ITU-T nas recomendações [Q.811] e [Q.812] define os perfis funcionais aceitáveis para a realização de uma interface Q₃. A Telebrás refinou as recomendações [Q.811] e [Q.812] e definiu na prática [TELE060] os perfis funcionais que são utilizáveis nas interfaces Q₃ para a rede brasileira.

A camada 7 utiliza o protocolo CMIP, a camada 5 e 6 suportam o CMIP e portanto são apenas selecionados alguns parâmetros desses protocolos. camadas de 1 a 4 não serão objeto de escolha deste trabalho porque são escolha do implementador no momento da definição da estrutura de conexão de rede do NE.

5.18. Implementação e Teste do Modelo de Informação

O modelo de informação gerado pela aplicação da metodologia foi utilizado para desenvolver um agente Q_3 para a sistema SIS da Telepar. O SIS fornece uma série de serviços para os usuários daquela operadora, tais como: 106 (serviço de informações da operadora), e outros. O SIS é uma plataforma de serviços baseada no SIPT da Dígito Tecnologia.

O agente Q_3 desenvolvido serve como interface entre a plataforma SIS e o sistema de gerência TMN que está sendo implantado pela Telepar. As vantagens obtidas são várias, sendo as mais importantes: possibilidade de ter as informações de falha concentradas em um centro de gerência, facilitando a identificação de causas e a otimização da rede pelos analistas; capacidade de executar alterações de configuração do centro de gerência, permitindo aos analistas reconfigurar o SIS, reagindo a mudanças internas ou externas (falhas, congestionamentos, etc). Com a utilização de uma TMN e principalmente com a gerência desta nos serviços de atendimento providos pelo SIS espera-se uma ganho significativo na eficiência da rede (taxa de congestionamento, número de operadores) e diminuição das perdas por falhas na rede.

6. CONCLUSÕES

As empresas operadoras estão inseridas em um mercado cada vez mais competitivo, onde diversos aspectos estão em permanente mudança. A desregulamentação, as mudanças tecnológicas e a possibilidade de concorrência aos monopólios são fatores que tem e terão fundamental importância para a sobrevivência e competitividade neste mercado.

O uso de protocolos de gerência padronizados se torna uma necessidade para as empresas, principalmente pelos aspectos de facilidade de integração com outros sistemas de gerência, redução no tempo de implantação e gerência de novas tecnologias na rede, possibilidade de agregar novos serviços com maior velocidade, redução de custos, etc.

Os serviços de atendimento ao usuário são serviços de grande visibilidade, sendo um dos principais mecanismos de aferição intuitiva de qualidade do sistema de telecomunicações pelos seus usuários. Esta percepção de qualidade exige das empresas operadoras uma grande atenção à qualidade destes serviços. A TMN se mostra como uma poderosa ferramenta para que estes requisitos de qualidade sejam atendidos a custos compatíveis.

Este trabalho visou o desenvolvimento de um modelo de informação para os serviços de atendimento, o que é um passo importante para a integração de um equipamento e/ou sistema a uma TMN. Dentre as necessidades de gerência foram privilegiadas as que foram consideradas mais importantes para a operação, administração e manutenção desses sistemas, que são a gerência de configuração e de falhas.

O ITU-T através da [M.3020] padronizou a metodologia que deve ser adotada para o desenvolvimento uma interface de gerência, que compreende o processo de especificação de um modelo de informação. Esta metodologia foi aplicada ao estudo e modelagem da implementação da arquitetura SIPT feita pela Dígito. A metodologia se baseia principalmente na decomposição funcional dos requisitos de gerência e no uso da orientação a objetos como forma de definir o modelo a ser adotado.

O uso da orientação a objetos na modelagem da informação se mostrou extremamente interessante, porque permitiu a reutilização de um grande volume de definições previamente desenvolvidas pelo ITU-T, NMF e outros, e pela forma simples de definir relações e hierarquias entre os objetos e classes criadas para este modelo. O uso de objetos e classes permite a definição de conceitos genéricos e especializá-los quando necessário, facilitando tanto a especificação dos modelos de informação de sistemas específicos, quanto a criação de aplicações de gerência mais genéricas.

A decomposição funcional é uma parte importante do processo de especificação, e quanto mais completa e detalhada, mais simples é a modelagem.

A experiência deste trabalho mostrou que o processo de modelagem se desenrola de forma iterativa, em que cada passo identifica pontos fortes e fracos dos resultados obtidos em cada tarefa. O modelo de informação se refina a cada iteração se tornando mais completo e coerente. A tarefa de consolidação, que pode ser considerada uma conferência dos resultados obtidos até aquele passo, é a principal ferramenta para a identificação das falhas do modelo e principalmente possíveis aperfeiçoamentos.

A modelagem dos objetos é feita de forma não padronizada, e a experiência do analista é extremamente valiosa para a escolha adequada das classes a serem reutilizadas e na definição das novas classes a serem criadas. Neste processo, há grande necessidade de se desenvolver formas de documentar o processo de criação, permitindo que esta experiência não fique restrita apenas a equipe responsável pelo modelo.

O modelo de informações gerado tem como principal falha o não atendimento da gerência de desempenho, o que dificulta a realização de uma gerência preventiva, identificando situações de início de congestionamento, etc. A gerência de desempenho, então, é um aperfeiçoamento natural deste modelo, que para ser executado, deve acrescentar as funções de gerência pertencente a esta área funcional na TIB B e refazer as outras tarefas considerando as TIBs obtidas neste trabalho como ponto de partida.

O custo de desenvolvimento dos modelos de informação e a possibilidade de manter como proprietárias algumas funcionalidades dos seus equipamentos, tem levado os fabricantes de equipamentos de telecomunicações a considerar os modelos de informação como informação confidencial, logo disponível somente para os proprietários e usuários destes equipamentos. Esta posição dos fabricantes gera grande dificuldade em se encontrar na literatura modelos de informação completos que possam servir como base de estudo e discussão para estudantes e profissionais da área, o que facilitaria a troca de experiências.

O uso de uma metodologia padronizada é um passo importante na obtenção de sistemas de gerência interoperáveis, fornecendo então todas as vantagens já previstas, tais como: redução de custos, aumento da qualidade dos serviços, maior interatividade com os usuários, etc.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [Adams96] E. K. Adams, K. J. Willetts. *The Lean Communications Provider*, McGraw-Hill, 1996.
- [Aidarous98] S. Aidarous, T. Plevyak. *Telecommunications Network Management*, IEEE Press, 1998.
- [Booch91] G. Booch. *Object Oriented Design*. The Benjamin/Cummings, 1991.
- [BRISA] BRISA. *Gerenciamento de Redes - Uma Abordagem de Sistemas Abertos*. Makron Books, 1993.
- [M.3000] ITU-T. *Recommendation M.3000. Maintenance - Telecommunications Management Network - Overview Of Tmn Recommendations*. ITU-T 1994
- [M.3010] ITU-T. *Recommendation M.3010. Maintenance - Telecommunications Management Network - Principles for a Telecommunications management network*. ITU-T, 1996
- [M.3020] ITU-T. *Recommendation M.3020. TMN Interface Specification Methodology*. ITU-T, 1995.
- [M.3100] ITU-T. *Recommendation M3100. Generic Network Information Model*. ITU-T, 1995.
- [M.3200] CCITT. *Recommendation M.3200. TMN Management Services: Overview*. CCITT, 1992.
- [M.3400] CCITT. *Recommendation M.3400. TMN Management Functions*. CCITT, 1992.
- [Mendonça] J. F. F. Mendonça, J. M. S. Nogueira. *Desenvolvimento de Agentes OSI para Gerência de Desempenho de Sistemas de Transmissão Digital PDH - Plesiochronous Digital Hierarchy*. Anais XIV SBRC, Fortaleza, CE, Maio de 1996.
- [NEC] NEC do Brasil. *Sistema de Comutação Crossbar - Sinalização*. NEC, 1975.
- [NMFVOL1] NMF. *Omnipoint Version1 Library Volume 1*. NMF, 1995.
- [NMFVOL1] NMF. *Omnipoint Version1 Library Volume 4*. NMF, 1995.
- [Q.811] ITU-T. *Recommendation Q.811. Specifications of Signalling System no. 7 - Q₃ Interface - Lower layer protocol profiles for the Q3 and X interfaces - Alarm Surveillance*. ITU-T, 1997.

- [Q.812] ITU-T. *Recommendation Q.812. Specifications of Signalling System no. 7 - Q₃ Interface - Upper Layer Protocol Profiles for the Q₃ Interface.* ITU-T,1993.
- [Q.821] ITU-T. *Recommendation Q.821. Specifications of Signalling System no. 7 - Q₃ Interface - Stage 2 and Stage 3 Description for the Q₃ Interface - Alarm Surveillance.* ITU-T,1993.
- [Schweitzer] A. Schweitzer. *Modelo de Informação de Gerência para a central AXS/20.* Dissertação de Mestrado apresentado à Universidade Federal de Santa Catarina. 1996.
- [Takanohashi] M. Takanohashi. *Padronização em gerenciamento de redes de telecomunicações: estudo e especificação de uma interface Q₃ da TMN.* Dissertação de Mestrado apresentada à Escola Politécnica da Universidade de São Paulo. 1993.
- [TELE059] TELEBRAS. *Prática TELEBRAS SDT 210-001-059: Especificação de Perfis Funcionais de Protocolos para Interface Q_x para Conexão entre Equipamentos / Sistemas da Rede de Telecomunicações à Rede de Gerência.* TELEBRAS, 1993.
- [TELE060] TELEBRAS. *Prática TELEBRAS SDT 210-001-060: especificação de Perfis Funcionais de Protocolos para Interface Q₃ da Rede de Gerência de Telecomunicações.* TELEBRAS, 1993.
- [TELE109] TELEBRAS. *Prática TELEBRAS SDT 501-100-109: Gerência Integrada de Rede - Ações a Curto Prazo.* TELEBRAS, 1992.
- [TELE111] TELEBRAS. *Prática TELEBRAS SDT 501-100-111: Serviços de Gerência, Requisitos de Funcionalidade de Gerência da Camada de Elemento de Rede.* TELEBRAS, 1994.
- [TELE704] TELEBRAS. *Prática TELEBRAS SDT 220-700-704: Especificação Funcional do Sistema Integrado de Posições de Tráfego.* TELEBRAS, 1990.
- [X.200] CCITT. *Recommendation X.200. Data Communication Networks - Open Systems Interconnection - Model and Notation, Service Definitions.* CCITT, 1988.
- [X.208] CCITT. *Recommendation X.208. Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).* CCITT, 1988.
- [X.701] CCITT. *Recommendation X.701. Information Technology - Open Systems Interconnection - Systems Management Overview.* CCITT, 1992.

