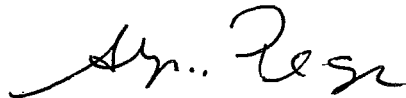


PADRÕES X.400: ANÁLISE, ESPECIFICAÇÃO FORMAL E PROPOSTA DE MIGRAÇÃO DE UM MHS

SUZAN KARINA ALMADA MENDES

Tese submetida ao corpo docente da Coordenação de Pós-Graduação de Engenharia da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários a obtenção do grau de Mestre em Ciências (M.Sc.) em Engenharia de Sistemas e Computação.

Aprovada por:



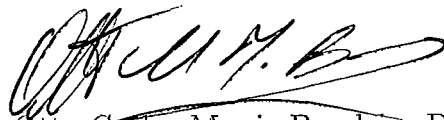
Prof. Aloysio de Castro Pinto Pedroza
(Presidente)



Prof. Valmir Carneiro Barbosa



Prof. Paulo Henrique de Aguiar Rodrigues



Prof. Otto Carlos Muniz Bandeira Duarte

Rio de Janeiro, RJ - Brasil

Janeiro de 1991

MENDES, SUZAN K. A.

Padrões X.400: Análise, Especificação Formal e Proposta de Migração de um MHS (RIO DE JANEIRO), 1991.

IX, 186 p., 29.7 cm (COPPE-UFRJ, M.Sc., Engenharia de Sistemas, 1991).

Tese - Universidade Federal do Rio de Janeiro, COPPE.

1. Redes de Computadores

I. COPPE-UFRJ

II. Título (série)

Aos meus pais, Eni & Elmo,
Aos meus irmãos Enio e Lilian,
As sobrinhas Yvonne e Ysabella.

AGRADECIMENTOS

Àquele que era, que é, e há de ser, e, sem o qual nada existiria.

Aos meus pais e irmãos pelo amor e energia, sempre.

Aos colegas Oswaldo, Luci, Esther, Marcelo, Sergio, Patrícia e Mikardo pela amizade, presença e força.

Ao professor Aloysio, verdadeiro orientador e amigo, sempre oportuno e presente.

Ao professor Valmir, pelo apoio.

Ao Núcleo de Computação Eletrônica da UFRJ que possibilitou o desenvolvimento desta tese.

Finalmente, a todos aqueles, que em qualquer plano, consciente ou inconscientemente contribuíram para a realização deste trabalho.

Resumo de Tese apresentada à COPPE-UFRJ como parte dos requisitos necessários a obtenção do grau de Mestre em Ciências (M.Sc.)

PADRÕES X.400: ANÁLISE, ESPECIFICAÇÃO FORMAL
E PROPOSTA DE MIGRAÇÃO DE UM MHS

SUZAN KARINA ALMADA MENDES

JANEIRO DE 1991

Orientador: ALOYSIO DE CASTRO PINTO PEDROZA

Programa: ENGENHARIA DE SISTEMAS E COMPUTAÇÃO

Este trabalho apresenta uma análise dos padrões X.400 para Sistemas de Manipulação de Mensagens (MHS), mostrando sua evolução, levantando aspectos problemáticos destes padrões em suas duas versões, 1984 e 1988, o impacto destes aspectos no sistema resultante, e definindo soluções para tais questões a nível de implementação.

A partir desta análise dos elementos funcionais e dos protocolos de MHS é desenvolvida uma especificação formal do sistema utilizando a Linguagem Estelle (ISO): Com esta modelagem obtém-se uma definição precisa do comportamento e das características relevantes do protocolos do MHS através dos recursos de Estelle. São também modelados os Serviços Comuns da Camada de Aplicação, permitindo uma visão global do funcionamento desta camada do Modelo OSI/ISO.

Finalmente, é definida uma estratégia de migração de um sistema X.400 1984 para um sistema X.400 1988. O processo parte de uma implementação de MHS 1984 onde as entidades de acesso e de transferência são co-residentes. Esta é a configuração predominante em sistemas X.400 1984 em operação, portanto, esta é uma estratégia que pode ser aplicada a várias implementações.

Com esta migração obtém-se uma configuração distribuída onde as entidades de acesso e de transferência se encontram em sistemas abertos distintos e as entidades de acesso dispõem de entidades de armazenamento remotas. O sistema resultante pode oferecer a funcionalidade de um MHS 1988 e interoperabilidade com MHS 1984.

Abstract of Thesis presented to COPPE-UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

**X.400 STANDARDS: ANALYSIS, FORMAL SPECIFICATION
AND MIGRATION PROPOSAL OF A MHS**

SUZAN KARINA ALMADA MENDES

JANUARY, 1991

Thesis Supervisor: ALOYSIO DE CASTRO PINTO PEDROZA

Department: ENGENHARIA DE SISTEMAS E COMPUTAÇÃO

This work presents an analysis of Message Handling Systems X.400 (MHS) standards, showing their evolution, discussing unsolved aspects of the 1984 e 1988 versions, and the impact of these aspects in the resulting system. Solutions at the implementation level are also presented.

A formal specification of the system is developed from the analisys of the functional elements and MHS protocols using the Estelle Language (ISO). A precise definition of the behavior and of important characteristics of the MHS protocols was obtained using Estelle resources. The Common Application Services were also modelled, allowing a global view of the correspondent layer in the OSI/ISO Model.

Finally, a migration strategy from a X.400 1984 to a X.400 1988 system was defined. The migration process is based on an MHS 1984 implementation where the access and tranfer entities are co-residents, which is the most commom configuration in X.400 1984 systems in operation today. This strategy that can be applied to several implementations.

As a result of this migration process a configuration where the access and transfer entities are in different open systems, and the access entities have remote storage entities was obtained. The resulting system offers MHS 1988 functionality and MHS 1984 interoperability.

Índice

I	Considerações Iniciais	1
I.1	Motivação	1
I.2	Objetivos Básicos	3
I.3	Contexto do Trabalho	4
I.4	Processo de Padronização do MHS	5
I.5	Estrutura da Tese	7
II	O Sistema de Manipulação de Mensagens	8
II.1	Introdução	8
II.2	Modelo OSI/ISO e o MHS	9
II.3	O Modelo do MHS	11
II.3.1	O Modelo Funcional do MHS	11
II.3.2	Configuração Física dos Elementos do MHS	15
II.3.3	Estrutura das Mensagens	16
II.3.4	O Sistema de Mensagens Interpessoal	17
II.3.5	Intercomunicação com Serviços de Entrega Física	19
II.3.6	O Uso do Sistema de Diretório pelo MHS	20
II.3.7	Listas de Distribuição	21
II.3.8	Segurança no MHS	24
II.4	Serviços do MHS	25
II.4.1	Serviço de Transferência de Mensagens	26
II.4.2	O Serviço de Mensagens Interpessoais	27
II.5	Protocolos e Endereçamento no MHS	30
II.5.1	A Estrutura da Camada de Aplicação	30
II.5.2	Os Protocolos do MHS	30
II.5.3	Domínios de Gerenciamento	34
II.5.4	Endereçamento no MHS	34

II.6	Comentários	36
III	Especificação Formal do MHS	37
III.1	Introdução	37
III.2	Aspectos Relativos à Especificação de Protocolos	38
III.3	As Linguagens Utilizadas	40
III.3.1	Estelle	40
III.3.2	ASN.1, ROS e sua Aplicação em MHS	45
III.4	Modelagem do MHS	52
III.4.1	Arquitetura Geral do Modelo	52
III.4.2	Modelagem do Agente Usuário (UA)	59
III.4.3	Modelagem do "Message Store" (MS)	62
III.4.4	Modelagem do Sistema de Transferência de Mensagens (MTS)	73
III.4.5	Modelagem dos Serviços Comuns de Aplicação (CASE)	98
III.5	O Sistema de Auxílio para Projetos em Estelle	109
III.6	Comentários e Propostas a Estelle	111
IV	Proposta de Migração de um MHS-84 para um MHS-88	115
IV.1	Introdução	115
IV.2	A Evolução dos Padrões X.400: Por que Migrar?	115
IV.2.1	Entrega de Mensagens e a Configuração Física do UA	116
IV.2.2	Redirecionamento de Mensagens	117
IV.2.3	Ações automáticas relacionadas à recepção de mensagens ("auto-actions")	117
IV.2.4	Prevenção e Detecção de "Loops"	118
IV.2.5	Uso dos Serviços de Diretório	118
IV.2.6	Listas de Distribuição	119
IV.2.7	Armazenamento de Mensagens	120
IV.2.8	Aspectos não Abordados pelos Padrões X.400	120
IV.2.9	Extensões aos Protocolos	122
IV.3	Proposta de Migração	127
IV.3.1	Primeira Fase: Distribuição das Entidades de Acesso do MHS (UA) e Implantação da Entidade de Armazenamen- to (MS)	128

IV.3.2 Segunda Fase: Uso dos Serviços de Diretório e das Extensões do Protocolo P1	129
IV.4 Implementação da Fase 1	130
IV.4.1 Arquitetura e Estruturas de Dados do Agente Usuário . . .	131
IV.4.2 Arquitetura e Estruturas de Dados do Message Store (MS)	142
IV.4.3 Ambiente e Estrutura de Implementação	155
IV.4.4 Comentários e Propostas	162
IV.5 Aspectos Relativos ao Desenvolvimento da Fase 2	165
IV.5.1 Resolução de Endereços	165
IV.5.2 Maximização de Conectividade e Roteamento	166
IV.5.3 Listas de Distribuição	167
IV.5.4 A Questão da Interoperabilidade	167
IV.6 Comentários	171
V Conclusões	173
V.1 Introdução	173
V.2 Resultados	173
V.3 Validação da Proposta de Implementação	176
V.4 Avaliação da Implementação e Testes de Conformidade	177
V.5 Continuidade do Trabalho e Futuras Tendências	178
Bibliografia	181

Capítulo I

Considerações Iniciais

I.1 Motivação

A sociedade moderna vem procurando ao longo dos anos aumentar cada vez mais a sua capacidade de comunicação a nível internacional. Dada a crescente necessidade de serviços mais integrados nas redes de comunicação de dados, espera-se uma maior integração dos serviços atualmente oferecidos, além do surgimento de novos serviços, com a utilização intensa de novas tecnologias digitais em redes.

Para tornar possível o estabelecimento efetivo de comunicação entre os diversos usuários de sistemas heterogêneos, os órgãos internacionais de padronização, como a International Standard Organization (ISO), desenvolveram um modelo comum para comunicação de dados, conhecido como Modelo de Referência para Interconexão de Sistemas Abertos, RM-OSI, ou Modelo OSI. Este modelo provê a base para desenvolvimento de protocolos e para a intercooperação entre ambientes heterogêneos. O modelo OSI divide os problemas de comunicação em sete camadas, cada uma com um nível lógico de abstração, e padroniza serviços e protocolos para comunicação fim-a-fim em cada um destes níveis.

Em particular, o estabelecimento em vários países de serviços telemáticos e de mensagens baseados em computador associados a redes de dados públicas criou a necessidade de produção de padrões para estes serviços, de modo a facilitar a troca de mensagens a nível internacional. Baseado no modelo OSI, o Comité Consultatif International Téléphonique et Télégraphique, CCITT, publicou uma série de recomendações que define um conjunto de procedimentos padrão conhecidos como Recomendações X.400. Estas Recomendações definem um Sistema para Manipulação de Mensagens, MHS, capaz de gerar, transmitir e entregar uma grande variedade de tipos de mensagens entre vários sistemas abertos. O serviço de manipulação de mensagens é provido pela combinação de ações de várias enti-

dades funcionais, em geral espalhadas por vários sistemas abertos. Em particular, a adoção das Recomendações X.400 tem recebido aprovação da comunidade internacional como padrão para os Sistemas Eletrônicos de Mensagens ou Correio Eletrônico.

Várias organizações têm trabalhado na área de padronização dos MHS. As Recomendações X.400 foram desenvolvidas no período de 1981 a 1984, e publicadas no chamado "Red Book" (X.400-84). Nos últimos anos, o CCITT e a ISO reuniram esforços na produção de uma versão revisada dos padrões X.400, publicada em 1988 no chamado "Blue Book", e portanto também conhecido como X.400-88.

O Blue Book estende significativamente o MHS, além de esclarecer alguns dos tópicos omitidos ou não especificados de forma completa pelo seu predecessor. Entretanto, alguns outros aspectos continuam a cargo do implementador.

A falta de soluções padrão nas Recomendações de 1984 para a indisponibilidade da máquina destino em aceitar entrega de mensagens; a falta de suporte para comunicação em grupo e de serviços de diretório reduzem significativamente a capacidade de distribuição do sistema. Desta forma, cada equipe de implementação teve que buscar sua própria solução.

A versão 1988 da série X400 provê solução padrão para os itens citados acima além de definir outros tópicos antes omitidos ou deficientemente especificados. Alguns itens, entretanto, continuam a cargo das equipes de implementação.

A coleção de serviços e protocolos de comunicação desenvolvidos para Interconexão de Sistemas Abertos tem o objetivo de permitir a intercooperação de sistemas computacionais heterogêneos para uma variedade de aplicações. Entretanto, a existência de padrões não torna direto o projeto de um sistema aberto baseado no modelo OSI (sistema OSI), uma vez que:

- (i) Mais de um tipo de serviço e/ou protocolo é definido para a maioria das camadas do modelo;
- (ii) Um serviço e/ou protocolo em particular pode incluir várias opções;
- (iii) Nas camadas superiores podem existir várias entidades ou subconjuntos funcionais.

Enfim, a correta implementação de sistemas OSI exige um investimento substancial no entendimento dos padrões e na avaliação das implicações de diferentes escolhas de projeto [41].

No caso das Recomendações X.400, o esforço de implementação esbarrou em questões de projeto deixadas explícita ou implicitamente nas normas como "ques-

tões locais”. Estas questões incluem entre outras, a disponibilidade de uma máquina destino em aceitar entrega de mensagens, ações relacionadas a esta entrega, detecção e supressão de "loops", serviços de diretório, suporte a listas de distribuição, estrutura de caixas postais, identificação de usuários no meio externo, e roteamento de mensagens. Além disso, os protocolos não foram formalmente especificados e varias versões do Guia do Implementador X.400 (Implementor's Guide [38]) foram publicadas de forma a esclarecer certos tópicos insuficientemente definidos.