- [X.710] CCITT. *Recommendation X.710. Common Management Information Service Definition for CCITT Applications*. CCITT, 1992.
- [X.711] CCITT. *Recommendation X.711. Common Management Information Protocol Specification for CCITT Applications*. CCITT, 1992.
- [X.720] CCITT. *Recommendation X.720. Information Technology - Systems Interconnection - Structure Of Management Information: Management Information Model*. CCITT, 1992.
- [X.722] CCITT. *Recommendation X.722. Information Technology – Open Systems Interconnection - Structure Of Management Information: Guidelines for the Definition of Managed Objects*. CCITT, 1992.
- [X.724] ITU-T. *Recommendation X.724. Information Technology – Open Systems Interconnection - Structure Of Management Information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management*. ITU-T, 1996.
- [X.725] ITU-T. *Recommendation X.725. Information Technology – Open Systems Interconnection – Structure of management Information: General Relationship Model*. ITU-T, 1995.
- [X.730] CCITT. *Recommendation X.730. Information Technology – Open Systems Interconnection - Systems Management: Object Management Function*. CCITT, 1992.
- [X.731] CCITT. *Recommendation X.731. Information Technology – Open Systems Interconnection – Systems Management: State Management Function*. CCITT, 1992.
- [X.732] CCITT. *Recommendation X.732. Information Technology – Open Systems Interconnection – Systems Management: Attributes for Representing Relationships*. CCITT, 1992.
- [X.733] CCITT. *Recommendation X.733. Information Technology – Open Systems Interconnection – Systems Management: Alarm Reporting Function*. CCITT, 1992.
- [X.734] CCITT. *Recommendation X.734. Information Technology – Open Systems Interconnection – Systems Management: Event Report Management Function*. CCITT, 1992.
- [X.735] CCITT. *Recommendation X.735. Information Technology – Open Systems Interconnection – Systems Management: Log Control Function*. CCITT, 1992.

[Yourdon91] E. Yourdon, P. Coad. *Object Oriented Analysis*. Second Edition, Yourdon Press, 1991.

ANEXO A: DESCRIÇÃO DO MODELO DE INFORMAÇÃO GDMO

```
-- Copyright (c) 27/11/97 by FLUG Telematica e Automacao Ltda. Todos os
-- direitos reservados
--
MODULE "AXS-MIB"

-- =====
-- axs MANAGED OBJECT CLASS
-- =====

axs MANAGED OBJECT CLASS
  DERIVED FROM "Rec. M.3100 : 1992":managedElement;
  CHARACTERIZED BY
    axsPkg PACKAGE
      BEHAVIOUR
        axsPkgBehaviour BEHAVIOUR
      DEFINED AS
        ! A classe axs e a classe de objetos gerenciados que representa a
        Central Telefonica AXS/20.
        !;;
  ATTRIBUTES
    copyrighth GET,
    configuration GET,
    performance GET,
    identification GET-REPLACE;;;
REGISTERED AS { axsObjectClass 1 };

-- =====
-- ecg MANAGED OBJECT CLASS
-- =====

ecg MANAGED OBJECT CLASS
  DERIVED FROM "Op1 Library Vol. 1":opEquipment;
  CHARACTERIZED BY
    ecgPkg PACKAGE
      BEHAVIOUR
        ecgPkgBehaviour BEHAVIOUR
      DEFINED AS
        ! A classe ecg e a classe de objetos gerenciados que representa o
        "Estagio de Coordenacao e Gerenciamento (ECG)".
        O ECG e a unidade de processamento principal que executa em um
        microcomputador com sistema Operacional Solaris. Se o sistema AXS
        possuir dualidade, existiraos dois ECG's, sendo um ativo e outro
        reserva
        !;;
  ATTRIBUTES
    internalId GET,
    ipAddress GET,
    codeStatus GET,
    ecgPreference GET-REPLACE,
    matrixPreference GET-REPLACE,
    actualStatus GET,
    hostname GET;;;
REGISTERED AS { axsObjectClass 2 };
```



```

-- =====
--  ecs MANAGED OBJECT CLASS
-- =====

ecs MANAGED OBJECT CLASS
DERIVED FROM "Op1 Library Vol. 1":opEquipment;
CHARACTERIZED BY
ecsPkg PACKAGE
  BEHAVIOUR
  ecsPkgBehaviour BEHAVIOUR
  DEFINED AS
  ! A classe ecs e a classe de objetos gerenciados que
  representa o "Estagio de Comutacao e Sincronismo (ECS)".
  O ECS implementa a matriz de comutacao do sistema.
  Se o sistema AXS possuir dualidade,
  existiraos dois ECS's, sendo um ativo e outro reserva
  !;;
  ATTRIBUTES
    internalId          GET,
    actualStatus        GET,
    codeStatus          GET;
  ACTIONS
    codeTransfer;;;
REGISTERED AS { axsObjectClass 3 };

```

```

-- =====
--  elc MANAGED OBJECT CLASS
-- =====

elc MANAGED OBJECT CLASS
DERIVED FROM "Op1 Library Vol. 1":opEquipment;
CHARACTERIZED BY
elcPkg PACKAGE
  BEHAVIOUR
  elcPkgBehaviour BEHAVIOUR
  DEFINED AS
  ! A classe elc e a classe de objetos gerenciados que
  representa o "Estagio de Linha (ELC)".
  Se o sistema AXS possuir dualidade,
  existiraos dois ELC's, sendo um ativo e outro reserva
  !;;
  ATTRIBUTES
    internalId          GET,
    elcPreference       GET-REPLACE,
    codeStatus          GET,
    ipAddress           GET,
    actualStatus        GET,
    hostname             GET;;;
REGISTERED AS { axsObjectClass 4 };

```

```

-- =====
-- eg MANAGED OBJECT CLASS
-- =====

```

```

eg MANAGED OBJECT CLASS
  DERIVED FROM "OP1 Library Vol. 1":opEquipment;
  CHARACTERIZED BY
  egPkg PACKAGE
    BEHAVIOUR
    egPkgBehaviour BEHAVIOUR
  DEFINED AS
    ! A classe eg e a classe de objetos gerenciados que
    representa o "Estagio de Grupo". O EG e' o componente
    da central no qual sao conectados os troncos de/para outras
    centrais.
    !;;
  ATTRIBUTES
    internalId      GET,
    matrixPreference GET-REPLACE,
    processorStatus GET,
    cardType        GET-REPLACE,
    codeStatus      GET;
  ACTIONS
    egReset,
    codeTransfer;;;
REGISTERED AS { axsObjectClass 5 };

```

```

-- =====
-- eld MANAGED OBJECT CLASS
-- =====

```

```

eld MANAGED OBJECT CLASS
  DERIVED FROM "OP1 Library Vol. 1":opEquipment;
  CHARACTERIZED BY
  eldPkg PACKAGE
    BEHAVIOUR
    eldPkgBehaviour BEHAVIOUR
  DEFINED AS
    ! A classe eld e' a classe de objetos gerenciados que
    representa o componente "Estagio de Linha Digital (ELD)".
    O ELD e o componente onde os sistemas digitais sao conectados
    e concentrados para comutar com a matriz de comutacao (ECS)
    !;;
  ATTRIBUTES
    internalId GET,
    codeStatus GET;
  ACTIONS
    codeTransfer;;;
REGISTERED AS { axsObjectClass 6 };

```

```

-- =====
-- era MANAGED OBJECT CLASS
-- =====

era MANAGED OBJECT CLASS
DERIVED FROM "OP1 Library Vol. 1":opEquipment;
CHARACTERIZED BY
eraPkg PACKAGE
  BEHAVIOUR
  eraPkgBehaviour BEHAVIOUR
  DEFINED AS
  ! A classe era e' a classe de objetos gerenciados que
  representa o componente da central "Estagio de Resposta Audivel
(ERA)".
  O ERA consiste de um microcomputador com sistema operacional
Solaris,
  e executa os programas relacionados ao armazenamento e
processamento
  de voz digitalizada.
  !;;
  ATTRIBUTES
  internalId GET,
  codeStatus GET;
  ACTIONS
  codeTransfer;;;
REGISTERED AS { axsObjectClass 7 };

-- =====
-- digr MANAGED OBJECT CLASS
-- =====

digr MANAGED OBJECT CLASS
DERIVED FROM "OP1 Library Vol. 1":opEquipment;
CHARACTERIZED BY
digrPkg PACKAGE
  BEHAVIOUR
  digrPkgBehaviour BEHAVIOUR
  DEFINED AS
  ! A classe digr representa componentes de
  hardware da central que implementa as portas fisicas
  de uma conexao.
  !;;
  ATTRIBUTES
  internalId GET;;;
REGISTERED AS { axsObjectClass 12 };