Portanto, a especificação dos protocolos são de particular importância, já que representam os padrões que são a base para a implementação e teste de sistemas abertos compatíveis e cooperantes. Geralmente, as especificações contidas nos padrões são escritas em linguagem natural. Especificações de protocolos em linguagem natural frequentemente são ambíguas, imprecisas e até mesmo incompletas. Conseqüentemente, as implementações baseadas em tais especificações podem ser incompatíveis e de difícil interoperação. Além disso, tais protocolos não podem ser facilmente validados e testados.

Pensando nisto, foram criados na ISO e no CCITT grupos de trabalho em "Técnicas de Descrição Formal" (FDT) para aplicação em OSI. Estes grupos desenvolveram três linguagens de especificação: Estelle (ISO), LOTOS (ISO) e SDL (CCITT).

O papel principal das especificações formais é prover referências definitivas e precisas que previnam interpretações ambíguas e conseqüentemente implementações incompatíveis. Dentro do espírito OSI, a principal exigência para as FDTs é permitir especificações dos serviços e protocolos OSI "independentes de implementação". As técnicas de descrição formal são úteis para o projeto, especificação, verificação, teste e implementação de protocolos, pois, a natureza formal das especificações torna possível a aplicação de ferramentas automatizadas durante o desenvolvimento do protocolo.

I.2 Objetivos Básicos

Com o progresso da padronização para Interconexão de Sistemas Abertos (OSI), espera-se que as redes OSI sejam largamente utilizadas em um futuro próximo. Dada a importância atual dos sistemas eletrônicos de mensagens e o esforço internacional de migração para a estrutura de comunicação definida pelo modelo OSI [11], os objetivos deste trabalho são analisar, mostrar a evolução dos padrões X.400 para MHS, discutindo e expondo soluções para os aspectos conflitantes dos

padrões, especificar formalmente os protocolos de MHS 1988 e propor a migração de um MHS construído de acordo com as Recomendações X.400 de 1984 (MHS-84) para um MHS de acordo com as Recomendações de 1988 (MHS-88). A proposta de migração consiste de duas fases distintas e bem definidas, o que permite que a complexidade do processo seja diluída, e propiciará o aumento da funcionalidade a nível de protocolos e serviços e da distribuição de um MHS-84.

A implementação de um MHS baseado nas Recomendações X.400 1984 fez parte do projeto REDE-RIO, financiado pela FINEP, com o objetivo de interligar os centros de pesquisa e universidades do Rio de Janeiro seguindo a tendência internacional de basear os desenvolvimentos de software e hardware no Modelo OSI. O projeto REDE-RIO resultou na geração de protótipos de aplicações como Terminal Virtual, Manipulação e Transferência de Jobs (JTM), Manipulação, Acesso e Transferência de Arquivos (FTAM), e Sistema de Mensagens baseado nos padrões X.400 de 1984 (MHS-84), e das camadas de sessão e transporte do Modelo OSI.

A proposta de migração, baseada na experiência adquirida durante o projeto REDE-RIO, definirá as etapas que serão seguidas para o desenvolvimento de um MHS que tenha a funcionalidade de um MHS-88 com capacidade de interoperação de um MHS-84, uma vez que o modelo do MHS-88 não se encontra completamente disseminado e a maioria dos Sistemas de Manipulação de Mensagens correntemente em operação são baseados em X.400 1984.

I.3 Contexto do Trabalho

Esta tese faz parte do projeto Inter-X do NCE-UFRJ e baseia-se na experiência adquirida com o projeto REDE-RIO. O projeto Inter-X objetiva gerar um Sistema Eletrônico de Mensagens avançado e distribuído, em conformidade com as Recomendações X.400. Este sistema deverá suportar a funcionalidade existente nos padrões X.400 1988 e a capacidade de interoperabilidade de um MHS-84; já que o modelo X.400 1988 não está completamente disseminado e a maioria dos produtos internacionais existentes ainda seguem o padrão X.400 1984. Além disso, este sistema será acoplado a um Sistema de Diretório baseado nas Recomendações X.500 [21] que facilitará a interação entre usuários e o gerenciamento de funções do Sistema.

Este projeto permitirá, em primeira instância, que usuários em microcomputadores do tipo PC e no equipamento VAX 8810, ligados por uma rede local do tipo Ethernet possam trocar mensagens entre si utilizando o MHS. Numa etapa posterior, para completar o processo de validação e testes, o sistema Inter-X será

acoplado a redes baseadas em endereçamento do tipo RFC 822, tais como Internet e Bitnet, através de comportas (gateways).

Com a crescente popularidade e suporte aos sistemas em conformidade com as Recomendações X.400, espera-se que as redes baseadas em X.400 sejam o maior tipo de sistema de mensagem nos próximos anos. Naturalmente, os outros tipos de sistemas de mensagens não desaparecerão e, portanto a comunicação com estes sistemas será, e, atualmente tem sido, conseguido através das comportas (gateways).

I.4 Processo de Padronização do MHS

Em 1984, o CCITT publicou uma série de recomendações, isto é, padrões, que regulariam a comunicação entre sistemas públicos de mensagens e a comunicação entre sistemas de mensagens públicos e privados. Estes padrões ficaram conhecidos como X.400-84, ou "Red Book". Mais tarde a ISO produziu um documento semelhante, mas estendido, chamado Message Oriented Text Interchange System (MOTIS). O resultado foi que durante alguns anos, co-existiam dois padrões para MHS. As desvantagens desta situação são imagináveis. Nos últimos anos ISO e CCITT trabalharam em conjunto na elaboração e publicação de uma nova versão das Recomendações X.400 em 1988 ("Blue Book").

As Recomendações X.400 1988 representam uma maior expansão dos padrões para MHS. Além disso, as séries X.400 em 1984 e 1988 diferem em sua organização. Durante a reestruturação algumas das recomendações da antiga série X.400 receberam números na série X.200, que define o Modelo OSI, ou seja, alguns elementos passaram a prestar serviços comuns neste Modelo. A Tabela I.1 mostra a relação entre as séries de recomendações X.400 de 1984 e 1988.

A primeira recomendação da série, a recomendação X.400, continua a apresentar uma visão geral do MHS na versão de 1988, mas a série passou a conter mais uma recomendação, a X.402, que aborda os mesmos aspectos da recomendação X.400 com um maior nível de detalhe e utilizando linguagem mais técnica. A classificação dos serviços de manipulação de mensagens passaram da recomendação X.401 em 1984 para um anexo da recomendação X.400 em 1988. A recomendação X.401, portanto, não faz parte da série X.400 de 1988.

A recomendação X.407 foi adicionada à serie X.400 de 1988. Este documento introduz os conceitos da notação abstrata fortemente utilizada em toda série X.400 de 1988.

A recomendação X.408, que define as regras de conversão no MHS, se manteve

1984 (CCITT)	1988 (CCITT)	1988 (ISO)
X.400	X.400 X.402	8505-1 8505-2
X.401	Anexo à X.400	—
	X.407	8883-2
X.408	X.408	—
X.409	X.208 X.209	8824 8825
X.410	X.217 X.277 X.218 X.228 X.219 X.229	8649 8650 9066/1 9066/2 9072/1 9072/2
X.411	X.411 parte da X.419	8883-1 8505-2
	X.413 parte da X.419	TBS-1 8505-2
X.420	X.420	9065
X.430	T.330	—

Tabela I.1: Correlação entre X.400-84 e X.400-88

em nome e organização na nova série.

A recomendação X.409 de 1984, que define as sintaxes abstrata e concreta utilizadas para definir os tipos de dados das mensagens, foi removida da série X.400 de 1988 e somada à série X.200: a recomendação X.208 define a notação de sintaxe abstrata (Abstract Syntax Notation - ASN.1), e a X.209 define as regras de codificação na sintaxe de transferência.

Semelhantemente, a recomendação X.410 de 1984, que define as Operações Remotas e o Serviço de Transferência Confiável, foi removida da série X.400 de 1988 e somada à série X.200: as recomendações X.219 e X.229 definem as Operações Remotas e as X.218 e X.228 definem o Serviço de Transferência Confiável. Além disso, os protocolos de MHS nesta nova versão utilizam o Elemento de Serviço de Controle de Associação (ACSE) definido nas recomendações X.217 e X.227.

As recomendações X.411 e X.420 de 1984 permanecem na série de 1988 definindo as operações do Agente de Transferência de Mensagens (MTA) e Agente Usuário (UA), respectivamente. A recomendação X.413 é adicionada à série X.400 em 1988 para definir as operações do Message Store. A recomendação X.419 é também adicionada à série para definir os protocolos do MHS.

A definição de um protocolo de acesso para terminais Teletex, objetivo de recomendação X.430 de 1984, agora faz parte da recomendação T.330.

I.5 Estrutura da Tese

O capítulo II descreve o modelo e os elementos do MHS, mostrando a evolução dos padrões X.400 e como este se enquadra no Modelo OSI/ISO.

O capítulo III descreve a especificação formal do MHS desenvolvida para este trabalho utilizando a linguagem Estelle.

O capítulo IV apresenta a proposta de migração de um MHS de acordo com as Recomendações X.400 1984 para um MHS em conformidade com as Recomendações X.400 1988 baseado na modelagem Estelle descrita no Capítulo III.

O capítulo V tece comentários finais sobre todo o trabalho.

Capítulo II

O Sistema de Manipulação de Mensagens

II.1 Introdução

Os Sistemas de Manipulação de Mensagens (MHS) permitem que seus usuários troquem mensagens segundo o mecanismo armazena-envia ("store-and-forward") através de cooperação de suas entidades funcionais espalhadas em diversos sistemas abertos.

As Recomendações X.400 padronizam o MHS e, evoluíram nos últimos anos com o objetivo de esclarecer aspectos antes definidos de maneira incompleta, de aumentar sua funcionalidade a nível dos serviços e facilidades, e de aumentar a capacidade de distribuição do sistema.

As Recomendações X400 1988 diferem drasticamente das recomendações X.400 1984 em aparência e apresentam definição mais formal e precisa dos serviços. Apesar do modelo no qual as definições são baseadas continuar basicamente o mesmo, foram necessárias extensões para que a funcionalidade dos serviços de manipulação de mensagens fosse aumentada.

O objetivo deste capítulo é descrever o MHS, mostrando como ele se enquadra no Modelo OSI, suas entidades funcionais, seu modo de endereçamento, seus protocolos e os serviços oferecidos, com a preocupação de apresentar a evolução das Recomendações X.400.

A seção II.2 deste capítulo descreve brevemente o Modelo OSI da ISO. A seção II.3 descreve o modelo e os elementos do MHS. A seção II.4 descreve os serviços do MHS. A seção II.5 descreve os protocolos e o modo de endereçamento do MHS. A seção II.6 tece comentários sobre o capítulo.

II.2 Modelo OSI/ISO e o MHS

A International Organization for Standardization (ISO) desenvolveu um Modelo de Referência para Interconexão de Sistemas Abertos (Modelo OSI) com o intuito de construir uma base comum para o desenvolvimento de padrões para a intercomunicação de sistemas, permitindo que os padrões existentes fossem englobados neste modelo [1].

Um outro propósito do modelo é identificar áreas que se prestam ao desenvolvimento e aperfeiçoamento de padrões, fornecendo uma referência comum que mantenha a consistência entre os padrões.

No Brasil, o modelo OSI/ISO foi estabelecido como padrão para o desenvolvimento de protocolos. A Associação Brasileira de Normas Técnicas (ABNT) referendou este modelo de referência e a Telebrás publicou o documento "Modelo de Referência para Interconexão de Sistemas Abertos Através da Rede Pública de Comunicações" [2] seguindo o mesmo propósito.

O Modelo OSI desenvolve uma estrutura em camadas ou níveis, cada um com um conjunto bem definido de funções, conforme mostra a Figura II.1. Os níveis são separados por interfaces definidas em termos dos serviços oferecidos por um nível ao nível imediatamente superior. A cooperação entre entidades de mesmo nível em sistemas abertos distintos processa-se segundo protocolos padronizados que definem os formatos e as sequências de mensagens que podem ser enviadas e recebidas em cada instante. Exceto para o nível mais alto, cada nível (N) fornece às entidades do nível superior (N+1) os serviços (N), que são obtidos usando as funções (N) executadas no nível (N) e os serviços disponíveis do nível (N-1) [3]. O modelo OSI identifica sete níveis ou camadas, a saber:

- Nível 1: Físico
- Nível 2: Enlace
- Nível 3: Rede
- Nível 4: Transporte.
- Nível 5: Sessão.
- Nível 6: Apresentação.
- Nível 7: Aplicação.

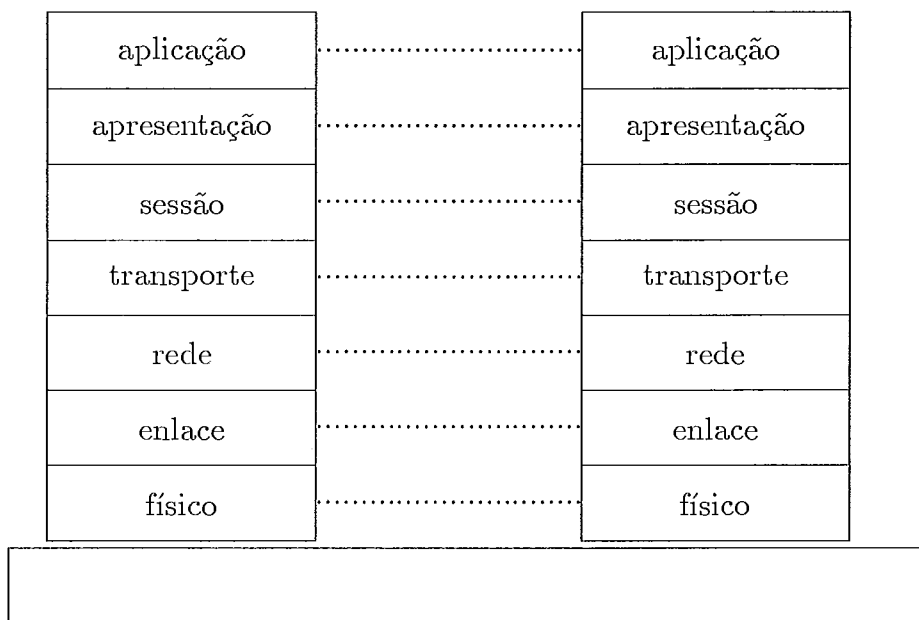


Figura II.1: Modelo OSI

O Nível de Aplicação, em particular, tem como objetivo servir de "janela" entre usuários comunicantes no ambiente OSI, através da qual ocorre toda troca de informação significativa entre esses usuários. Cada usuário é representado para os demais pela sua entidade de aplicação correspondente.

É importante observar que a totalidade de uma aplicação final não reside completamente nesta camada. Apenas a parte da aplicação que precisa se comunicar com entidades remotas de aplicação fazem parte desta camada e utiliza protocolos de aplicação. O restante não se enquadra no Modelo OSI.

O MHS é projetado de acordo com os princípios do modelo de referência OSI, enquadra-se na camada de aplicação deste modelo e utiliza os serviços de apresentação e os serviços oferecidos pelos elementos de serviço comuns da camada de aplicação. Portanto, um MHS pode ser construído sobre qualquer rede que se enquadre no Modelo OSI.

II.3 O Modelo do MHS

II.3.1 O Modelo Funcional do MHS

A recomendação X.400 define um modelo constituído de diferentes entidades funcionais que cooperam a fim de prover os serviços de manipulação de mensagens. Inicialmente, é definido um Ambiente para Manipulação de Mensagens, constituído pelo próprio MHS e todos os seus usuários.

Os serviços oferecidos pelos MHS tornam o Ambiente para Manipulação de Mensagens um meio de intercâmbio entre seus usuários para fins de troca de mensagens segundo o mecanismo armazena-envia (store-and-forward).

Segundo as Recomendações de 1984, o MHS é constituído por um conjunto de Agentes Usuários (User Agents - UA) e pelo Sistema de Transferência de Mensagens (Message Transfer System - MTS). O MTS é constituído de Agentes de Transferência de Mensagens (Message Transfer Agent - MTA). As Recomendações de 1988 estendem este modelo, que passa a conter duas novas entidades funcionais: o Message Store (MS) e as Unidades de Acesso (Access Units - AU).

O modelo funcional do MHS 1984 e 1988 são ilustrados nas Figuras II.2 e II.3, e suas entidades funcionais são descritas a seguir.

(i) - O Usuário do MHS

Um usuário do MHS pode ser uma pessoa, interagindo com o sistema através de um dispositivo de Entrada e Saída; ou pode ser um processo aplicativo do computador.

Os usuários interagem com os UAs para fazerem uso dos serviços oferecidos pelo MHS. Os UAs utilizam os serviços oferecidos pelo MTS para efetuar a transferência de mensagem através das redes de comunicação de dados.

(ii) - O Agente Usuário (UA)

O UA é o componente através do qual o usuário interage como o MHS. Ele o assiste na transmissão e recepção de mensagens. O UA contém as funções necessárias para interagir com o MTS na sua operação de submissão e entrega de mensagens.

A principal função de UA é tornar os serviços de manipulação de mensagens disponíveis ao usuário, usando o serviço provido pelo MTS.

O UA é um conjunto de processos de aplicação contendo, no mínimo, as funções necessárias às interações de submissão e entrega de mensagens junto

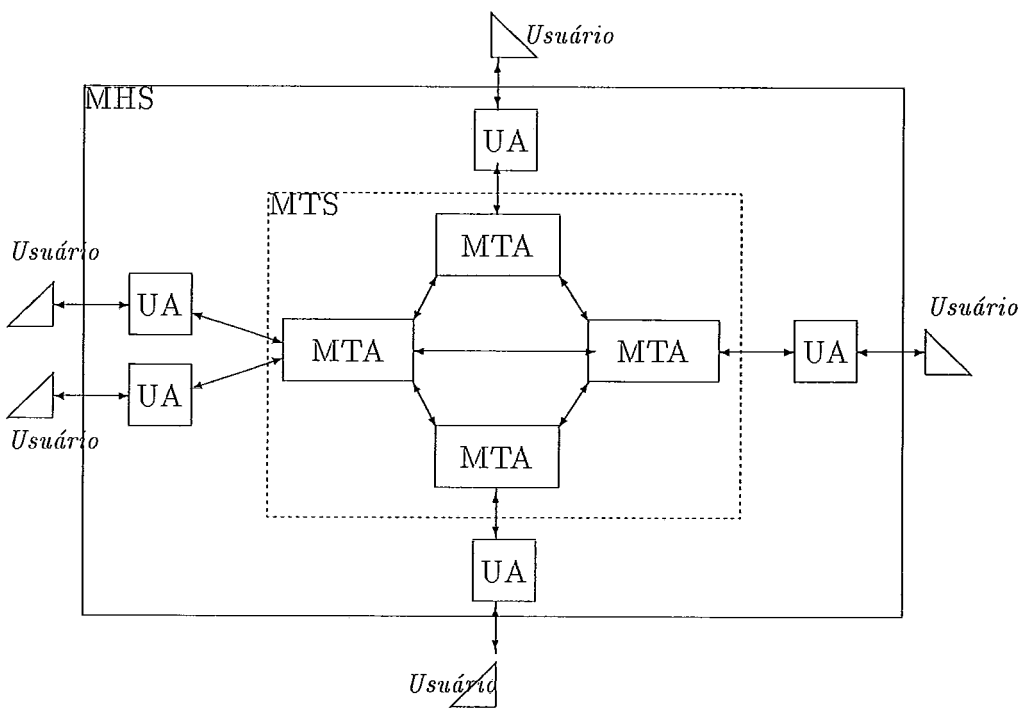


Figura II.2: Modelo Funcional do MHS-84

ao MTS.

Os UAs são agrupados em classes baseando-se no tipo de conteúdo de mensagens que podem manipular. UAs de uma mesma classe são chamados "UAs cooperantes", já que devem cooperar a fim de fornecer serviços e comunicação a seus respectivos usuários. Para que tal cooperação se estabeleça, os UAs utilizam protocolos padronizados.

Um usuário acessa seu UA a fim de obter serviços de processamento de mensagens. Para tal, o usuário interage com seu UA via dispositivo de entrada/saída ou um processo. Um UA pode ser implementado como um conjunto de processos de um sistema operacional ou em um terminal inteligente.

(iii) - O Agente de Transferência de Mensagens (MTA)

O MTA é uma entidade de propósito geral que suporta todas as aplicações de manipulação de mensagens. O MTS é constituído de um conjunto de MTAs, que interagem entre si em modo armazena-envia para retransmitir as mensagens, entregando-as aos UAs destinatários. Mensagens entram e saem do MTS através de um protocolo de submissão e entrega.

O MTS provê o meio para troca de mensagens. Existem duas interações básicas entre MTAs e seus usuários: a interação de submissão e a interação de entrega de mensagens. A partir do MTA originador, cada MTA transfere a mensagem a outro MTA até que a mensagem chegue ao MTA destino, que então a entrega ao UA destino usando a interação de entrega.

Nas Recomendações X.400 1988 este modelo foi estendido e neste, o MHS contém duas novas entidades: o Message Store (MS) e as Unidades de Acesso (AU). O modelo funcional do MHS 1988 é mostrado na Figura II.3.

(iv) - As Unidades de Acesso (AUs)

As Unidades de Acesso são entidades funcionais que interligam outros sistemas de comunicação ao MTS, como um sistema postal ou uma rede telex. Isto permite que usuários destes outros sistemas sejam usuários indiretos do MHS. Através das AUs, o MHS está agora disponível a estes usuários.

Dois tipos de unidades de acesso são definidos: a Unidade de Acesso genérica para serviços Telemáticos, como Telex e Teletex; e uma Unidade de Acesso de Entrega Física (Physical Delivery Access Unit - PDAU). A PDAU habilita usuários do MHS a endereçarem destinatários que receberão a mensagem em

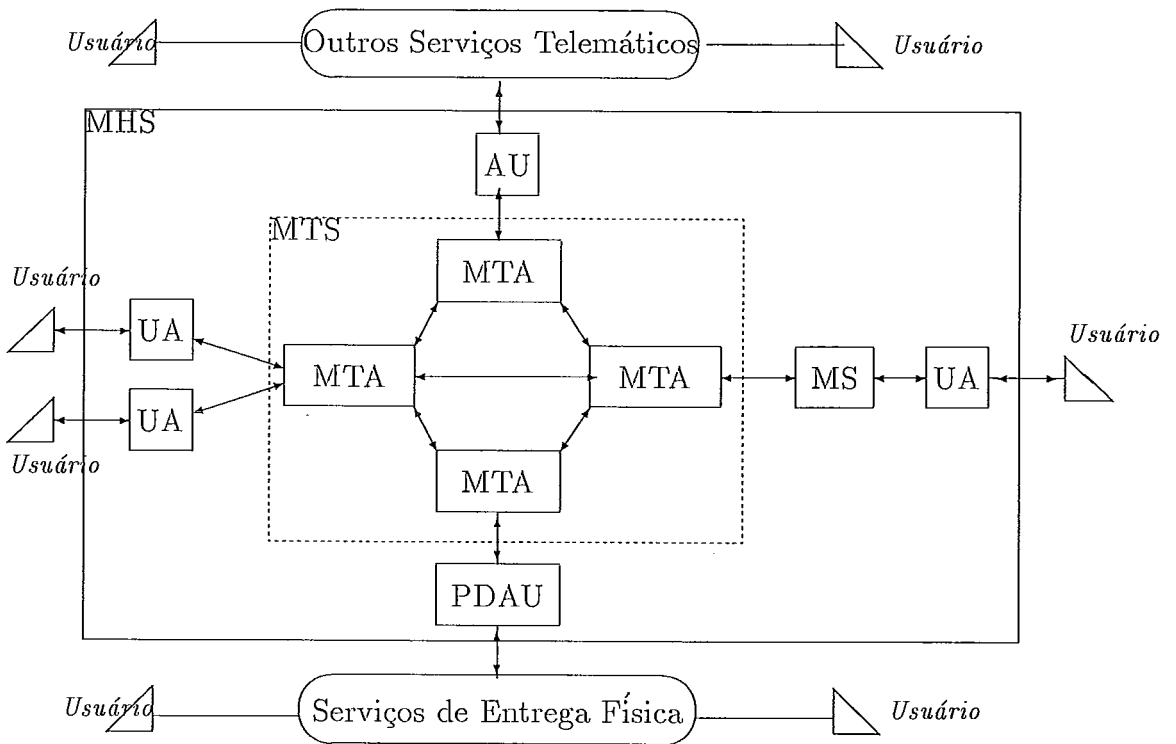


Figura II.3: Modelo Funcional do MHS-88

meio físico, por exemplo, impressão em papel. O PDAU realiza a transformação de uma "mensagem eletrônica" para uma "mensagem física" e a transporta ao sistema de entrega física. A intercomunicação entre o MHS e o Sistema de Entrega Física será descrita na seção II.3.5.

(v) - O Message Store (MS)

O MS é uma entidade funcional do MHS que age como intermediário entre o MTS e o UA. Ele interage com o MTA para recepção de mensagens e notificações e as armazena até que seja acessado pelo UA. O MS permite que o UA (usuário) manipule mensagens e notificações armazenadas (e.g., leitura, delegação) e permite que o UA submeta mensagens ao MTA. Esta submissão é realizada transparentemente através do MS.

Como os UAs podem ser implementados em equipamentos de pequeno porte, como microcomputadores do tipo PC, o MS pode complementar o UA, provendo um mecanismo de armazenamento disponível continuamente. Conceitualmente, o Message Store consiste de "bases de informação". Cada base de informação é uma coleção de entradas, cada uma das quais é um objeto tal como uma mensagem. As entradas são rotuladas com números de sequência.

A entidade Message Store foi definida para facilitar a integração de computadores de pequeno porte (e.g., PC) ao MHS. Dada a capacidade limitada do PC e o fato de que um PC agindo como UA ou MTA não tem a possibilidade de negociar a entrega de mensagens, tem-se a necessidade de uma entidade no sistema que possa armazenar mensagens.

II.3.2 Configuração Física dos Elementos do MHS

O modelo do MHS não impõe qualquer condição quanto à configuração física das diferentes entidades. UAs, MSs e MTA podem estar localizados em um mesmo ambiente (co-residentes), como na Figura II.4(a), ou, podem ser implementados em sistemas distintos. Também é possível que um MTA seja implementado em um sistema sem UAs ou MSs, em um sistema aberto fisicamente distinto.

O MS pode estar fisicamente localizado em relação ao MTA de várias maneiras. O MS pode estar co-residente com o UA, e/ou com o MTA, ou pode ser implementado em um sistema sem outras entidades, independente, em um sistema aberto fisicamente distinto.