```

```

=====
-- egConnectionTP MANAGED OBJECT CLASS
=====

egConnectionTP MANAGED OBJECT CLASS
  DERIVED FROM "Rec. M.3100 :
1992":connectionTerminationPointBidirectional;
  CHARACTERIZED BY
  egConnectionTpPkg PACKAGE
  BEHAVIOUR
  egConnectionTpPkgBehaviour BEHAVIOUR
  DEFINED AS
  ! Representa a classe de objetos gerenciados que origina e termina
  uma conexao.
  O valor dos atributos desta classe depende do padrao de
configuracao
ao qual ela esta associada. O padrao e' determinado pelo atributo
cfgConnectionTPId. O atributo cfgConnectionTPId identifica
unicamente uma instancia da classe cfgConnectionTP.
Os atributos condicionais "Rec. X.721 | ISO/IEC 10165-2 :
1992":alarmStatus e
"Rec. X.721 | ISO/IEC 10165-2 : 1992":operationaState
estao presente em todas as instancias desta classe.
!;;
  ATTRIBUTES
  internalId GET,
  cfgConnectionTPId GET-REPLACE,
  egConnectionTPStatus GET,
  associatedService GET,
  sinalizationType GET,
  cfgStatus GET,
  terminationPointType GET,
  commonChannelSignaling GET,
  connectionTPType GET,
  lineSinalization GET,
  signalPath GET,
  mfcStatus GET,
  timeoutALE GET,
  timeoutMFC GET,
  directionality GET;;;
REGISTERED AS { axsObjectClass 8 };

```

```
-- =====
-- egTrailTP MANAGED OBJECT CLASS
-- =====
```

```
egTrailTP MANAGED OBJECT CLASS
```

```
DERIVED FROM "Rec. M.3100 : 1992":trailTerminationPointBidirectional;
```

```
CHARACTERIZED BY
```

```
egTrailTPPkg PACKAGE
```

```
BEHAVIOUR
```

```
egTrailTPPkgBehaviour BEHAVIOUR
```

```
DEFINED AS
```

```
! Representa a classe de objetos gerenciados que origina e termina
um link de 8 Mbs.
```

```
O atributos condicionais "Rec. X.721 | ISO/IEC 10165-2 :
1992":alarmStatus
```

```
e "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState
```

```
estao presentes em todas as instancias desta classe.
```

```
!;;
```

```
ATTRIBUTES
```

```
internalId          GET,
interfaceType       GET-REPLACE,
egTrailTPType       GET-REPLACE,
destination          GET-REPLACE,
egTrailTPStatus     GET-REPLACE,
connectionType       GET-REPLACE,
timeSlotType        GET-REPLACE;;;
```

```
REGISTERED AS { axsObjectClass 9 };
```

```
-- =====
-- dfid MANAGED OBJECT CLASS
-- =====
```

```
dfid MANAGED OBJECT CLASS
```

```
DERIVED FROM "Op1 Library Vol. 1":opEquipment;
```

```
CHARACTERIZED BY
```

```
dfidPkg PACKAGE
```

```
BEHAVIOUR
```

```
dfidPkgBehaviour BEHAVIOUR
```

```
DEFINED AS
```

```
! A classe dfid representa componentes de hardware
que realizam o processamento do sinal digital usado na troca
de sinalizacao entre centrais.
```

```
!;;
```

```
ATTRIBUTES
```

```
internalId          GET,
numberOfDTMFfilter GET,
filteringType       GET-REPLACE,
masterProcessor      GET,
filterType          GET-REPLACE;;;
```

```
REGISTERED AS { axsObjectClass 13 };
```

```

-- =====
-- axService  MANAGED OBJECT CLASS
-- =====

```

```

axService  MANAGED OBJECT CLASS
  DERIVED FROM "OP1 Library Vol. 4":service;
  CHARACTERIZED BY
  axServicePkg PACKAGE
  BEHAVIOUR
  axServicePkgBehaviour BEHAVIOUR
  DEFINED AS
  ! A classe axService representa grande parte dos servicos
  disponivel na central AXS.
  !;;
  ATTRIBUTES
  description          GET,
  finalSelection      GET-REPLACE,
  answerSignal        GET-REPLACE,
  checkTone           GET-REPLACE,
  waitTime            GET-REPLACE,
  serviceTPstatus     GET,
  internalKey         GET,
  serviceStatus       GET;
  ACTIONS
  activatedService,
  deactivatedService,
  deactivatedServiceImmediately;;;
  CONDITIONAL PACKAGES
  hostConfigPackage  PRESENT IF "uma instance desta classe o suporta",
  paAllocationPackage  PRESENT IF "uma instance desta classe o
suporta",
  paTrainingPackage  PRESENT IF "uma instance desta classe o suporta",
  workTimePackage    PRESENT IF "uma instance desta classe o suporta";
  REGISTERED AS { axObjectClass 14 };

```

```

-- =====
-- pa MANAGED OBJECT CLASS
-- =====

pa MANAGED OBJECT CLASS
  DERIVED FROM "OP1 Library Vol. 1":opEquipment;
  CHARACTERIZED BY
    paPkg PACKAGE
      BEHAVIOUR
        paPkgBehaviour BEHAVIOUR
      DEFINED AS
        ! A classe pa e' a classe de objetos gerenciados que representa
        uma posicao de atendimento (PA). Uma PA e' o elemento no qual um
operador
        se conecta ao sistema e estabelece o audio para o atendimento de
        chamadas. A PA pode ser discada ou digital. PA discada e' uma
posicao
        de atendimento que se utiliza de uma linha telefonica normal para
        estabelecer audio com o sistema. A PA digital se conecta com o AXS
        via ELD (Estagio de Linha Digital).
        A PA possui uma fila de atendimento, onde as chamadas entrantes
        aguardam na fila para serem atendidas.
        O atributo paQueue define a fila de atendimento de chamadas.
        O atributo condicional "Rec. X.721 | ISO/IEC 10165-2 :
1992":usageState
        esta presente em todas as instancias desta classe
        !;;
  ATTRIBUTES
    paType          GET,
    paNumber        GET,
    paLogicNumber   GET,
    consoleType     GET,
    paBasicState    GET,
    paComplementState GET,
    paDetailState   GET,
    paQueue         GET;;;
REGISTERED AS { axsObjectClass 16 };

```

```

-- =====
-- paOperatorGroup MANAGED OBJECT CLASS
-- =====

paOperatorGroup MANAGED OBJECT CLASS
  DERIVED FROM "Rec. M.3100 : 1992":software;
  CHARACTERIZED BY
    paOperatorGroupPkg PACKAGE
    BEHAVIOUR
      paOperatorGroupPkgBehaviour BEHAVIOUR
    DEFINED AS
      ! Esta classe representa um Grupo de Operadores. Um operador pode
      ou nao estar associado a um ou mais grupos. Cada grupo possui uma
      fila de atendimento que contem as chamadas destinadas aos
operadores
      pertencentes a este grupo. Um grupo pode ser definido como normal
ou
      transbordo. Um grupo de transbordo e' aquele que atende as chamadas
      de um outro grupo. Em cada grupo pode ser definido uma condicao de
      transbordo ( condicao que define quando as chamadas deste grupo
      sera transferidas a um grupo de transbordo), que pode ser por
      tempo de espera de uma chamada ou pela quantidade de chamadas na
      fila de um grupo.
      A operacao Create/Delete deve ser usada para criar/remover
      instancia desta classe. Estas operacoes de gerenciamento permitem
      criar ou excluir um grupo de operador do sistema AXS.
      !;;
  ATTRIBUTES
    groupID          GET,
    groupType        GET-REPLACE,
    groupElemNumber  GET,
    groupQueue       GET,
    groupQueueSize   GET-REPLACE,
    overflowCondition GET-REPLACE,
    overflowTimeCondition GET-REPLACE,
    groupOperators   GET-REPLACE ADD-REMOVE,
    overflowedGroup  GET-REPLACE ADD-REMOVE,
    overflowingGroup GET-REPLACE ADD-REMOVE;;;
  REGISTERED AS { axsObjectClass 17 };

```



```

-- =====
-- paOperator MANAGED OBJECT CLASS
-- =====

```

```

paOperator MANAGED OBJECT CLASS
  DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    paOperatorPkg PACKAGE
      BEHAVIOUR
        paOperatorPkgBehaviour BEHAVIOUR
      DEFINED AS
        ! Esta classe representa operadores da PA (Atendentes, Supervisora,
etc.).
        A operacao Create/Delete deve ser usada para criar/remover
uma instancia desta classe. Estas operacoes de gerenciamento
permitem
criar ou remover um operador do sistema AXS. Um operador pode ou
nao pertencer a um grupo de operadores . O atributo associatedGroup
identifica os grupos a qual o operador pertence
!;;
  ATTRIBUTES
    operatorName          GET,
    operatorPassword      GET-REPLACE,
    privilegeNumber       GET-REPLACE,
    attendingPriority      GET-REPLACE,
    associatedGroup        GET-REPLACE ADD-REMOVE;
  NOTIFICATIONS
    "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
    "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion;;;
REGISTERED AS { axsObjectClass 18 };