Do ponto de vista externo, não existe distinção entre a configuração UA e MS co-residentes e a configuração UA independente. MS e MTA co-residentes remotos

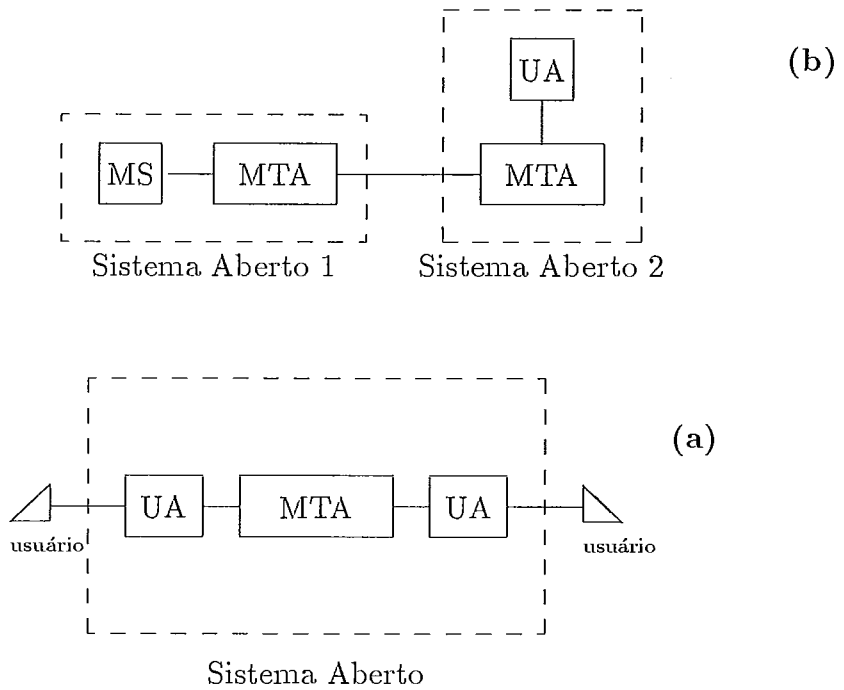


Figura II.4: Configuração Física do MHS

ao UA oferece vantagens significativas, conforme será visto no Capítulo IV, e esta configuração provavelmente será predominante.

Os exemplos das Figuras II.4(a) e II.4(b) não representam todas as possíveis variantes de implementações de MHS; a intenção aqui é demonstrar a variedade de possibilidades de configurações.

II.3.3 Estrutura das Mensagens

No MHS uma mensagem é composta de um envelope e um conteúdo (Figura II.5).

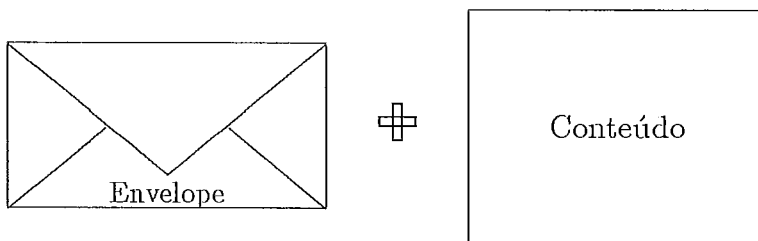


Figura II.5: Estrutura Básica das Mensagens no MHS

O envelope contém os endereços do originador e destinatário da mensagem bem como todas as informações necessárias ao encaminhamento da mensagem ao seu destino e à realização de serviços (e.g., pedido de notificação de entrega). O conteúdo da mensagem contém dados que o usuário deseja transferir e, portanto, passa transparentemente pelo MTS, a menos que serviços de conversão de conteúdo sejam solicitados.

II.3.4 O Sistema de Mensagens Interpessoal

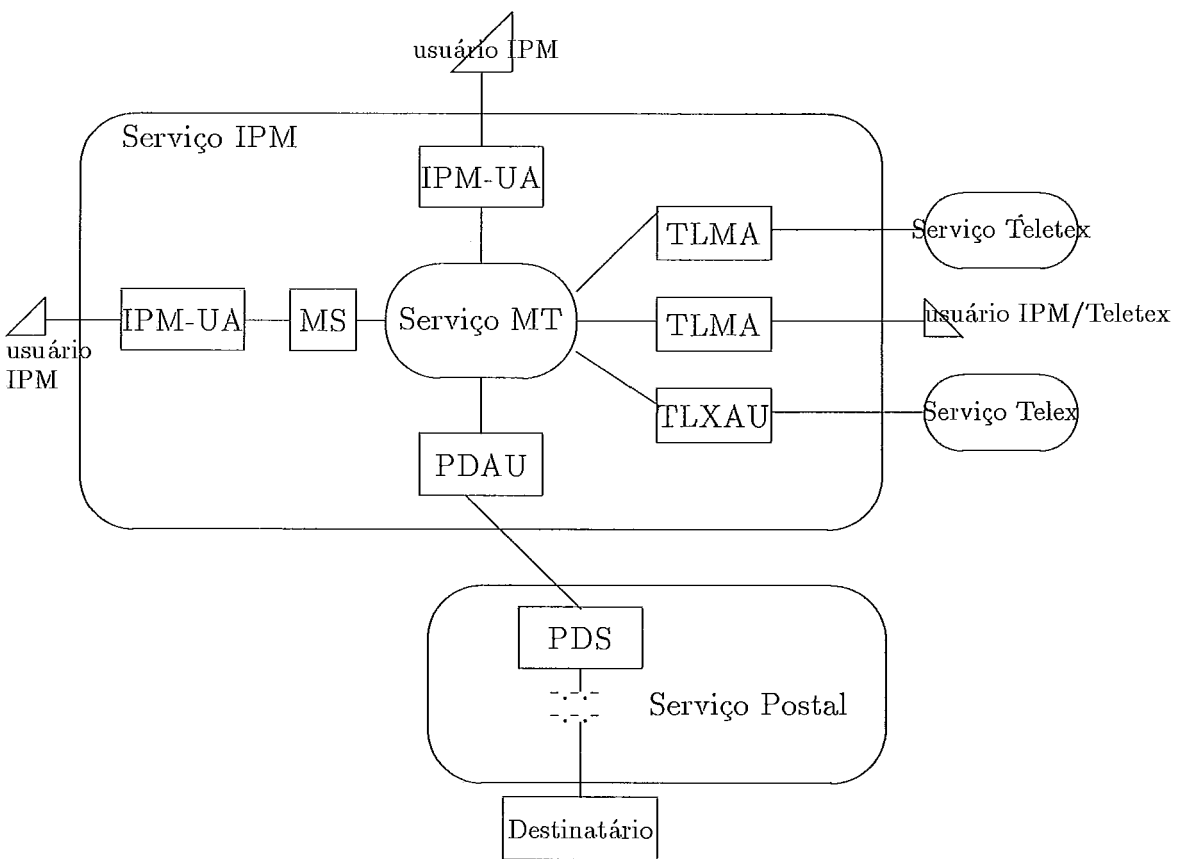


Figura II.6: O Sistema de Mensagens Interpessoal

Como os conteúdos das mensagens passam transparentemente pelo MTS, estes podem ter qualquer tipo de codificação (e.g., texto, telex, facsimile). Entretanto, a necessidade original do sistema era transferir mensagens entre pessoas. De forma a atender esta necessidade, as recomendações X.400 definem uma classe de UAs cooperantes que oferecem o Serviço de Mensagens Interpessoais (IPMS)

(Figura II.6). Os UAs que pertencem a esta classe denominam-se IPM-UAs. As recomendações também definem protocolos de acesso para os serviços de Telex e Teletex ao IPMS. Com isto o MHS se torna o primeiro provedor de facilidades de interoperação de serviços telemáticos antes incompatíveis. Nas recomendações de 1988 as especificações deste serviço foram melhoradas. Com estes melhoramentos foram definidos uma unidade de acesso para telex (Telex Access Unit - TLXAU), uma unidade de acesso para serviços telemáticos (Telematic Access Unit - TLMA), e uma unidade de acesso para entrega física (Physical Delivery Access Unit - PDAU), isto é, a conexão com o serviço de correio convencional.

O MTS permanece transparente para as mensagens interpessoais, também chamadas mensagens-IP, desta forma, os MTAs podem transferir mensagens cujo conteúdo é codificado de acordo com o IPMS bem como mensagens cujo conteúdo é codificado de acordo com alguma outra definição, mesmo que esta seja privada.

No IPMS uma mensagem é constituída de um "cabeçalho" e um "corpo". O cabeçalho é sub-dividido em campos de significação padrão, por exemplo, "destinatários", "assunto", "data-hora" (Figura II.7).

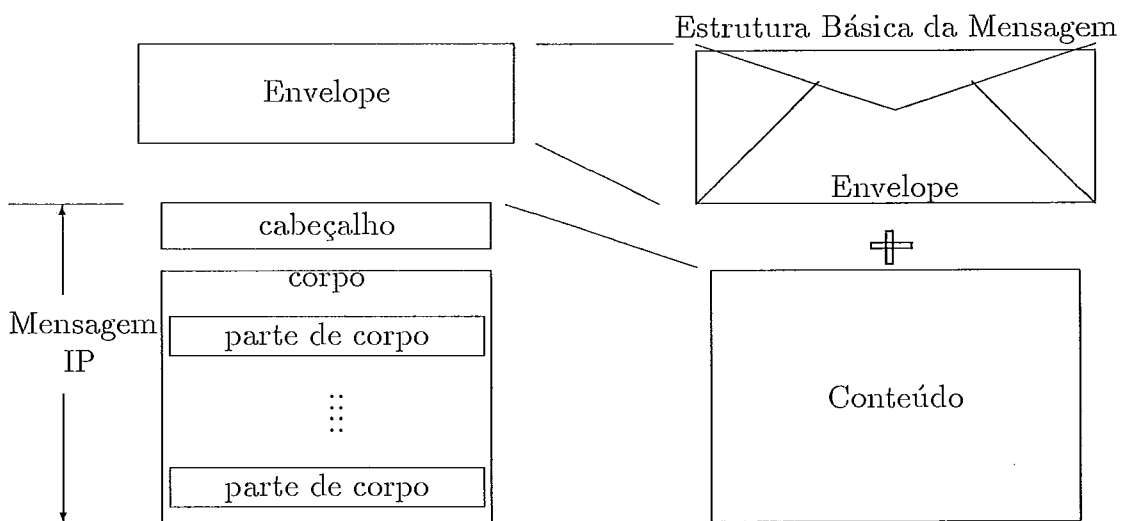


Figura II.7: Estrutura da Mensagem IP

O corpo da mensagem também pode ser estruturada em uma ou mais partes de corpo onde cada uma destas partes pode ter codificação distinta. Em particular, é possível que uma parte de corpo contenha uma outra mensagem-IP completa constituída de cabeçalho e corpo. Este encapsulamento é análogo à situação, no serviço de correio convencional, onde, uma carta contém, em seu envelope, uma outra carta, com seu próprio envelope

II.3.5 Intercomunicação com Serviços de Entrega Física

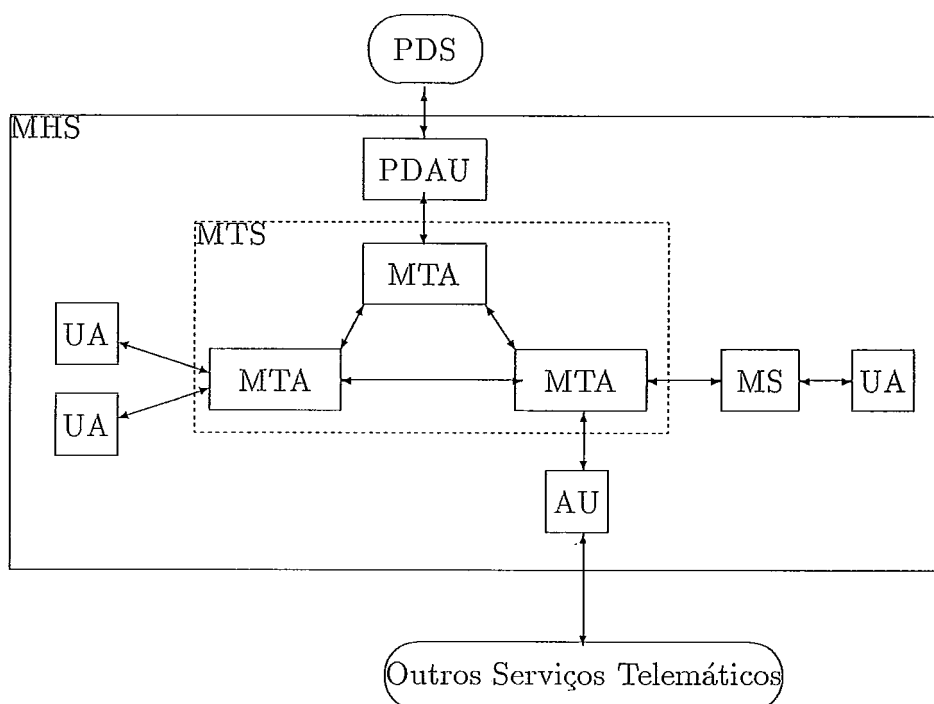


Figura II.8: A Intercooperação entre PDS e MHS

A funcionalidade do MHS pode ser aumentada conectando-o a Sistemas de Entrega Física (Physical Delivery Systems - PDS). Um PDS é um sistema que transporta e entrega mensagens físicas. Uma mensagem física é um objeto físico constituído de um envelope e seu conteúdo. Um exemplo de PDS é o serviço postal. Um exemplo de mensagem física é a carta convencional em papel em seu envelope.

A intercomunicação entre MHS e PDS permite a entrega física de mensagens originadas no MHS para destinatários fora do MHS através da Unidade de Acesso para Entrega Física (PDAU), e em alguns casos permitirá o retorno de notificações do PDS ao originador dentro do MHS.

Uma PDAU converte uma mensagem originada no MHS para a forma física. Este processo é chamado "physical rendition". Um exemplo disto seria a listagem de uma mensagem e seu automático envelopamento em um envelope de papel. O PDAU passaria então a mensagem física a um PDS para posterior encaminhamento e eventual entrega. A geração de mensagens em um PDS para submissão ao MHS

através do PDAU é ainda uma questão em estudo.

Uma PDAU pode ser visto como um conjunto de UAs, cada um dos quais sendo identificado por um endereço postal. Para realizar suas funções, uma PDAU deve suportar as interações de submissão e entrega com o MTS, e também cooperar com outros UAs. Portanto, a intercomunicação entre os serviços de Manipulação de Mensagens e os serviços de Entrega Física é provido como parte do serviço de transferência de mensagens.

A intercomunicação entre MHS e PDS é um serviço opcional do MHS, e pode ser utilizado por qualquer aplicação tal como o serviço IPM. A Figura II.8 mostra o modelo funcional desta intercomunicação. A intercooperação entre os serviços públicos de manipulação de mensagens e serviços de entrega física é descrita na Recomendação F.415.