```

```

-- =====
-- cfgEgConnectionTP MANAGED OBJECT CLASS
-- =====

cfgConnectionTP MANAGED OBJECT CLASS
DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
  cfgConnectionTpPkg PACKAGE
  BEHAVIOUR
  cfgConnectionTpPkgBehaviour BEHAVIOUR
  DEFINED AS
    ! Esta classe representa um padrao de configuracao. Este padrao
    define um conjunto de valores que caracterizacao um objeto
    da classe egConnetionTP. Existira um ou mais padroes de
configuracao
    ja definidos no sistema AXS.
    Obrigatoriamente, um objeto da classe egConnectionTP
    estara associado a um objeto desta classe atraves do atributo
    cfgConnectionTPId.
    !;;
  ATTRIBUTES
    cfgConnectionTPId GET,
    sinalizationType GET-REPLACE,
    cfgStatus GET-REPLACE,
    terminationPointType GET-REPLACE,
    commonChannelSignaling GET-REPLACE,
    connectionTPType GET-REPLACE,
    lineSinalization GET-REPLACE,
    signalPath GET-REPLACE,
    mfcStatus GET-REPLACE,
    timeoutALE GET-REPLACE,
    timeoutMFC GET-REPLACE,
    directionality GET-REPLACE;;;
REGISTERED AS { axsObjectClass 20 };

-- =====
-- operatorPrivilege MANAGED OBJECT CLASS
-- =====

operatorPrivilege MANAGED OBJECT CLASS
DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
  operatorPrivilegePkg PACKAGE
  BEHAVIOUR
  operatorPrivilegePkgBehaviour BEHAVIOUR
  DEFINED AS
    ! Uma instancia desta classe representa um privilegio de
atendimento.
    O privilegio de atendimento define um conjunto de permissoes e
    caracteristicas de atendimento para um determinado operador.
    Todo objeto da classe paOperator esta associado a um objeto
    operatorPrivilege atraves do atributo privilegeNumber .
    Existira ja definido no sistema AXS um ou mais privilegios de
    atendimento.
    !;;
  ATTRIBUTES
    privilegeNumber GET,
    privilegePermissions GET-REPLACE;;;
REGISTERED AS { axsObjectClass 22 };

```

```
-- =====
-- dacService  MANAGED OBJECT CLASS
-- =====
```

```

dacService  MANAGED OBJECT CLASS
  DERIVED FROM axsService;
  CHARACTERIZED BY
  dacServicePkg PACKAGE
  BEHAVIOUR
  dacServicePkgBehaviour BEHAVIOUR
  DEFINED AS
  ! A classe dacService e a classe de objetos gerenciados que
  representa os servicos do tipo DAC - Distribuidor Automatico de
  Chamadas.
  !;;
  ATTRIBUTES
    serviceNumber          GET-REPLACE,
    messages               GET-REPLACE,
    license                 GET;
  ACTIONS
    activatedService,
    deactivatedService,
    deactivatedServiceImmediately;;;
REGISTERED AS { axsObjectClass 23 };

```

```
-- =====
-- PACKAGES
-- =====
```

```

hostConfigPackage PACKAGE
  BEHAVIOUR hostConfigPackageBehaviour BEHAVIOUR
  DEFINED AS " ";
  ATTRIBUTES hostConfig GET-REPLACE;
REGISTERED AS { axsPackage 1};

paAllocationPackage PACKAGE
  BEHAVIOUR paAllocationPackageBehaviour BEHAVIOUR
  DEFINED AS " ";
  ATTRIBUTES paAllocation GET-REPLACE;
REGISTERED AS { axsPackage 2};

paTrainingPackage PACKAGE
  BEHAVIOUR paTrainingPackageBehaviour BEHAVIOUR
  DEFINED AS " ";
  ATTRIBUTES paTraining GET-REPLACE;
REGISTERED AS { axsPackage 3};

workTimePackage PACKAGE
  BEHAVIOUR workTimePackageBehaviour BEHAVIOUR
  DEFINED AS " ";
  ATTRIBUTES workTime GET-REPLACE;
REGISTERED AS { axsPackage 4};

```

```

-- =====
-- ATTRIBUTES
-- =====

copyrighth ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.Copyrighth;
REGISTERED AS { axsAttribute 2 };
-- Este atributo define o copyrighth de uma central AXS

configuration ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.Configuration;
REGISTERED AS { axsAttribute 3 };
-- Este atributo define o numero de estagios e operadores configurados
-- no sistema AXS

identification ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.Identification;
REGISTERED AS { axsAttribute 5 };
-- Este atributo identifica a central, definido o
-- numero de telefone, categoria, tipo, numero de apresentacao e
-- tipo de identificacao

matrixPreference ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.MatrixPreference;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 7 };
-- Este atributo define a preferencia do operador do sistema AXS
-- quanto ao plano da matriz de comutacao que devera ser o ativo
-- Uma das seguintes escolhas pode ser feita:
--     plano A ativo ,
--     plano B ativo,
--     plano A ativo e plano B reserva,
--     plano B ativo e plano A reserva.

-- CHM_ALT_ELC_ATIVO
elcPreference ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.ElcPreference;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 9 };
-- Este atributo define a preferencia do operador do sistema AXS
-- quanto ao elc que devera ser o ativo
-- Uma das seguintes escolhas pode ser feita:
--     elc 0 ativo ,
--     elc 1 ativo,
--     elc 0 ativo e elc 1 reserva,
--     elc 1 ativo e elc 0 reserva

hostname ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.Hostname;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 11 };
-- Este atributo define o hostname de um estagio com sistema operacional
-- Solaris

ecgPreference ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.EcgPreference;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 12};

```

```

-- Este atributo define a preferencia do operador do sistema AXS
-- quanto ao ecg que devera ser o ativo.
-- Uma das seguintes escolhas pode ser feita:
--     ecg 0 ativo ,
--     ecg 1 ativo,
--     ecg 0 ativo e elc 1 reserva,
--     ecg 1 ativo e elc 0 reserva

timeSlotType  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.TimeSlotType;
    MATCHES FOR  EQUALITY;
REGISTERED AS  {axsAttribute 14};
-- define se o time-slot 0 do link possui sincronismo ou e' em modo
-- transparente

internalId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.InternalId;
    MATCHES FOR EQUALITY;
REGISTERED AS  { axsAttribute 15 };
-- Identifica unicamente um objeto

ipAddress  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.IpAddress;
    MATCHES FOR  EQUALITY;
REGISTERED AS  {axsAttribute 16};
-- Este atributo mostra o endereco IP (Internet Protocol) de um estagio

sinalizationType  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.SinalizationType;
    MATCHES FOR  EQUALITY;
REGISTERED AS  {axsAttribute 18};
-- Este atributo define o tipo de sinalizacao de linha

connectionType  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.ConnectionType;
    MATCHES FOR  EQUALITY;
REGISTERED AS  {axsAttribute 20};
-- Define se o tipo de conexao e' canal de servico ou via normal

egTrailTPStatus  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.EgTrailTPStatus;
    MATCHES FOR  EQUALITY;
REGISTERED AS  {axsAttribute 21};
-- Define o estado de um link

interfaceType  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.InterfaceType;
    MATCHES FOR  EQUALITY;
REGISTERED AS  {axsAttribute 22};
-- Define o tipo de interface: HDB3 ou 422

egConnectionTPStatus  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.EgConnectionTPStatus;
    MATCHES FOR  EQUALITY;
REGISTERED AS  {axsAttribute 23};
-- Define o estado de um juntor

```

```
directionality ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.Directionality;
    MATCHES FOR EQUALITY;
REGISTERED AS {axsAttribute 24};
-- Define o sentido de sinalizacao

egTrailTPTType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.EgTrailTPTType;
    MATCHES FOR EQUALITY;
REGISTERED AS {axsAttribute 26};
-- Define se o link e' E1 ou T1

cardType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.CardType;
    MATCHES FOR EQUALITY;
REGISTERED AS {axsAttribute 27};
-- Define o tipo de cartao do EG (dmfs ou digr)

numberOfDTMFfilter ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.NumberOfDTMFfilter;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 29 };
-- Mostra o numero de filtros DTMF

filterType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.FilterType;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 31 };
-- Define o tipo de filtro utilizado

masterProcessor ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.MasterProcessor;
    MATCHES FOR EQUALITY;
REGISTERED AS {axsAttribute 32};
-- Mostra se o processador principal esta ativo ou nao

processorStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.ProcessorStatus;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 33 };
-- Mostra o estado do processador

codeStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.CodeStatus;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 34 };
-- Mostra o estado de um estagio.Pode ser:
-- a) deactive: o estagio esta nao operacional;
-- b) loadingCode: o estagio esta em um estado transitorio, fazendo a
-- carga de software. Neste estado o estagio esta nao operacional;
-- c) withCode : o estagio esta operacional;
-- d) defect : nao operacional por motivo de defeito.

filteringType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.FilteringType;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 36 };

-- Define o tipo de filtro
```

```

finalSelection ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.FinalSelection;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 39 };
-- Define diversas caracteristicas da chamada, tais como:
-- a) ToneInsertion : seleciona a forma como sera inserido tom ao
--     se finalizar a troca MFC;
-- b) ToneFrequency : seleciona a frequencia do tom a ser inserido caso
--     tenha-se optado pela insercao do tom;
-- c) Cadence: permite a escolha de qual cadencia, pre-programada na
matriz
--     de comutacao, sera usada para o tom a ser inserido, caso tenha-se
optado
--     pela insercao do tom. Permite tambem determinar quantas vezes a
cadencia
--     escolhida sera repetida;
-- d) SourceSignal: indica se a chamada deve ser tarifada
--     pela origem ou nao. Com assinante livre significa que informacoes
desta
--     natureza (assinante livre, ocupado, nao atende, etc) serao
repassadas
--     pela origem. Com retencao por B permite que o AXS retenha a
chamada
--     mesmo que o assinante A tenha desligado.

answerSignal ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.AnswerSignal;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 42 };
-- Determina a forma como serao atendidas as chamadas entrantes (Tipo de
-- atendimento do juntor. Pode ser:
-- a) withoutAnswering : o sistema AXS considera "chamada atendida" sem
haver
--     enviado atendimento para a origem;
-- b) withAnswering : o sistema AXS considera "chamada atendida", e
provoca o
--     envio de atendimento para a origem
-- c) doubleAnswering : envio de duplo atendimento para a origem,
causando a
--     liberacao de chamadas entrantes DDC.

destination ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.Destination;
    MATCHES FOR EQUALITY;
REGISTERED AS {axsAttribute 43};
-- Define o equipamento conectado a outra ponta do link

paType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.PaType;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 47};
-- Mostra o tipo da PA. Pode ser PA digital ou PA discada.

paNumber ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.PaNumber;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 48};
-- Define o endereco fisico da PA no ELD. Quando paNumber for igual a
0xffff,
-- significa PA sem console

```

```
paLogicNumber  ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.PaLogicNumber;
                MATCHES FOR      EQUALITY;
REGISTERED AS   {axsAttribute 49};
-- Atributo que identifica unicamente uma PA

consoleType    ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.ConsoleType;
                MATCHES FOR      EQUALITY;
REGISTERED AS   {axsAttribute 50};
-- Define o tipo de console

associatedGroup ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.AssociatedGroup ;
                MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS   { axsAttribute 51};
-- Define os grupos a qual um operador pertence

paBasicState   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.PaBasicState;
                MATCHES FOR      EQUALITY ;
REGISTERED AS   {axsAttribute 52};
-- Mostra o estado da PA

paDetailState  ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.PaDetailState;
                MATCHES FOR      EQUALITY ;
REGISTERED AS   {axsAttribute 53};
-- Mostra o estado detalhado da PA

paQueue        ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.PaQueue;
                MATCHES FOR      EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
REGISTERED AS   {axsAttribute 54};
-- Mostra os elementos que estao na fila de um determinado grupo

groupID        ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.GroupID;
                MATCHES FOR EQUALITY;
REGISTERED AS   { axsAttribute 55};
-- Mostra o numero de identificacao do grupo

groupType      ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.GroupType;
                MATCHES FOR EQUALITY;
REGISTERED AS   { axsAttribute 56};
-- Define se o grupo de operadores e' do tipo normal ou transbordo

groupElemNumber ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  AXSMODULEASN1.GroupElemNumber;
                MATCHES FOR EQUALITY;
REGISTERED AS   { axsAttribute 57};
-- Define o numero maximo de elementos que podem estar na fila
-- de um grupo
```



```

groupQueue ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.GroupQueue;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
REGISTERED AS { axsAttribute 58};
-- Mostra os elementos que estao na fila de um grupo

groupQueueSize ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.GroupQueueSize;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 59};
-- Mostra o tamanho da fila de um grupo

overflowCondition ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.OverflowCondition;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 60};
-- Determina o tamanho maximo da fila de um grupo, onde acima deste
-- tamanho as chamadas serao transferidos para um grupo de transbordo

overflowTimeCondition ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.OverflowTimeCondition;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 61};
-- Determina o tempo maximo que uma chamada pode esperar na fila, onde
-- ultrapassando este limite de tempo as chamadas serao transferidos
-- para um grupo de transbordo

groupOperators ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.GroupOperators ;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
REGISTERED AS { axsAttribute 62};
-- Determina as instacias da classe paOperator que pertencem
-- a um determinado grupo. Pode ser zero ou mais.

overflowedGroup ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.OverflowedGroup ;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
REGISTERED AS { axsAttribute 63};
-- Identifica os grupos que transferiram (transbordaram) suas chamadas
para
-- um determinado grupo. Pode ser zero ou mais.

overflowingGroup ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.OverflowingGroup;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS { axsAttribute 64};
-- Identifica os grupos para onde um determinado grupo ira
--transferir (transbordar) suas chamadas. Pode ser zero ou mais

operatorName ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.OperatorName ;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 65};
-- Define o nome do operador

operatorPassword ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.OperatorPassword ;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 66};
-- Determina a senha do operador

```

```
privilegePermissions ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.PrivilegePermissions ;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS { axsAttribute 67};
-- Define um conjunto de permissões e características de atendimento

attendingPriority ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.AttendingPriority ;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS { axsAttribute 68};
-- Define a prioridade de atendimento de um operador no atendimento
-- de chamadas direcionadas a um grupo em que o mesmo trabalha.

paComplementState ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.PaComplementState;
    MATCHES FOR EQUALITY ;
REGISTERED AS {axsAttribute 69};
-- Define os estados complementares da PA

serviceTPstatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.ServiceTPstatus;
    MATCHES FOR EQUALITY ;
REGISTERED AS {axsAttribute 70};
-- Define o número de troncos entrada/saída disponíveis/ocupados

actualStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.ActualStatus;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 71 };
-- Define se um estágio é o ativo ou reserva, no caso de existir
dualidade

checkTone ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.CheckTone;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 72 };
-- Check-ton é um procedimento interno a matriz de comutação que
verifica
-- a validade de uma comutação bidirecional. Este atributo determina qual
-- percentagem do valor de comutações bidirecionais será verificada
através
-- do procedimento de check-ton. Pode assumir os valores de 0 a 100.

waitTime ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.WaitTime;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 73 };
-- Define o intervalo de tempo(segundos) que é feito a conferência dos
arquivos
-- de configuração entre ECG reserva e o ativo. Define também o tempo
(segundos)
-- para providências emergenciais quando na falha dos processos de
dualidade.

workTime ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.WorkTime;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 74 };
-- Define o horário de atendimento de um serviço.
```

```

hostConfig ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.HostConfig;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 75 };
-- Determina se deve-se ou nao periodicamente verificar a conexao
-- com o mainframe. Caso positivo, define o intervalo de tempo em
-- segundos da verificacao.

serviceStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.ServiceStatus;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 76 };
-- Define o estado do servico: se o mainframe esta ativo ou nao, se o
servico
-- esta ativo, numero de instancias ocupadas e numero maximo de
instancias.

cfgStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.CfgStatus;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 78 };
-- Determina se um objeto da classe egTrailTP ou egConnectionTP esta ou
nao
-- configurado

terminationPointType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.TerminationPointType;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 79 };
-- Define se a conexao e' interna (dentro da propria central, conectando
-- dois estagios), ou externa

commonChannelSignaling ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.CommonChannelSignaling;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 80 };
-- Define a existencia ou nao de sinalizacao por canal comum

connectionTPType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.ConnectionTPType;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 81 };
-- Define se a conexao e' canal de servico ou via normal

lineSinalization ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.LineSinalization;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 90 };
-- Define a existencia ou nao de sinalizacao de linha

signalPath ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.SignalPath;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 91 };
-- Define o tipo da sinalizacao

mfcStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.MFCStatus;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 93 };
-- Define se a sinalizacao MFC esta ativa ou nao

```

```
timeoutALE ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.TimeoutALE;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 94 };
-- Define o tempo de espera para analise da linha estendida (ALE)

timeoutMFC ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASNI.TimeoutMFC;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 95 };
-- Define o tempo de espera para analise da sinalizacao MFC

associatedService ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.AssociatedService;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 96 };
-- Define se existe um servico associado a um determinado ponto final
-- de uma conexao. Caso positivo, determina o servico associado

cfgConnectionTPId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.InternalId;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 97 };
-- Identifica unicamente um objeto da classe cfgConnectionTP

sourceName ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.InternalId;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 98 };
-- Identifica a origem de um determinado servico

baseSource ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.BaseSource;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 99 };
-- Define o numero da origem de um determinado servico

alternativeSource ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.AlternativeSource;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 100 };
-- Define o numero da origem alternativa de um determinado servico

privilegeNumber ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSMODULEASN1.IntegerNumber;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 101 };
-- Identifica unicamente um objeto da classe privilegeNumber
```

```
performance ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.Performance;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 102};
-- Mostra valores de desempenho da central AXS. As seguintes informacoes
-- sao definidas :
-- valor medio maximo de 10 amostras de 5 seg
-- valor maximo em 5 seg
-- numero de chamadas entrantes
-- numero de chamadas saintes
-- numero de chamadas que nao foram atendidas por timeout
-- numero de chamadas cujo servico solicitado nao existia
-- numero de chamadas derrubadas por A4 no EG
-- numero de chamadas com falha no EG
-- numero de chamadas cujo servico requisitado estava desativado
-- numero total de chamadas
-- numero de chamadas liberadas aguardando 1. cifra
-- numero de chamadas liberadas aguardando 2. cifra
-- numero de chamadas liberadas aguardando 3. cifra
-- numero de chamadas liberadas aguardando 4. cifra
-- numero de chamadas liberadas aguardando 2.e 3. cifra
-- numero de chamadas que completaram troca MFC

description ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.Description;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 103};

paAllocation ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.PaAllocation;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 104};

paTraining ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.PaAllocation;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 105};

internalKey ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.IntegerNumber;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 106};

serviceNumber ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.ServiceNumber;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 107};

license ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.ServiceLicense;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 109};

messages ATTRIBUTE
    WITH ATTRIBUTE SYNTAX AXSModuleASN1.ServiceMessages;
    MATCHES FOR EQUALITY;
REGISTERED AS { axsAttribute 110};
```

```

-- =====
-- ACTION
-- =====

codeTransfer ACTION
    BEHAVIOUR codeTransferBehaviour;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX AXSMODULEASN1.CodeTransferInfo;
    WITH REPLY SYNTAX AXSMODULEASN1.CodeTransferResult;
REGISTERED AS { axsAction 1 };

codeTransferBehaviour BEHAVIOUR DEFINED AS
! Esta acao faz a regarga de software de um ECS,EG,ERA ou ELD!;

egReset ACTION
    BEHAVIOUR egResetBehaviour;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX AXSMODULEASN1.EgResetInfo;
    WITH REPLY SYNTAX AXSMODULEASN1.EgResetResult;
REGISTERED AS { axsAction 2 };

egResetBehaviour BEHAVIOUR DEFINED AS
! Esta acao reinicia um EG !;

deactivatedService ACTION
    BEHAVIOUR deactivatedServiceBehaviour;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX AXSMODULEASN1.DeactivatedServiceInfo;
    WITH REPLY SYNTAX AXSMODULEASN1.DeactivatedServiceResult;
REGISTERED AS { axsAction 3 };

deactivatedServiceBehaviour BEHAVIOUR DEFINED AS
! A acao deactivatedService desativa um servico. Se o servico esta
atendendo
    chamadas no momento da execucao da acao, o servico e' desativado
somentemente apos
    terminar o atendimento das chamadas !;

activatedService ACTION
    BEHAVIOUR activatedServiceBehaviour;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX AXSMODULEASN1.ActivatedServiceInfo;
    WITH REPLY SYNTAX AXSMODULEASN1.ActivatedServiceResult;
REGISTERED AS { axsAction 4 };

activatedServiceBehaviour BEHAVIOUR DEFINED AS
! A acao activatedService ativa um servico !;

deactivatedServiceImmediately ACTION
    BEHAVIOUR deactivatedServiceImmediatelyBehaviour;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX
AXSMODULEASN1.DeactivatedServiceImmediatelyInfo;
    WITH REPLY SYNTAX AXSMODULEASN1.DeactivatedServiceImmediatelyResult;
REGISTERED AS { axsAction 5 };

deactivatedServiceImmediatelyBehaviour BEHAVIOUR DEFINED AS
! A acao deactivatedServiceImmediately desativa um servico no momento
em que a acao e' executada, independente se o servico estava ou nao
atendendo chamadas !;