II.3.6 O Uso do Sistema de Diretório pelo MHS

O Sistema de Diretório (DS) definido pela Série de Recomendações X.500 [21] funciona como um servidor de nomes e de informação para vários serviços de telecomunicações. O uso do DS pelo MHS foi introduzido pelas Recomendações X.400 1988. Esta seção descreve como o DS pode ser usado pelo MHS.

Os facilidades de Diretório usadas em Manipulação de Mensagens enquadram-se basicamente em quatro categorias:

- Identificação amigável: originadores e destinatários de mensagens podem ser identificados através de seu nome de diretório ao invés do endereço O/R, o MHS pode obtê-lo consultando o DS;
- Listas de Distribuição (DL): um grupo cujos membros estão armazenados no Diretório pode ser utilizado como uma DL. O originador somente fornece o nome da lista, e no ponto de expansão da DL, o MHS pode obter os endereços de cada membro da lista consultando o DS;
- Capacidades do UA destinatário: as capacidades de manipulação de mensagens de um usuário pode ser armazenada com o seu nome no DS. Quando necessário, o MHS pode obtê-las consultando o DS;
- Autenticação: antes que duas entidades funcionais do MHS se comuniquem é necessário que elas verifiquem a identificação uma da outra. Isto pode ser feito usando o serviço de autenticação do MHS baseado na informação armazenada no Diretório.

Além disso, um usuário pode acessar diretamente o DS, para, por exemplo, determinar o endereço O/R ou as capacidades de manipulação de mensagens de outro usuário. O nome de diretório é fornecido ao DS, que retorna a informação solicitada.

II.3.6.1 Modelo Funcional do Sistema de Diretório

Os UAs e os MTAs podem fazer uso do DS. Isto é feito através do Agente Usuário do Diretório (Directory User Agent - DUA) que por sua vez acessa o Agente do Sistema de Diretório (Directory System Agent - DSA) que fornece a informação desejada. O modelo funcional que descreve a intercooperação entre MHS e DS é mostrado na Figura II.9.

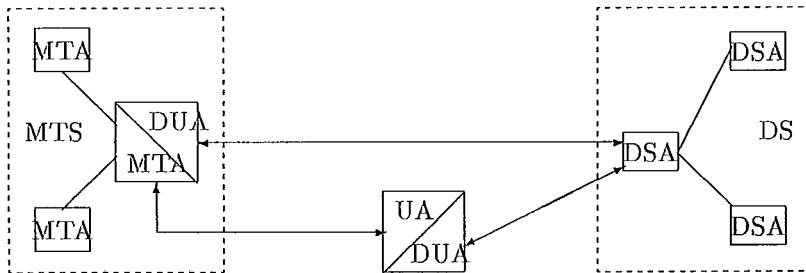


Figura II.9: Intercooperação entre MHS e DS

II.3.6.2 Configurações Físicas do DS

Algumas das possíveis configurações físicas do modelo funcional acima são mostrados na Figura II.10. Onde o DUA e DSA estão localizados em sistemas abertos distintos, um protocolo de diretório padronizado pelas Recomendações X.500 é utilizado para governar suas interações. Provavelmente, a configuração predominante será com UA ou MTA co-residentes a um DUA/DSA. Entretanto, outras configurações são também possíveis.

II.3.7 Listas de Distribuição

O uso de Listas de Distribuição (DL) é uma facilidade opcional do MHS provida através do serviço de transferência de mensagens. Esta facilidade foi introduzida de forma padronizada no MHS pelas Recomendações X.400 1988. A DL permite que o originador transmita uma mensagem para um grupo de destinatários, fornecendo somente o nome do grupo ao invés do de uma lista de nomes O/R.

As propriedades de uma DL são descritas abaixo:

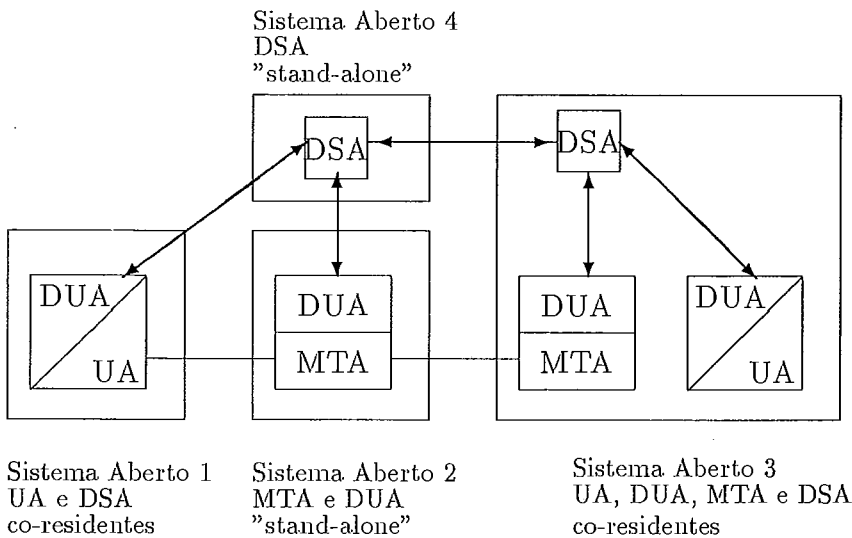


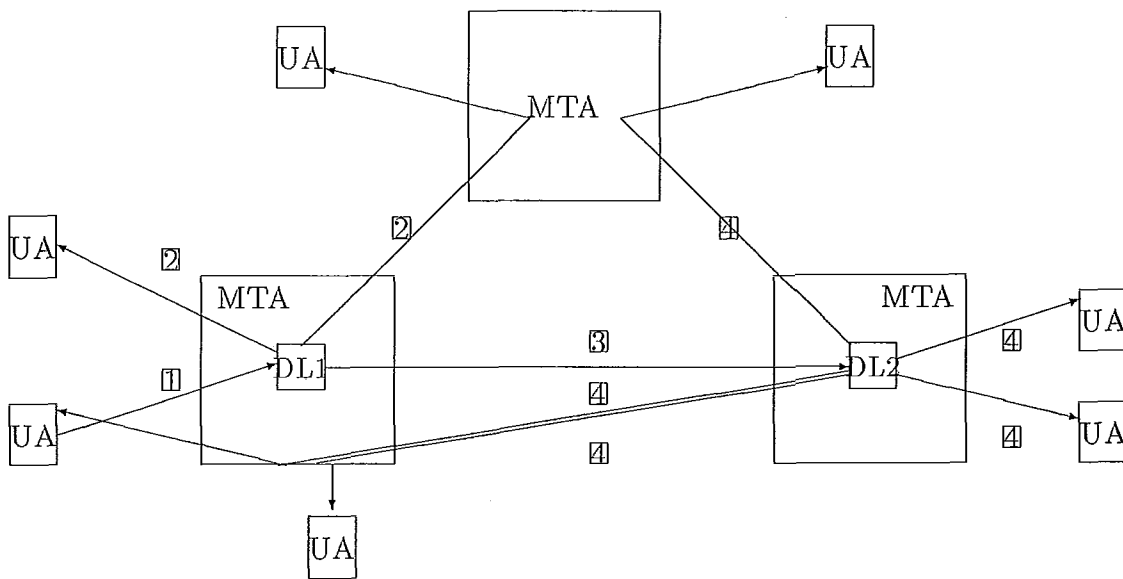
Figura II.10: Configurações Físicas DS-MHS

- **Membros de uma DL:** são os usuários ou outras DLs que podem receber mensagens endereçadas à DL;
- **Permissão para Submissão de uma DL:** é a lista de usuários e outras DLs aos quais é permitido fazer uso da DL para enviar mensagens aos seus membros;
- **Ponto de expansão de uma DL:** cada DL tem um único endereço O/R. Este endereço O/R identifica seu ponto de expansão, que é o domínio ou MTA onde os nomes dos membros da DL são adicionados à lista de destinatários da mensagem;
- **Proprietário de uma DL:** um usuário que é responsável pelo gerenciamento da DL.

O DS pode ou não ser utilizado para armazenar informação sobre as propriedades de uma DL.

A submissão de uma mensagem a uma DL é semelhante à submissão de uma mensagem a um usuário. O originador utiliza o nome O/R da mesma forma que utiliza para usuários normais (vide seção II.9), mas não necessita estar ciente que o nome O/R usado pertence a uma DL. O originador pode, entretanto, proibir o MTS de expandir uma mensagem inconscientemente endereçada a uma DL.

II.3.7.1 Expansão e Aninhamento de DLs



- 1 Submissão
- 2 Entrega após primeira expansão
- 3 Transferência
- 4 Entrega após segunda expansão

Figura II.11: Expansão de Listas de Distribuição

No ponto de expansão, o MTA responsável pela expansão da DL, obterá informação sobre a DL e verificará se o originador tem permissão para submissão à esta DL. Se a expansão é permitida, o MTA adiciona os endereços dos membros da DL à lista de destinatários da mensagem e segue a transmissão da mensagem.

Um membro de uma DL pode ser outra DL como mostra a Figura II.11. Neste caso a mensagem é transmitida do ponto de expansão da DL pai (e.g., DL1 da Figura II.11) para o ponto de expansão da DL membro para posterior expansão.

Durante a expansão de uma DL aninhada, a identificação da DL pai ao invés da identificação do originador da mensagem, é verificada para permissão de submissão à DL membro. O MTS possui mecanismos para controle de recursão de modo que se uma DL é direta ou indiretamente um membro dela mesma ela não circule infinitamente no MTS.

II.3.7.2 Entrega e Notificações

Na entrega da mensagem, o destinatário será avisado que recebeu a mensagem como um membro de uma DL e através de que DL ele recebeu a mensagem.

Notificações de entrega e não-entrega podem ser geradas no ponto de expansão da DL e na entrega ao destinatário final. Quando uma mensagem vinda via DL gera uma notificação, esta é enviada ao ponto de expansão da DL. O ponto de expansão, dependendo da política da lista, redestinará ao proprietário da lista e/ou ao originador da mensagem, como mostra a Figura II.12.

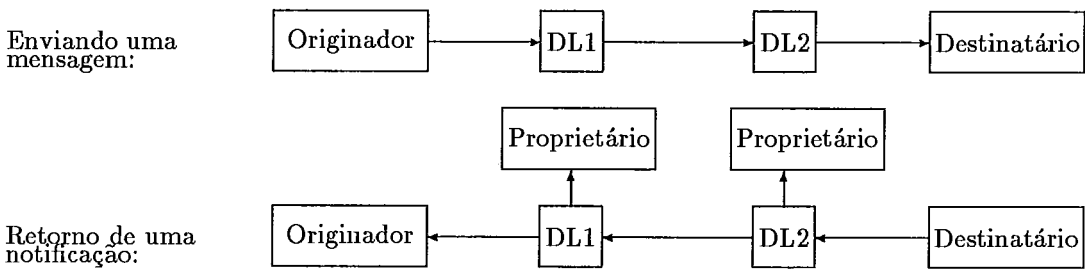


Figura II.12: Entrega e Notificações para DLs

Um MTA pode ou não prover diferentes políticas de manipulação de DLs. Tais políticas determinarão para onde devem ser enviadas as notificações geradas no momento de entrega da mensagem aos membros da lista. Se a política determina que as notificações sejam enviadas somente ao proprietário da lista, então o originador receberá notificações, se houver solicitado, somente durante a expansão da DL. Para isto, o MTS deletará os pedidos de notificação do originador do envelope da mensagem, enquanto realiza a expansão da DL.

II.3.8 Segurança no MHS

A natureza distribuída do MHS torna desejável a existência de mecanismos de proteção de ameaças contra a sua segurança em diversos aspectos tais como: acesso a usuários não cadastrados no sistema, alteração de mensagens durante a transferência, interceptação de mensagens, alteração de dados relativos à roteamento no MHS, repúdio de mensagens (isto pode ocorrer quando um dos participantes da comunicação nega seu envolvimento). Este fato pode ter sérias implicações em transações financeiras.

Serviços de segurança podem ser providos estendendo-se a capacidade dos componentes do MHS de forma a incluir mecanismos de segurança. Existem dois

aspectos de segurança no MHS: o gerenciamento de acesso seguro e a troca segura de mensagens.

O gerenciamento de acesso seguro está relacionado à segurança no estabelecimento de associações entre os componentes do sistema e o estabelecimento de parâmetros de segurança para a associação de forma a prover proteção de recursos contra uso não autorizado. Isto pode ser aplicado à qualquer par de componentes do MHS: UA/MTA, MTA/MTA, MS/MTA, etc.

A troca segura de mensagens está relacionada à aplicação de mecanismo de segurança para proteção de mensagens de acordo com uma política de segurança definida. Isto inclui serviços que habilitam vários componentes do MHS a verificar a origem de mensagens e a integridade de seus conteúdos, e serviços que previnem acesso não autorizado ao conteúdo das mensagens.

Os serviços de segurança são suportados através do envelope das mensagens transferidas no MTS. Uma descrição completa do modelo de segurança do MHS pode ser encontrada na seção 10 da Recomendação X.402.

Muitas das técnicas utilizadas no modelo de segurança do MHS de baseiam em mecanismos de cifragem. Os serviços de segurança permitem flexibilidade na escolha de algoritmos, entretanto, em alguns casos somente o uso de cifragem assimétrica foi descrito nas Recomendações.

Os aspectos do esquema de gerenciamento de chave assimétrica que suporta os serviços de segurança são providos no modelo de autenticação do Sistema de Diretório descrito pela Recomendação X.509 [4]. O diretório armazena cópias das chaves públicas para os usuários do MHS, e pode ser usado para prover autenticação e facilitar a troca de chaves para uso em mecanismo de confidencialidade e integridade de dados.

II.4 Serviços do MHS

Os usuários do MHS dispõem de uma coleção de recursos e facilidades para enviar e receber mensagens de forma eficiente e expressiva. Esta coleção de recursos oferecidos pelo MHS reflete a capacidade funcional do sistema. Os Serviços do MHS são proporcionados através de duas classes de serviços, a saber: o Serviço de Transferência de Mensagem (Serviço MT) e o Serviço de Mensagens Interpessoais (Serviço IPM).

II.4.1 Serviço de Transferência de Mensagens

O Serviço Básico de MT permite um UA submeter e receber mensagens, sendo informado no caso em que a mensagem não pode ser entregue ao UA destinatário. Cada mensagem é identificada sem ambiguidade pelo UA.

As Facilidades Opcionais de MT oferecem recursos adicionais ao UA que aumentam a capacidade funcional do MHS. Esta classe de serviços permite a submissão de uma mensagem a vários destinatários, ou seja, a submissão de uma mensagem com vários endereços no envelope. O MTS criará cópias da mensagem para que esta possa ser entregue a cada um dos destinatários mencionados no envelope. Para cada destinatário, o originador pode solicitar que o MTS não revele os demais destinatários.

O MTS oferece ainda o serviço opcional de "conversão de conteúdo". Através deste serviço, é possível, por exemplo, que um destinatário que somente possua equipamentos Fax receba uma mensagem que foi originariamente codificada como uma mensagem de texto. A conversão é implicitamente determinada pelas capacidades do equipamento do originador e do destinatário, entretanto, o originador pode também solicitar uma determinada conversão explicitamente, ou até mesmo proibir qualquer tipo de conversão. A proibição de conversão pode fazer que uma mensagem não possa ser entregue.

O UA pode indicar ao MTA as suas capacidades com respeito ao recebimento de mensagens, tais como os tipos de codificação da informação que podem ser aceitos nas mensagens entregues a ele, tamanho máximo e prioridade mínima de mensagens.

É fornecido em cada mensagem entregue ao UA o tipo de conteúdo, os tipos originais de codificação da informação da mensagem, uma indicação de conversões que possam ter sido realizadas, os tipos resultantes de codificação da informação e as datas de submissão e entrega da mensagem ao MTS.

Os serviços opcionais de MT permitem também que UA submeta mensagens ao MTA para entrega pré-datada ou cancele esta entrega. O UA pode solicitar ao MTA que retenha as mensagens e/ou notificações para entrega até última ordem. Quanto a entrega de mensagens, é oferecido ao UA a possibilidade de ser informado da entrega da mensagem. O UA pode ainda estabelecer prioridade para as mensagens submetidas ao MTS.

No caso de impossibilidade de entrega, o UA pode solicitar o retorno de mensagens, permitir que a mensagem seja entregue a outro UA dentro de certos critérios, ou proibir o MTA de notifica-lo.

Existe ainda, o serviço de sondagem. A sondagem é basicamente um envelope sem conteúdo, através do qual pode-se verificar, antes da submissão, se uma mensagem em particular poderia ser entregue com sucesso. Um possível tipo de utilização deste serviço seria o teste de transmissão de uma mensagem muito longa a um destinatário para o qual o originador em questão ainda não tenha enviado qualquer mensagem. O sistema retorna uma notificação de entrega ou não-entrega relativo à sondagem, indicando se uma mensagem com tais parâmetros poderia ou não ser entregue ao(s) destinatário(s) em questão.

Com as novas Recomendações muitos novos serviços foram adicionados [2]. Entre eles, dois importantes serviços foram definidos: as Listas de Distribuição e os Serviços de Segurança.

Através das Listas de Distribuição (DL), pode-se enviar uma mensagem à um grupo de destinatários especificando-se apenas o nome do grupo (endereço da lista). O endereço de uma lista de distribuição (DL) não difere daquele estabelecido para destinatários normais. Entretanto, quando a mensagem chega ao MTA responsável pela DL em questão, o MTA expande a DL, isto é, substitui o endereço daquela lista pelos endereços de seus membros.

DLs podem ser membros de outras DLs e as Recomendações definem um algoritmo para detecção de loops em DLs.

Os serviços de Segurança no MHS referem-se ao cuidado contra inspeção casual ou proposital de uma mensagem por terceiros. O MHS define um sistema de autenticação baseado em sistema de chave de cifragem pública assimétrica, onde originador e destinatário entram em acordo sobre uma chave para cifragem e decifragem de uma mensagem. Os MTAs devem ser incluídos neste contexto de segurança, já que eles necessitam de informação no envelope para rotear e executar serviços.

II.4.2 O Serviço de Mensagens Interpessoais

O Serviço de Básico de IPM possibilita um usuário enviar e receber mensagens-IP. Para isto, este serviço engloba o Serviço Básico de MT e ainda, permite que IPM-UAs cooperantes estabeleçam um identificador único para cada mensagem-IP transmitida ou recebida, e que a natureza e os atributos do corpo da mensagem-IP sejam transferidos junto com o corpo desta mensagem. O usuário pode ainda enviar uma mensagem-IP cujo corpo contém varias partes de natureza e tipos distintos, entre os quais textos cifrados, e futuramente voz.

As Facilidades Opcionais de IPM visam tornar a troca de mensagens mais ex-

pressiva e objetiva a nível da comunicação interpessoal. Esta classe de serviço engloba as Facilidades Opcionais de MT, permite que o usuário ao enviar a mensagem, se identifique de forma amigável ao destinatário da mensagem, designe assunto, importância, data de obsolescência, grau de segurança da mensagem e associe esta mensagem com outra(s) mensagens-IP. O usuário pode ainda solicitar que o destinatário da mensagem envie uma resposta desta mensagem a ele ou a outro(s) usuário(s), que seja informado em caso de recebimento ou no caso de impossibilidade de recebimento da mensagem por parte do(s) destinatário(s), e que seu UA redestine as mensagens recebidas a outro usuário. As mensagens redestinadas podem ser enviadas no corpo de outra mensagem-IP.

Um UA deve executar funções locais para seu usuário além dos serviços padronizados. Os serviços locais não requerem cooperação entre as entidades do MHS e, desta forma, não são objeto de padronização.

Os novos de serviços de IPM, definidos nas Recomendações X.400 1988, permitem que o usuário indique os idiomas utilizados em sua mensagem e que um originador indique que uma determinada mensagem é uma cópia incompleta de uma outra mensagem com a mesma identificação, indicando que uma ou mais partes de corpo e/ou campos do cabeçalho da mensagem estão ausentes.

Com a extensão das Recomendações X.400 em 1988, duas novas classes de serviços opcionais foram acrescentadas: o Serviço de Entrega Física (Serviço de PD) e o Serviço de Armazenamento de Mensagens (Serviço de MS).

O Serviço de PD permitem que o usuário do MHS envie mensagens à destinatários que as receberão via algum sistema de entrega física, tal como o serviço postal convencional.

O Serviço de MS permite o armazenamento e gerenciamento de mensagens entregues pelo MTS.

A descrição completa dos serviços do MHS está fora do escopo desta tese, mas pode ser encontrada no Anexo B da Recomendação X.400 [3]. A Tabela II.1 lista os novos serviços acrescentados pela série X.400-88.

Serviço	MT	IPM	PD	MS
Additional physical rendition			X	
Basic physical rendition			X	
Content confidentiality	X			
Content integrity	X			
Conversion prohibition in case of loss of information	X			
Counter collection			X	
Counter collection with advice			X	
Delivery via Bureaufax service			X	
Designation of recipient by directory name	X			
DL expansion history indication	X			
DL expansion prohibited	X			
EMS (express mail service)			X	
Incomplete copy indication		X		
Language indication		X		
Latest delivery designation	X			
Message flow confidentiality	X			
Message origin authentication	X			
Message security labelling	X			
Message sequence integrity	X			
Non-repudiation of delivery	X			
Non-repudiation of origin	X			
Non-repudiation of submission	X			
Ordinary mail			X	
Originator requested alternated recipient	X			
Physical delivery notification by MHS			X	
Physical delivery notification by PDS			X	
Physical forwarding allowed			X	
Physical forwarding prohibited			X	
Probe origin authentication	X			
Proof of delivery	X			
Proof of submission	X			
Redirection disallowed by originator	X			
Redirection of incoming messages	X			
Registered mail			X	
Registered mail to address in person			X	
Report origin authentication	X			
Request for forwarding address			X	
Requested delivery method	X			
Restricted delivery	X			
Secure access management	X			
Special delivery			X	
Stored message alert				X
Stored message auto-forward				X
Stored message deletion				X
Stored message fetching				X
Stored message listing				X
Stored message summary				X
Undeliverable mail with return of physical message			X	
Use of distribution list	X			
User/UA capabilities registration	X			

Tabela II.1: Os Novos Serviços das Recomendações X.400 1988

II.5 Protocolos e Endereçamento no MHS

II.5.1 A Estrutura da Camada de Aplicação

Nas recomendações de 1984, por analogia com os princípios do Modelo OSI, o serviço de manipulação de mensagens foi definido em duas camadas: a camada de transferência de mensagens (MTL) e a camada de agente usuário (UAL). Antes da publicação das recomendações de 1988, a ISO definiu uma estrutura para o nível de aplicação na qual os chamados "elementos de serviço" fornecem serviços comuns ou específicos às aplicações. Os elementos de serviço comuns são: o Elemento de Serviço para Controle de Associação (ACSE), o Elemento de Serviço para Operações Remotas (ROSE) e o Elemento de Serviço para Transferência Confiável (RTSE).

O ACSE é o meio através do qual as associações de aplicações são estabelecidas, terminadas e gerenciadas. O ACSE é definido na recomendação X.218 do CCITT e na norma ISO/IEC 9072 da ISO [23,29].

O ROSE é o elemento de serviço comum da camada de aplicação que suporta aplicações interativas usando mecanismo pedido/resposta. O ROSE é definido na recomendação X.217 do CCITT e na norma ISO/IEC 9072 da ISO [25,30].

O RTSE é o elemento de serviço comum da camada de aplicação que provê transferência de dados entre aplicações de forma confiável realizando recuperação de erros e minimizando a quantidade de retransmissões. O RTSE é definido na recomendação X.218 do CCITT e na norma ISO/IEC 9066 [24,31].

Com esta estruturação da camada de aplicação, a definição em níveis do serviço do MHS foi removida das recomendações X.400. Nas novas recomendações os protocolos do MHS são definidos utilizando a estrutura em elementos de serviço definida para a camada de aplicação, conforme será visto a seguir.

II.5.2 Os Protocolos do MHS

A Recomendações X.400 1984 definiram três tipos distintos de protocolos para realizar as interações necessárias entre suas entidades (Figura II.2). São eles os protocolos P1, P2 e P3 definidos abaixo:

O Protocolo de Transferência de Mensagens, comumente chamado Protocolo P1, governa a interação de transferência entre MTAs, provendo a operação distribuída do MTS.

O Protocolo de Mensagens Interpessoais, comumente chamado Protocolo P2,

define a sintaxe e a semântica das mensagens trocadas entre IPM-UAs.

O Protocolo de Acesso ao MTS, comumente Protocolo P3, governa o acesso ao MTS, provendo as interações de submissão e entrega de mensagens. Para satisfazer as exigências da modelagem em camadas citada acima e ilustrada na Figura II.13, uma entidade considerada sem significância (Submission and Delivery Entity, SDE) foi introduzida no modelo para suporte ao Protocolo P3.

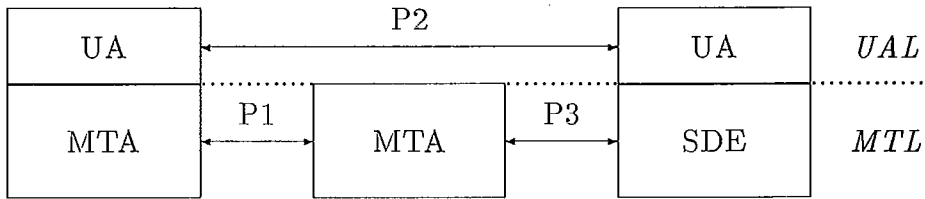


Figura II.13: Modelagem em Camadas do MHS-84

As Recomendações X.400 1988 eliminaram esta modelagem em camadas, aderiram à nova estrutura da camada de Aplicação, estenderam estes protocolos e adicionaram mais um: o Protocolo P7.

O Protocolo de Acesso ao MS, comumente chamado Protocolo P7, governa o acesso ao MS, provendo as interações de recuperação e submissão indireta de mensagens.

Em um ambiente OSI, a comunicação entre processos de aplicação é representado em termos da comunicação entre um par de Entidades de Aplicação (AEs) utilizando o serviço de apresentação. A funcionalidade de uma AE é fatorada em um conjunto de um ou mais Elementos de Serviço de Aplicação (ASEs). A interação entre AEs é descrita em termos do uso dos serviços providos pelas ASEs.

Seguindo esta estrutura da camada de aplicação, as Recomendações de 1988 dividiram as funções dos protocolos de MHS em ASEs, conforme será descrito a seguir.

Para o protocolo P1, os serviços de transferência do MTS são suportados pelo Elemento de Serviço de Transferência de Mensagem (Message Transfer Service Element - MTSE), que suporta a interação de transferência de mensagens entre MTAs.

A Figura II.14 mostra a estrutura de dois MTAs que se comunicam segundo o protocolo P1. O Elemento de Serviço de Transferência de Mensagem (MTSE), que é um elemento de serviço específico da aplicação MHS, utiliza o RTSE, que por sua vez utiliza o ACSE. Tanto RTSE como ACSE utilizam o serviço da camada de apresentação, a camada 6 do Modelo OSI.