```

```

-- =====
-- NAME BINDING
-- =====

ecg-axs NAME BINDING
  SUBORDINATE OBJECT CLASS ecg;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "Rec. M.3100 : 1992":equipmentId;
REGISTERED AS { axsNameBinding 2 };

ecs-axs NAME BINDING
  SUBORDINATE OBJECT CLASS ecs;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :equipmentId;
REGISTERED AS { axsNameBinding 3 };

eg-axs NAME BINDING
  SUBORDINATE OBJECT CLASS eg;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :equipmentId;
REGISTERED AS { axsNameBinding 4 };

eld-axs NAME BINDING
  SUBORDINATE OBJECT CLASS eld;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :equipmentId;
REGISTERED AS { axsNameBinding 5 };

elc-axs NAME BINDING
  SUBORDINATE OBJECT CLASS elc;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :equipmentId;
REGISTERED AS { axsNameBinding 6 };

era-axs NAME BINDING
  SUBORDINATE OBJECT CLASS era;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :equipmentId;
REGISTERED AS { axsNameBinding 7 };

egConnectionTP-egTrailTP NAME BINDING
  SUBORDINATE OBJECT CLASS egConnectionTP;
  NAMED BY SUPERIOR OBJECT CLASS egTrailTP;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :cTPId;
REGISTERED AS { axsNameBinding 8 };

dfid-eg NAME BINDING
  SUBORDINATE OBJECT CLASS dfid;
  NAMED BY SUPERIOR OBJECT CLASS eg;
  WITH ATTRIBUTE "Rec. M.3100 : 1992":equipmentId;
REGISTERED AS { axsNameBinding 11 };

pa-axs NAME BINDING
  SUBORDINATE OBJECT CLASS pa;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :equipmentId;
REGISTERED AS { axsNameBinding 12 };

```

```
paOperatorGroup-axs NAME BINDING
  SUBORDINATE OBJECT CLASS paOperatorGroup;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE groupID;
  CREATE ;
  DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { axsNameBinding 13 };

paOperator-axs NAME BINDING
  SUBORDINATE OBJECT CLASS paOperator;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE operatorName;
  CREATE ;
  DELETE ;
REGISTERED AS { axsNameBinding 14 };

digr-eg NAME BINDING
  SUBORDINATE OBJECT CLASS digr;
  NAMED BY SUPERIOR OBJECT CLASS eg;
  WITH ATTRIBUTE "Rec. M.3100 : 1992":equipmentID;
REGISTERED AS { axsNameBinding 15};

axsService-axs NAME BINDING
  SUBORDINATE OBJECT CLASS axsService;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "OP1 Library Vol. 4":serviceID;
REGISTERED AS { axsNameBinding 16 };

egTrailTP-eg NAME BINDING
  SUBORDINATE OBJECT CLASS egTrailTP;
  NAMED BY SUPERIOR OBJECT CLASS eg;
  WITH ATTRIBUTE "Rec. M.3100 : 1992" :tTPID;
REGISTERED AS { axsNameBinding 18 };

cfgConnectionTP-axs NAME BINDING
  SUBORDINATE OBJECT CLASS cfgConnectionTP;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE cfgConnectionTPID;
REGISTERED AS { axsNameBinding 20 };

levelSource-axsService NAME BINDING
  SUBORDINATE OBJECT CLASS levelSource;
  NAMED BY SUPERIOR OBJECT CLASS axsService;
  WITH ATTRIBUTE sourceName;
  CREATE;
  DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { axsNameBinding 21 };

operatorPrivilege-axs NAME BINDING
  SUBORDINATE OBJECT CLASS operatorPrivilege;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE privilegeNumber;
REGISTERED AS { axsNameBinding 22 };

dacService-axs NAME BINDING
  SUBORDINATE OBJECT CLASS axsService;
  NAMED BY SUPERIOR OBJECT CLASS axs;
  WITH ATTRIBUTE "OP1 Library Vol. 4":serviceID;
REGISTERED AS { axsNameBinding 23 };
END
```


ANEXO B: DESCRIÇÃO DO MODELO DE INFORMAÇÃO ASN.1

```
AXSModuleASN1 {iso(1) member-body(2) br(76) type-org(2) digitro(4880)
axs(1)
```

```
    informationModule(0) asn1Module(0) 1}
```

```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

```
IMPORTS ObjectInstance FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1)
modules(0) protocol(3)};
```

```
-- OIDs for AXS Project
```

```
axsInformationModel OBJECT IDENTIFIER ::=
    {iso(1) member-body(2) br(76) type-org(2)
    digitro(4880) axs(1) informationModule(0)}
axsAsn1Module OBJECT IDENTIFIER ::=
    {axsInformationModel asn1Module(0)}
axsObjectClass OBJECT IDENTIFIER ::=
    {axsInformationModel objectClass(1) }
axsPackage OBJECT IDENTIFIER ::=
    {axsInformationModel package(2)}
axsNameBinding OBJECT IDENTIFIER ::=
    {axsInformationModel nameBinding(3)}
axsAttribute OBJECT IDENTIFIER ::=
    {axsInformationModel attribute(4)}
axsAttributeGroup OBJECT IDENTIFIER ::=
    {axsInformationModel attributeGroup(5)}
axsParameter OBJECT IDENTIFIER ::=
    {axsInformationModel parameter(6) }
axsAction OBJECT IDENTIFIER ::=
    {axsInformationModel action(7)}
axsNotification OBJECT IDENTIFIER ::=
    {axsInformationModel notification(8)}
```

```
ActualStatus ::= ENUMERATED {
    active (0),
    backup (1),
    undefined (3)
}
```

```
AnswerSignal ::= ENUMERATED {
    withAnswering (0),
    withoutAnswering (1),
    doubleAnswering (2)
}
```

```
AttendingPriority ::= INTEGER
```

```
AssociatedGroup ::= SET OF GroupID
```

```
AssociatedService ::= SEQUENCE {
    status ENUMERATED { with-service (0), without-service (1) },
    serviceId INTEGER
}
```

```
AlternativeSource ::= CHOICE {
    none NULL,
    alternativeSource INTEGER
}

BaseSource ::= INTEGER

Copyrighth ::= GraphicString

CommonChannelSignaling ::= ENUMERATED {
    active (0),
    deactivate (1)
}

Configuration ::= SEQUENCE {
    numberOfFEG INTEGER,
    numberOfECS INTEGER,
    numberOfELD INTEGER,
    numberOfERA INTEGER,
    numberOfPA INTEGER,
    maxNumberOfAttendent INTEGER,
    numberOfAttendent INTEGER
}

Category ::= ENUMERATED {
    equipamentoComunicacao (0),
    chamadorDesconhecido (1),
    telefonista (2),
    assinanteComum (3),
    reservaAssinanteComPrioridade (4),
    equipamentoManutencao (5),
    telefonePublicoLocal (6),
    telefonePublicoInter (7),
    assinanteComFacilidadeTransf (8),
    assinanteComTarifEspecial (9)
}

CardType ::= ENUMERATED {
    dmfs (0),
    digr (1)
}

ConnectionType ::= ENUMERATED {
    serviceChannel (0),
    normalChannel (1)
}

CfgConnectionTPIId ::= INTEGER

Cadence ::= SEQUENCE {
    cadenceId INTEGER,
    repetitions ENUMERATED { perpetual (0), programmed (1) },
    numberOfRepetitions INTEGER
}

CfgStatus ::= ENUMERATED {
    configured (0),
    not-configured (1)
}
```

```

CmdStatus ::= ENUMERATED {
    accept (0),
    reject (1)
}

ConnectionTPTType ::= ENUMERATED {
    analog (0),
    digital (1)
}

CodeStatus ::= ENUMERATED {
    deactivate (0),
    loadingCode (1),
    withCode (2),
    defect (3)
}