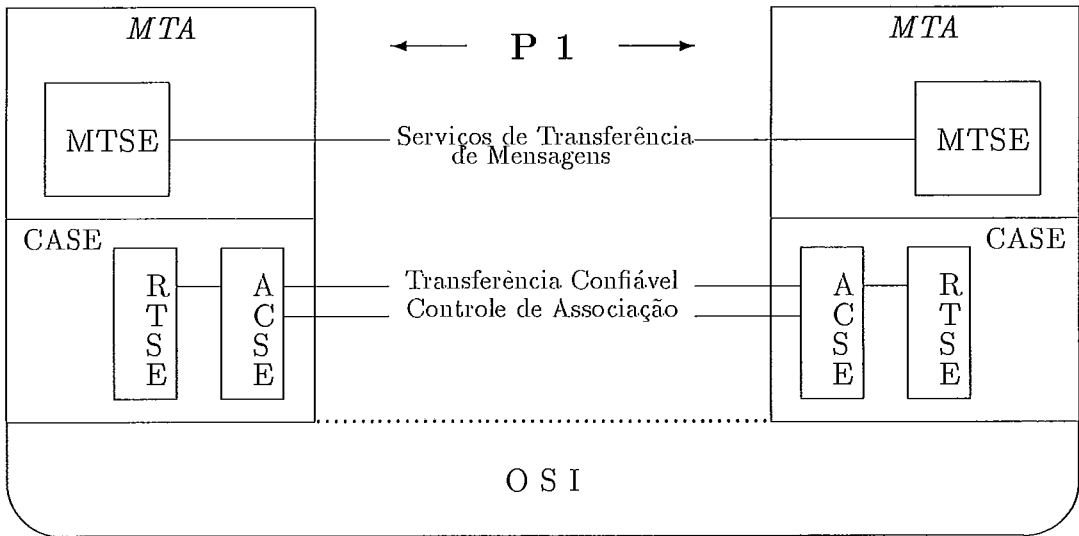


Figura II.14: O Protocolo P1

Para o protocolo P2 não são definidos elementos de serviço deste tipo já que este protocolo não define interação entre entidades, e sim a sintaxe e semântica de mensagens interpessoais.

Para o protocolo P3, o acesso ao MTS é suportado por três elementos de serviço, cada um suportando um tipo de interação entre o MTS e seu usuário (UA, MS ou AU). O Elemento de Serviço de Submissão de Mensagem (Message Submission Service Element - MSSE) suporta a interação de submissão de mensagens. O Elemento de Serviço de Entrega de Mensagem (Message Delivery Service Element - MDSE) suporta a interação de entrega de mensagens. O Elemento de Serviço de Administração de Mensagem (Message Administration Service Element - MASE) suporta as interações de gerenciamento de informações do usuário no MTS.

A Figura II.15 mostra a estrutura do protocolo P3. Nesta Figura, o usuário do MTA, que pode ser o UA, o MS ou a AU, faz acesso aos serviços do MTA localizado em um sistema aberto remoto. Os elementos de serviço específicos da aplicação MHS, MSSE, MDSE e MASE, utilizam o ROSE, e o ACSE. O RTSE pode ser opcionalmente utilizado, caso seja desejada transferência confiável de dados. Estes elementos de serviço comuns, por sua vez, utilizam o serviço da camada de apresentação, a camada 6 do Modelo OSI.

Para o protocolo P7, o acesso ao MS é suportado por três elementos de serviço, cada um suportando um tipo de interação entre o MS e seu usuário (UA). O Elemento de Serviço de Recuperação de Mensagem (Message Retrieval Service Element - MRSE) suporta a interação de recuperação de mensagens. O Elemento de Serviço de Submissão de Mensagem (Message Submission Service Element -

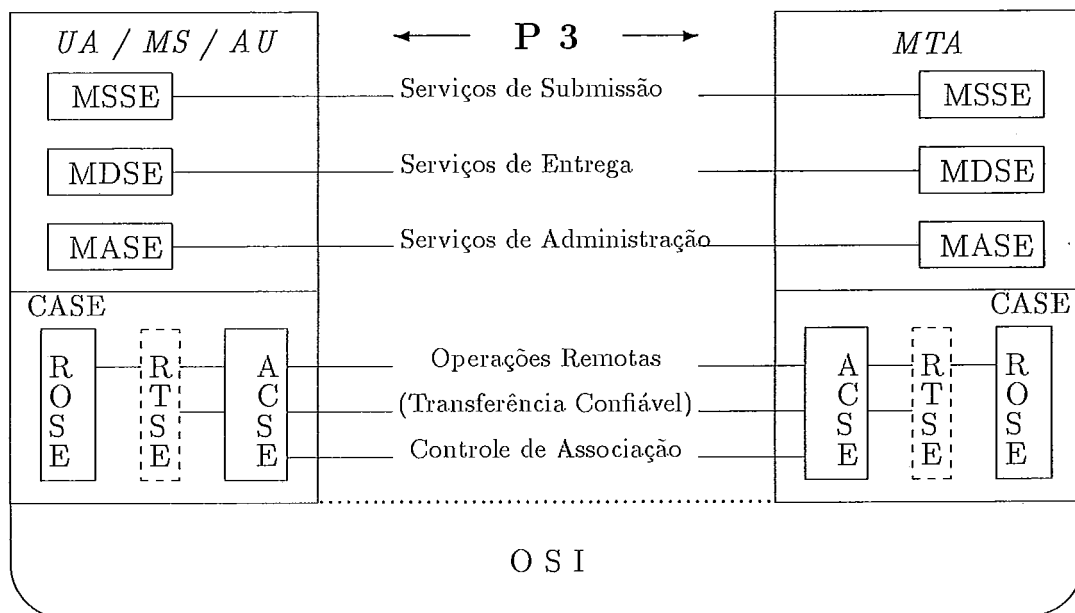


Figura II.15: O Protocolo P3

MSSE) suporta a interação de submissão indireta de mensagens. O Elemento de Serviço de Administração de Mensagem (Message Administration Service Element - MASE) suporta as interações de gerenciamento de informações do usuário no MTS.

A Figura II.16 mostra a estrutura do protocolo P7. Nesta Figura, o usuário do MS, o UA, faz acesso aos serviços do MS localizado em um sistema aberto remoto. Os elementos de serviço específicos da aplicação MHS, MSSE, MRSE e MASE, utilizam o ROSE, e o ACSE. O RTSE pode ser opcionalmente utilizado, caso seja desejada transferência confiável de dados. Estes elementos de serviço comuns, por sua vez, utilizam o serviço da camada de apresentação, a camada 6 do Modelo OSI.

Quanto ao suporte das camadas inferiores do Modelo OSI, para o MHS, é necessário o subconjunto do serviço de Sessão X.215 do CCITT, que corresponde ao "Basic Activity Subset (BAS)" do serviço de Sessão ISO. Quanto ao nível de Transporte, é necessário somente a Classe 0 (zero). Para as camadas de Rede, Enlace e Física, utiliza-se a padronização X.25 do CCITT [5], que é amplamente adotada para suportar as redes públicas de comunicação de dados.

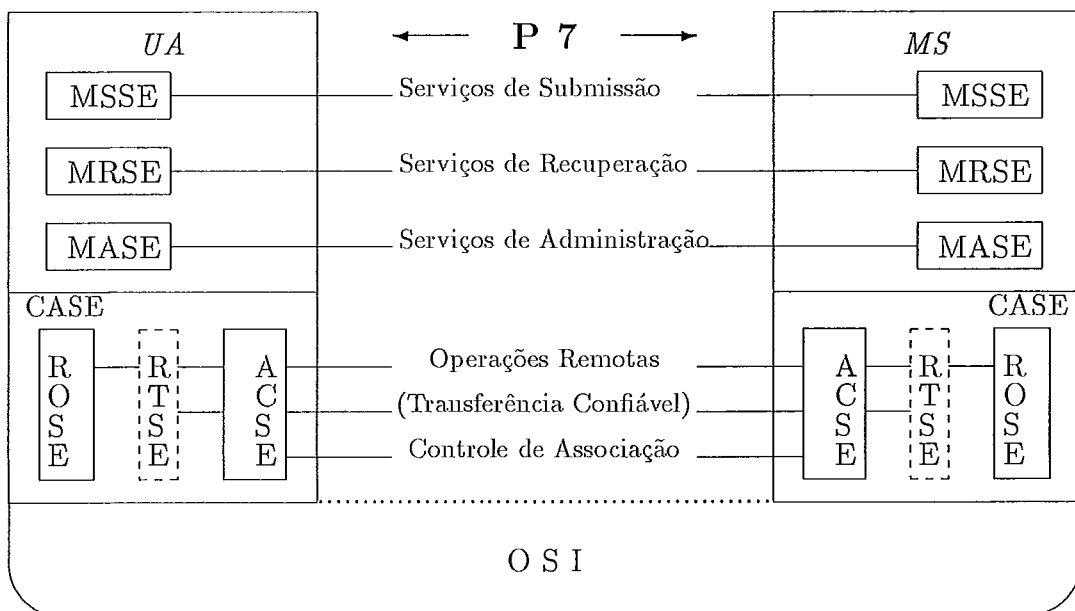


Figura II.16: O Protocolo P7

II.5.3 Domínios de Gerenciamento

A tarefa de administrar e manter um sistema do porte do MHS é facilitada se o sistema é dividido em partes menores. Para este propósito, as Recomendações X.400 introduzem o conceito de "Domínio de Gerenciamento" (MD). No MHS, um MD é definido por um conjunto formado por pelo menos um MTA e zero ou mais UAs.

Existem dois tipos de MD: O MD Administrativo ou Público (ADMD), que é gerenciado por empresas públicas (e.g., Embratel) e o MD Privado (PRMD) que é gerenciado por organizações tais como companhias privadas e universidades.

II.5.4 Endereçamento no MHS

No MHS, as principais entidades que requerem identificação são seus usuários (originadores e destinatários de mensagens), e estes são identificados pelo Nome O/R (Originator/Recipient Name - O/R Name).

As Recomendações de 1984 definem o Nome O/R como uma lista de atributos que identificam unicamente uma entidade. Se os tipos de atributos escolhidos são tais que o MHS pode localizar o usuário, então o nome O/R é também um "endereço O/R". Desta forma, os atributos do endereço O/R estão relacionados à arquitetura do MHS. No MHS-84 todo nome O/R é um endereço O/R. Este endereço é utilizado para rotear e entregar uma mensagem ao seu destinatário.

Country (C) =	BR
Administrative Domain (A) =	EBT
Privative Domain (P) =	RNP
Organization (O) =	UFRJ
Organizational Unit (OU) =	NCE
Organizational Unit (OU) =	VAX1
Surname (S) =	SUZAN

Figura II.17: Exemplo de um endereço O/R

Normalmente, um endereço O/R inclui pelo menos um nome de Domínio de Gerenciamento Administrativo (ADMD) e de Domínio de Gerenciamento Privado (PRMD), e possivelmente uma identificação do usuário única no âmbito do domínio de gerenciamento.

As Recomendações X.400 definem várias formas e variantes de endereços O/R: A forma 1, variante 1 definida nas Recomendações representa a forma mais frequente:

Nome do País - Código do país, padronizado pela ISO.

Nome do ADMD - O nome do ADMD ao qual o usuário ou PRMD está conectado ou associado, respectivamente.

Nome do PRMD - Opcional. Nome do PRMD ao qual o usuário pertence.

Nome da Organização - Opcional. O nome da organização, por exemplo, a companhia ou universidade.

Nomes das Unidades Organizacionais - Opcional. Uma lista de nomes de subunidades da organização, normalmente refletindo a estrutura da organização.

Nome Pessoal - Opcional. O nome do usuário ao qual o UA representa.

Atributos Definidos no Domínio - Opcional. Estes atributos podem ser usados para qualquer propósito internamente ao MD, por exemplo, para acomodar outros componentes de endereçamento que não podem ser facilmente mapeados em outros atributos padrão.

Nas Recomendações X.400 de 1988, o nome O/R foi estendido, e é constituído de um "Nome de Diretório" e/ou um endereço O/R. Com este nome de diretório,

o endereço O/R do usuário pode ser obtido pelo uso do Sistema de Diretório (DS) padronizado pelas Recomendações X.500. O nome de diretório é utilizado para localizar o endereço O/R no Sistema de Diretório; o endereço é então retornado pelo DS. A intenção é tornar a identificação de usuários mais amigável.

No MHS-88, um dos dois componentes ou ambos podem ser utilizados. Se somente o nome de diretório é usado, o MHS acessará o DS para tentar obter o endereço O/R do usuário. Se o nome de diretório não está presente, o endereço O/R é utilizado diretamente. Se ambos são fornecidos, o MHS usará o endereço O/R mas transportará ambos ao destinatário, e se o endereço O/R é inválido, o MHS usará o nome de diretório para obter o endereço correto.

II.6 Comentários

Neste capítulo foram descritos o modelo, os elementos e a estrutura do MHS, tendo-se em vista a evolução dos padrões X.400.

A série de recomendações X.400 1988 diferem significativamente das recomendações X.400 1984. Apesar do modelo funcional no qual as definições são baseadas não ter sofrido alterações drásticas, foram necessárias extensões aos serviços e protocolos para que a funcionalidade dos serviços de manipulação de mensagens fosse aumentada.

Vários novos serviços foram somados às recomendações, muitos dos quais refletem a nova funcionalidade (e.g., Entrega Física, Listas de Distribuição, Armazenamento Remoto de Mensagens).

A nível dos protocolos, do ponto de vista externo, a mudança se inicia na estrutura da camada de aplicação. Esta modificação deve-se ao fato que o CCITT ter acompanhado, nestas novas recomendações, o trabalho da ISO na definição da estrutura da camada de aplicação. As extensões feitas a cada protocolo, do ponto de vista interno, serão analisadas no Capítulo IV.

Capítulo III

Especificação Formal do MHS

III.1 Introdução

A tarefa de especificar sistemas distribuídos é mais complexa do que a especificação de sistemas "centralizados" clássicos. As dificuldades estão relacionadas à necessidade de descrever vários componentes que podem cooperar entre si e executar em paralelo.

A intercooperação entre os diversos componentes de um sistema distribuído é controlada por protocolos usados na comunicação entre estes componentes. Estes componentes devem ser compatíveis entre si, isto é, devem satisfazer aos protocolos de comunicação definidos. Para facilitar a implementação de componentes compatíveis no sistema, é importante ter uma definição precisa dos protocolos de comunicação utilizados. A especificação de protocolos é utilizada para este propósito.