ConsoleType ::= CHOICE {
    pcApplicationAddress INTEGER,
    terminalType PaTerminalType
}

CheckTone ::= INTEGER (0..100)

DfidProcessorStatus ::= SET OF SEQUENCE {
    processorId INTEGER,
    status ENUMERATED{ absent(0), present(1)}
}

DialPa ::= CHOICE {
    paNotMCP30 PaNotMCP30,
    paMCP30 PaMCP30
}

DspStatus ::= ENUMERATED {
    absent (0),
    active (1),
    defect (2)
}

Destination ::= ENUMERATED {
    publicSwitching (0),
    gateway (1) ,
    era (2),
    ecs (3),
    remotePa (4),
    dpcr (5)
}

DigitalPa ::= SEQUENCE {
    dpad INTEGER,
    switchCircuit INTEGER
}

DaysCfg ::= SEQUENCE {
    day INTEGER,
    type ENUMERATED { holiday(0), sunday(1), saturday(2),
specialDay(3)}
}

```

```

DaysOfMonth ::= SEQUENCE {
    month INTEGER,
    daysCfg SET OF DaysCfg
}

Description ::= GraphicString

EgConnectionTPNumber ::= INTEGER

EgTrailTPStatus ::= ENUMERATED {
    active (0),
    deactivate (1)
}

ElcPreference ::= ENUMERATED {
    elc0 (0),
    elc1 (1),
    elc0-elc1 (2),
    elc1-elc0 (3)
}

ElcPreferenceUsed ::= ENUMERATED {
    elc0 (0),
    elc1 (1),
    undefined (2)
}

EcgPreference ::= ENUMERATED {
    ecg0 (0),
    ecg1 (1),
    ecg0-ecg1 (2),
    ecg1-ecg0 (3)
}

EcgPreferenceUsed ::= ENUMERATED {
    ecg0 (0),
    ecg1 (1),
    undefined (2)
}

Directionality ::= CHOICE {
    notApplicable NULL,
    directionality ENUMERATED {inbound(0), outbound(1),
bidirectional(2) }
}

EgConnectionTPStatus ::= ENUMERATED {
    idle (0),
    allocated (1),
    busy-without-confirmation (2),
    busy-with-confirmation (3),
    busy (4),
    attended (5),
    subscriber-A-disconnected (6),
    subscriber-B-disconnected (7),
    locked (8),
    fail (9),
    waiting-disconnect-confirmation-A-B (10),
    processor-fail (11),
    link-fail (12),
    indeterminate (13)
}

```

```

EgTrailTPTType ::= ENUMERATED {
    trail-E1 (0),
    trail-T1 (1)
}

FilterType ::= SEQUENCE {
    dtmf-receptor ENUMERATED{ disable(0), enable(1) },
    dtmf-generator ENUMERATED{ disable(0), enable(1) },
    hz425-receptor ENUMERATED{ disable(0), enable(1) },
    hz1600-receptor ENUMERATED{ disable(0), enable(1) },
    fax-receptor ENUMERATED{ disable(0), enable(1) }
}

FilteringType ::= ENUMERATED {
    dtmf (0),
    mfc-ale (1),
    mix-audio (2),
    others (3),
    general-filter (4)
}

FinalSelection ::= SEQUENCE {
    toneInsertion ToneInsertion ,
    toneFrequency ToneFrequency,
    cadence Cadence ,
    sourceSignal SourceSignal
}

GroupID ::= INTEGER

GroupType ::= ENUMERATED {
    normal(0),
    overflow(1)
}

GroupElemNumber ::= INTEGER

GroupOperators ::= SET OF INTEGER

GroupQueue ::= SET OF INTEGER

GroupQueueSize ::= INTEGER

Hostname ::= GraphicString

HostConfig ::= SEQUENCE {
    periodicalActivation ENUMERATED { yes (0) , no (1) },
    periodTime INTEGER (0..65535)
}

IpAddress ::= GraphicString

InternalId ::= INTEGER

IntegerNumber ::= INTEGER

```

```
InterfaceType ::= ENUMERATED {
    type-HDB3 (0),
    type-422 (1)
}

Identification ::= SEQUENCE {
    phoneNumber PhoneNumber,
    category Category,
    numberType NumberType,
    numberApresentation NumberApresentation,
    identificationType IdentificationType
}

IdentificationType ::= ENUMERATED {
    with-incomplete-identify (0),
    without-incomplete-identify (1)
}

LineSinalization ::= ENUMERATED {
    enable (0),
    disable (1)
}

License ::= GraphicString

MatrixPreference ::= ENUMERATED {
    planeA (0),
    planeB (1),
    planeA-B (2),
    planeB-A (3)
}

MatrixPreferenceUsed ::= ENUMERATED {
    planeA (0),
    planeB (1),
    undefined (2)
}

MasterProcessor ::= ENUMERATED {
    enable (0),
    disable (1)
}

MFCStatus ::= ENUMERATED {
    enable (0),
    disable (1)
}

NumberType ::= ENUMERATED {
    local (0),
    national (1),
    international (2)
}

NumberApresentation ::= ENUMERATED {
    restricted(0),
    not-restricted(1)
}
```

```
NumberOfDTMFfilter ::= ENUMERATED {
    n24 (0),
    n30 (1)
}

OverflowCondition ::= INTEGER

OverflowTimeCondition ::= INTEGER

OverflowedGroup ::= SET OF INTEGER

OverflowingGroup ::= SET OF INTEGER

OperatorName ::= GraphicString

OperatorPassword ::= GraphicString

PaType ::= CHOICE{
    digitalPa DigitalPa,
    dialPa DialPa
}

PaNotMCP30 ::= PaIdentifier

PaIdentifier ::= PhoneNumber

PaMCP30 ::= EgConnectionTPNumber

PaNumber ::= INTEGER

PaLogicNumber ::= INTEGER

PaTerminalType ::= ENUMERATED {
    noneConsole(0),
    hostTerminal(1)
}

PaComplementState ::= ENUMERATED {
    undefined (0),
    inMonitoring(1),
    inLogoutProcess(2),
    blocked(4),
    autoblocked(8)
}
```

```

PaDetailState ::= ENUMERATED {
    defect(0),
    active(1),
    idle(2),
    waitPa(3),
    waitTp(4),
    waitRespInterc(5),
    wait(6),
    attending(7),
    diffuse(8),
    receivingDiffusedMsg(9),
    interpolateProcess(10),
    interpolating(11),
    attendingOtherPa(12),
    attendingTp(13),
    waitDischargeAttendingAudio(14),
    waitDischargeDisattendingAudio(15),
    waitDischargeConferenceAudio(16),
    waitDischargeTransferAudio(17),
    waitDischargeConsultAudio(18),
    inCalling(19),
    monitoringOtherPa(20),
    waitResponseForConference(21),
    waitResponseForInterpolate(22),
    waittingCircConference(23),
    monitoringTp(24),
    commandingConference(25),
    paccWaitting(26),
    paccJpaWatting(27),
    paccAlone(28),
    inConference(29),
    waitRespPInterpolate(30),
    waitResponseRecord(31),
    waitRecordMixer(32),
    waitRespCallGenerate(33),
    waitCallResult(34),
    elaCapture(35),
    elaCaptureResponse(36),
    waitTpSvsTransferResponse(37),
    waitTpGrSTransferResponse(38),
    waitTpTransferGRGR01(39),
    waitTpTransferGRGR02(40),
    waitTpTransferGRGR03(41),
    waitTpTransferGRGR04(42),
    waitTpTransferGRGR05(43),
    waitTpTransferGRGR06(44),
    waitTpTransferGRGR07(45),
    waitTpTransferGRGR08(46),
    waitTpTransferGRGR09(47),
    waitTransferResponseTpGRS(48),
    waitTransferResponseTpSvs(49),
    elaTransferPa(50)
}

```



```

PaBasicState ::= ENUMERATED {
    defect(0),
    inactive(1),
    idle(2),
    busy(3),
    blocked(4),
    autoblocked(5),
    undefined (6)
}

PaQueue ::= SET OF PaQueueElement

PaQueueElement ::= INTEGER

PhoneNumber ::= GraphicString

Performance ::= SEQUENCE {
    mediaMax INTEGER, -- valor medio maximo de 10 amostras de 5 seg
    picoMax  INTEGER, -- valor maximo em 5 seg
    chmEnt   INTEGER, -- n. chamadas entrantes
    chmSai   INTEGER, -- n. chamadas saintes
    chmTimeout INTEGER, -- n. chamadas EG informou timeout
    chmSemServ INTEGER, -- n. chamadas nao encontrou servico
    chmDerrA4 INTEGER, -- n. chamadas derrubadas por A4 no EG
    chmFalha INTEGER, -- n. chamadas com falha no EG
    chmServDesat INTEGER, -- n.chamadas com serv. desativo
    chmEncamSv INTEGER, -- n. chamadas encaminhadas ao Trata Servico
    chmLib1Cif INTEGER, -- n cham. liberadas aguardando 1. cifra
    chmLib2Cif INTEGER, -- n cham. liberadas aguardando 2. cifra
    chmLib3Cif INTEGER, -- n cham. liberadas aguardando 3. cifra
    chmLib4Cif INTEGER, -- n cham. liberadas aguardando 4. cifra
    cifra
    chmLib2e3Cif INTEGER, -- n cham. liberadas aguardando 2.e 3.
}

cifra
    chmComplTroca INTEGER -- n cham. completaram troca MFC
}

ProcessorStatus ::= SEQUENCE {
    master DspStatus,
    sinalizationLine DspStatus,
    internalCheckTom DspStatus,
    externalCheckTom DspStatus
}

PaAllocation ::= SEQUENCE {
    paAllocated PaAllocated,
    paGroupAllocated PaGroupAllocated,
    paAllocationPriority INTEGER (0..255)
}

PaAllocated ::= CHOICE {
    notAllocate NULL,
    pa INTEGER
}

PaGroupAllocated ::= CHOICE {
    notAllocate NULL,
    paGroup SET OF INTEGER
}

```

```

PrivilegePermissions ::= SEQUENCE {
    cfg0 Cfg0,
    cfg1 Cfg1,
    cfg2 Cfg2,
    cfg3 Cfg3,
    cfg4 Cfg4,
    cfg5 Cfg5,
    cfg6 Cfg6,
    cfg7 Cfg7
}

SinalizationType ::= ENUMERATED {
    em-pulsed (0),
    r2d (1),
    em-continuos (2),
    remote-exchange (3),
    em-variant-pulsed (4),
    vague (5)
}

ServiceName ::= GraphicString

ServiceNumber ::= GraphicString

SignalPath ::= ENUMERATED {
    none (0),
    a (1),
    b (2),
    a-b (3)
}

ServiceConfiguration ::= SEQUENCE {
    maxNumberOfInputTP INTEGER,
    maxNumberOfOutputTP INTEGER,
    timeout-ALE INTEGER,
    timeout-MFC INTEGER,
    numberDial GraphicString, --cifras
    serviceInitialTime INTEGER,
    serviceFinalTime INTEGER
}

ServiceTPstatus ::= SEQUENCE {
    numberOfInputTPbusy INTEGER,
    numberOfInputTPfree INTEGER,
    numberOfOutputTPbusy INTEGER,
    numberOfOutputTPfree INTEGER
}

SourceSignal ::= SEQUENCE {
    withBilling      ENUMERATED { disable (0), enable (1) },
    freeSubscriber  ENUMERATED { disable (0), enable (1) },
    b-retention     ENUMERATED { disable (0), enable (1) }
}

ServiceMessages ::= SEQUENCE {
    initial INTEGER,
    serviceOff INTEGER,
    serviceOutOfTime INTEGER
}

```

```

ServiceLicense ::= SEQUENCE {
    expirationDate GraphicString,
    numberOfLicensedServices INTEGER,
    numberOfLicensedHosts INTEGER,
    licensePassword GraphicString
}

ServiceStatus ::= SEQUENCE {
    hostState ENUMERATED { active(0) ,deactive(1), indeterminated (2)
},
    serviceState ENUMERATED { active(0) ,backup(1), indeterminated
(2)
},
    numberOfBusyIstances INTEGER,
    numberOfInstances INTEGER
}

TimeSlotType ::= ENUMERATED {
    sincInformation (0),
    transparent (1)
}

ToneInsertion ::= ENUMERATED {
    notInsert (0),      -- nenhum tom sera inserido
    withSilence (1),   -- o tom sera inserido e o audio presente
    removido
    withSignal (2)     -- o tom sera inserido em cima do audio
    presente
}

ToneFrequency ::= ENUMERATED {
    f425Hz (0),
    f1Khz (1),
    f1600Hz (2),
    none (3)
}

TerminationPointType ::= ENUMERATED {
    internalTerminationPoint (0),
    connectionTerminatinPoint (1)
}

TimeoutALE ::= INTEGER

TimeoutMFC ::= INTEGER

WaitTime ::= SEQUENCE {
    emergencyCheck INTEGER ,
    fileCheck      INTEGER
}

WorkTimeConfig ::= SEQUENCE {
    inicialWorkTime SEQUENCE { hour INTEGER, min INTEGER},
    finalWorkTime   SEQUENCE { hour INTEGER, min INTEGER}
}

```

```

WorkTime ::= SEQUENCE {
    weekdays      WorkTimeConfig,
    sunday        WorkTimeConfig,
    saturday      WorkTimeConfig,
    holiday       WorkTimeConfig,
    specialDay    WorkTimeConfig,
    daysOfMonth   SET OF DaysOfMonth
}

Cfg0 ::= SEQUENCE {
    transferTpToGroup      ENUMERATED{ disable(0), enable(1) },
    canAutoblock           ENUMERATED{ disable(0), enable(1) },
    canBlockOthersPas      ENUMERATED{ disable(0), enable(1) },
    canPutTpInWait        ENUMERATED{ disable(0), enable(1) },
    canPutPaInWait        ENUMERATED{ disable(0), enable(1) },
    canBePutInWait        ENUMERATED{ disable(0), enable(1) },
    automaticAttending    ENUMERATED{ disable(0), enable(1) },
    canChangeAttendingMode ENUMERATED{ disable(0), enable(1) }
}

Cfg1 ::= SEQUENCE {
    canDiffuseMsg          ENUMERATED{ disable(0), enable(1) },
    canReceiveDiffusedMsg ENUMERATED{ disable(0), enable(1) },
    canAbandonMsg          ENUMERATED{ disable(0), enable(1) },
    withDisplayOnLine      ENUMERATED{ disable(0), enable(1) },
    canTariff              ENUMERATED{ disable(0), enable(1) },
    canCalling             ENUMERATED{ disable(0), enable(1) },
    canChangeConfigProfile ENUMERATED{ disable(0), enable(1) }
}

Cfg2 ::= SEQUENCE {
    canMonitoredTp        ENUMERATED{ disable(0), enable(1) },
    paCanBeMonitored      ENUMERATED{ disable(0), enable(1) },
    canInitConference     ENUMERATED{ disable(0), enable(1) },
    canEnterConference    ENUMERATED{ disable(0), enable(1) },
    canDoUrgentConsult    ENUMERATED{ disable(0), enable(1) },
    canAnswerUrgentConsult ENUMERATED{ disable(0), enable(1) },
    canCapturePa          ENUMERATED{ disable(0), enable(1) },
    canCaptureTP          ENUMERATED{ disable(0), enable(1) }
}

Cfg3 ::= SEQUENCE {
    paCanBeCaptured      ENUMERATED{ disable(0), enable(1) },
    tpCanBeCaptured      ENUMERATED{ disable(0), enable(1) },
    canConsult            ENUMERATED{ disable(0), enable(1) },
    canTransferPa         ENUMERATED{ disable(0), enable(1) },
    canTransferTp         ENUMERATED{ disable(0), enable(1) },
    canBeTransferToOtherPa ENUMERATED{ disable(0), enable(1) }
}

```

```

Cfg4 ::= SEQUENCE {
    CanTransferTpToPaQueue    ENUMERATED{ disable(0), enable(1) },
    cantransferPaToPaQueue    ENUMERATED{ disable(0), enable(1) },
    canAcceptTpTransfer       ENUMERATED{ disable(0), enable(1) },
    canAcceptPaTransfer       ENUMERATED{ disable(0), enable(1) },
    canTransferTpOfGrpQueueToPa ENUMERATED{ disable(0), enable(1)},
    canTransferTpOfGroupQueueToGrp ENUMERATED{ disable(0),enable(1)},
    canTransferTpInAttendToOtherGrp ENUMERATED{disable(0),enable(1)},
    canRotateQueue            ENUMERATED{ disable(0), enable(1) }
}

Cfg5 ::= SEQUENCE {
    canRecordAudioOfOtherPa    ENUMERATED{ disable(0), enable(1) },
    canSufferRecord            ENUMERATED{ disable(0), enable(1) },
    receiveTpTransferredByDAC  ENUMERATED{ disable(0), enable(1) },
    haveAttendingBip          ENUMERATED{ disable(0), enable(1) },
    canTransferTpAttended      ENUMERATED{ disable(0), enable(1) },
    isBasePa                   ENUMERATED{ disable(0), enable(1) },
    transferCallToBasePa      ENUMERATED{ disable(0), enable(1) }
}

Cfg6 ::= SEQUENCE {
    consultationBasePa         ENUMERATED{ disable(0), enable(1) },
    receiveCallOfBasePa       ENUMERATED{ disable(0), enable(1) },
    receiveConsultOfBasePa    ENUMERATED{ disable(0), enable(1) },
    enterBlockedInLogin       ENUMERATED{ disable(0), enable(1) },
    canBeBlocked              ENUMERATED{ disable(0), enable(1) },
    canChangeOperationProfile  ENUMERATED{ disable(0), enable(1) },
    canChangeConfigurationServices ENUMERATED{disable(0),enable(1)},
    canChangeTpConfiguration  ENUMERATED{ disable(0), enable(1)}
}

Cfg7 ::= SEQUENCE {
    CanChangePaDacOperatorConfiguration    ENUMERATED{    disable(0),
enable(1)},
    CanChangeMsgMachineConfiguration      ENUMERATED{    disable(0),
enable(1)}
}

--ACTIONS

CodeTransferInfo ::= ObjectInstance

CodeTransferResult ::= SEQUENCE {
    objectInstance ObjectInstance,
    cmdStatus CmdStatus
}

EgResetInfo ::= ObjectInstance

EgResetResult ::= SEQUENCE {
    objectInstance ObjectInstance,
    cmdStatus CmdStatus
}

DeactivatedServiceInfo ::= ObjectInstance

```

```
DeactivatedServiceResult ::= SEQUENCE {
    objectInstance ObjectInstance,
    cmdStatus CmdStatus
}

ActivatedServiceInfo ::= ObjectInstance

ActivatedServiceResult ::= SEQUENCE {
    objectInstance ObjectInstance,
    cmdStatus CmdStatus
}

DeactivatedServiceImmediatelyInfo ::= ObjectInstance

DeactivatedServiceImmediatelyResult ::= SEQUENCE {
    objectInstance ObjectInstance,
    cmdStatus CmdStatus
}

END
```