A coleção de padrões para protocolos e serviços de comunicação desenvolvidos para Interconexão de Sistemas Abertos destina-se a permitir a intercooperação de sistemas heterogêneos para uma variedade de aplicações. Neste contexto, as especificações de protocolos e serviços são de particular importância, já que eles representam a base para a implementação e teste de sistemas OSI compatíveis. As especificações fornecidas pelos padrões são geralmente escritas em linguagem natural, às vezes acrescidas de algum formalismo. Faz-se então necessário, o estabelecimento de Técnicas de Descrição Formal (FDT) para descrição precisa de ambientes OSI.

A seção III.2 deste capítulo discute brevemente aspectos da especificação de protocolos no contexto OSI. A seção III.3 apresenta a linguagem Estelle, utilizada para a modelagem do MHS neste trabalho, e apresenta as técnicas utilizadas pelos padrões X.400, a saber: ASN.1 e ROS. A seção III.4 descreve a modelagem do



Figura III.1: Especificação de Protocolos OSI

MHS desenvolvida neste trabalho utilizando-se a linguagem Estelle. A seção III.5 descreve o Sistema de Auxílio utilizado durante a modelagem. A seção III.6 tece comentários e propõe extensões à linguagem Estelle.

III.2 Aspectos Relativos à Especificação de Protocolos

No contexto OSI, três objetos precisam ser especificados: o serviço de comunicação, o protocolo de comunicação, e as interfaces.

O serviço de comunicação especifica os aspectos relacionados às propriedades locais do ponto de acesso do serviço em questão e às propriedades globais relacionadas a troca de interações nos diferentes pontos de acesso.

O protocolo a ser seguido pelas entidades de uma dada camada é definido pelo conjunto de regras de diálogo e formato das mensagens que caracterizam a comunicação entre as entidades da camada.

As entidades (N), para fornecer um serviço à camada superior, cooperam segundo o protocolo N e utilizam o serviço (N-1) para troca de mensagens. A Figura III.1 mostra esta interação.

A interface é o meio através do qual o serviço de comunicação é provido aos usuários no ponto de acesso em questão.

As especificações para o serviço e protocolo de comunicação têm relevância global. Por outro lado, as propriedades da interface poderiam ser diferentes para cada ponto de acesso; portanto, em cada caso estas propriedades devem ser consistentes com as propriedades locais do serviço utilizado.

As interações primitivas entre uma entidade de protocolo que provê um serviço e o usuário do serviço são chamadas *primitivas de serviço* [43]. Elas são invocadas nos pontos de acesso entre o usuário e o serviço de comunicação. Cada primitiva de

serviço geralmente contém vários parâmetros que são trocados entre as entidades envolvidas. Alguns destes parâmetros podem representar os dados que usuário deseja transmitir ao usuário remoto através do serviço de comunicação. Os dados trocados entre entidades pares de protocolo são chamadas *Unidades de Dados de Protocolo* (PDU). É importante notar que, freqüentemente, as propriedades abstratas de uma entidade de protocolo são descritas em termos da troca de PDUs entre as entidades pares e as primitivas de serviço com o usuário.

A especificação do serviço, protocolo de comunicação, ou outro componente do sistema deve definir todos os aspectos de comportamento relevantes, a saber [43]:

- (a) *Ordem das Interações*: este aspecto define a ordem na qual as diferentes interações podem ser executadas. Isto inclui a distinção entre os diferentes tipos de primitivas de serviço e PDUs, e a ordem temporal permitida para estas interações;
- (b) *Valores Possíveis para Parâmetros*: este aspecto define a extensão de valores admissíveis para os parâmetros das primitivas e PDUs;
- (c) *Regras para Interpretação e Seleção de Valores para Parâmetros*: este aspecto representa um refinamento do aspecto (b); ele define os valores de parâmetros permitidos como função das interações anteriores executadas pela entidade; e define para cada interação recebida, o significado dos parâmetros recebidos para subsequente processamento pela entidade;
- (d) *Codificação de PDUs*: enquanto o aspecto (c) define os valores possíveis dos parâmetros das PDUs, este aspecto define o formato no qual as PDUs e seus parâmetros são codificados para transmissão através do serviço de comunicação;
- (e) *Propriedades de Vivacidade*: definem as regras que devem ser seguidas para que todas as interações do protocolo ocorram;
- (f) *Propriedades de Tempo Real*: enquanto o aspecto (e) estabelece que certas interações ocorrerão (medidas qualitativas), as propriedades de tempo real fornecem medidas quantitativas. Para serviços e protocolos de comunicação, estas propriedades estão geralmente relacionadas à "throughput" alcançável e a atrasos na comunicação. Outras propriedades de tempo real estão relacionadas à confiabilidade, e à probabilidade de rejeição de pedidos de serviços.

Os demais aspectos devem ser deixados indefinidos, de forma que possam ser escolhidos de acordo com as exigências particulares de cada implementação.

Um aspecto importante é a escolha da linguagem de especificação a ser utilizada. A maioria das especificações são escritas em linguagem natural, entretanto, em muitos casos é vantajoso definir certos aspectos de comportamento de maneira formal. Especificações em linguagem natural freqüentemente contêm ambigüidades e dificultam a verificação da completude e da consistência. Além disso, não permitem o uso de ferramentas para automatização no desenvolvimento do sistema considerado.

A ISO e o CCITT desenvolveram três Técnicas de Descrição Formal (FDT) para aplicação em OSI: Estelle (ISO), SDL (CCITT) e LOTOS (OSI). Este capítulo descreve a modelagem de todo o MHS, com exceção das Unidades de Acesso, utilizando a linguagem Estelle [37].

A modelagem das Unidades de Acesso está fora do escopo deste trabalho por exigir conhecimentos específicos de cada serviço telemático ao qual a Unidade de Acesso se aplica. Porém, o protocolo do MHS através do qual estas Unidades de Acesso utilizam os serviços de manipulação de mensagens é modelado nesta tese.

Através da linguagem Estelle é descrito um MHS completamente distribuído, ou seja, onde as entidades de acesso (UA), armazenamento (MS) e transferência (MTA) encontram-se em sistemas abertos distintos; além disto, foram também modelados os Elementos de Serviço Comuns da Camada de Aplicação utilizando a mesma técnica, e permitindo, assim, uma visão global do funcionamento da camada de aplicação.

Esta modelagem permitirá a definição precisa do MHS, a aplicação de ferramentas automatizadas para a validação, implementação e futuros testes de conformidade do MHS e facilitará possíveis alterações de parâmetros de projeto.

Antes, entretanto, que a modelagem seja descrita, a linguagem Estelle, utilizada para esta modelagem, e as técnicas de descrição utilizadas pelas Recomendações X.400 (ASN.1 e ROS) são apresentadas.

III.3 As Linguagens Utilizadas

III.3.1 Estelle

Estelle (Extended State Transition Language) [37], é uma Técnica de Descrição Formal (FDT) desenvolvida pela ISO para a especificação de sistemas distribuídos, protocolos de comunicação e, em particular, os padrões relativos ao Modelo OSI.

Estelle é baseada em um modelo de transição que combina os tipos de notação de Máquina de Estados Finita (estados, interações e transições) e linguagem de

programação Pascal (variáveis, parâmetros e prioridades).

Quanto à arquitetura, em Estelle uma especificação é composta de um conjunto de módulos (*module*). Cada módulo é apresentado como uma "caixa preta" com portas de entrada e saída (*interaction point*). Canais (*channel*) de comunicação bidirecionais podem conectar (*connect*) módulos através de suas portas.

Um módulo pode ser refinado em sub-módulos, definindo-se um parentesco e uma estrutura hierárquica entre os componentes da especificação. Canais internos podem conectar sub-módulos e as portas dos sub-módulos filhos podem ser vinculadas (*attach*) às portas do módulo pai.

A estrutura interna de um módulo bem como as conexões e as vinculações de seus sub-módulos podem mudar dinamicamente. O comportamento desse módulo é descrito através de um conjunto de transições.

Quanto à estruturação, uma especificação Estelle descreve:

- Uma estrutura de módulos aninhados;
- Como essa estrutura é "inicializada";
- O comportamento interno dos módulos.

Um módulo pode ser ativo ou não. Ele será ativo se possuir transições e inativo caso contrário. Cada módulo pode ou não possuir um dos seguintes atributos: *systemprocess*, *systemactivity*, *process* ou *activity*, com princípios definidos para a atribuição dos módulos [37].

Na atribuição dos módulos deve ser levado em consideração, além da estruturação, o tipo de paralelismo desejado. Neste caso, a linguagem define princípios que relacionam os tipos de módulo com os tipos de paralelismo [37].

O tempo de execução das transições é variável, dependendo da implementação. A presença da cláusula *delay* na especificação de uma transição retarda a sua execução. Dois valores (mínimo e máximo) de atraso podem ser associados à cláusula *delay*.

Quanto à comunicação entre os módulos, estes trocam primitivas de comunicação (interações). Um módulo pode enviar uma primitiva, por uma de suas portas, a um outro módulo, desde que ambos estejam conectados através de um canal.

Uma primitiva, ao ser recebida por um módulo (através de uma de suas portas), é anexada a uma fila do tipo FIFO (First In - First Out). Uma fila FIFO pode estar associada a uma única porta (*individual queue*) ou a um conjunto de portas (*common queue*).

Módulos podem exportar variáveis. Um módulo pai pode ter acesso às variáveis exportadas pelos seus filhos.

Quanto à sintaxe da linguagem, todos os objetos manipulados são tipificados. Estelle adiciona aos tipos e variáveis Pascal os seus próprios objetos:

- Variáveis de estado principal (controle), que são do tipo enumerada;
- Variáveis módulo, onde cada variável módulo é do tipo módulo;
- Pontos de interação (portas), onde cada ponto de interação é do tipo canal.

Definição de um canal:

O canal é uma construção que permite desvincular a especificação das primitivas de comunicação da especificação dos módulos, e é definido como mostrado abaixo:

```
channel Ident_Canal (Papel1, Papel2);  
by Papel1: Ident_Interação (Lista_Parâmetros);  
    :  
    :  
    Ident_Interação (Lista_Parâmetros);  
by Papel2: Ident_Interação (Lista_Parâmetros);  
    :  
    :  
    Ident_Interação (List_Parâmetros);
```

A especificação acima indica que os módulos a serem conectados através de suas portas às extremidades desse canal desempenharão as funções Papel1 e Papel2. O módulo que desempenhar a função Papel1 pode emitir as primitivas definidas em Papel1 e deve receber as primitivas definidas em Papel2. O contrário deve ocorrer com o módulo que desempenhar a função Papel2.

Definição de um Módulo:

A especificação de um módulo é constituída de duas partes: a definição do cabeçalho do módulo, e a definição do corpo do módulo, sendo que diferentes módulos podem ser atribuídos ao mesmo cabeçalho.

O cabeçalho de um módulo representa o seu nível mais alto de abstração e sua definição é baseada na descrição de seus pontos de interação *ip*. Esse cabeçalho define um tipo módulo. Variáveis de um tipo módulo representam cópias desses tipo e todas possuem a mesma visibilidade externa. A definição de um módulo em Estelle é mostrada abaixo.

```
module Ident_Cabeça Atributo_Módulo (Lista_Parâmetros);
ip Ident_Porta: Ident_Canal (Papel) Tipo_Fila;
    :                                     (cada porta é associada a uma)
    :                                     (fila dedicada ou compartilhada)
    :
    Ident_Porta: Ident_Canal (Papel) Tipo_Fila;
export Lista_Var_Exportadas;
end;
```

Nessa especificação são indiretamente definidas as interações de um módulo, vinculando-se suas portas aos canais. Um *ip* do tipo *Ident_Canal (Papel)* troca as primitivas definidas no canal *Ident_Canal* e desempenha a função *Papel* em relação a esse canal. Podem ser definidas também as variáveis que o módulo deseja exportar ao seu pai.

Definição do Corpo de um Módulo:

Pelo menos um corpo deve ser declarado para cada cabeçalho de módulo definido. A definição de um corpo de módulo em Estelle é mostrada abaixo.

```
body Ident_Corpo for Ident_Cabeça;
< definição do corpo >                                     (faz parte da especificação)
end;
body Ident_Corpo for Ident_Cabeça;
external;                                                  (é definido em outra
                                                           especificação)
```

A definição do corpo de um módulo pode conter três partes: declarações, inicializações e transições.

A parte relativa às declarações pode conter:

- (i) A especificação de objetos a serem manipulados, tais como constantes, tipos, variáveis, funções, procedimentos, estados (controle), conjuntos de estados, canais, pontos de interação internos;
- (ii) A especificação de sub-módulos e de variáveis do tipo módulo (declaração *modvar*).

A parte de inicializações pode conter:

- (i) A inicialização da variável de estado de controle;
- (ii) A inicialização das variáveis de estado adicionais (Pascal);
- (iii) A criação de instâncias (*instances*) dos sub-módulos (*init*);
- (iv) O estabelecimento dos *links* de comunicação (*connect* e *attach*).

A declaração *connect* realiza as conexões entre portas de instâncias de sub-módulos interligadas por canais. A declaração *attach* realiza as atribuições entre as portas de instâncias de sub-módulos filho e portas do módulo pai.

A criação de instâncias dos sub-módulos bem como o estabelecimento dos "links" de comunicação podem ser realizados estática ou dinamicamente.

Na parte relativa às transições é especificado o comportamento interno de um módulo. Cada transição é composta de duas partes: condições e ações.

As condições são definidas por cláusulas próprias a Estelle, enquanto as ações são definidas pela cláusula Estelle *to*, de declarações Pascal, de extensões Estelle e de restrições. A definição de uma transição em Estelle é mostrada abaixo.

As extensões Estelle em relação à linguagem Pascal são as declarações: *output*, *init*, *release*, *attach*, *detach*, *connect*, *disconnect*, *all*, *forone*, e *exist*. As restrições podem ser encontradas em [44].

trans

(condições)

priority < número $\in \mathcal{N}$ > (em relação às outras transições)

when < *ip.evento* > (interação de entrada)

provided < *predicado* > (expressão booleana habilitadora)

from < *estado_a* > (estado de controle vigente)

delay < *t1, t2* > (atrasos mín. e máx. permitidos)

```
(ações)
to < estado_b >                                (próximo estado de controle)
begin                                           (atualização das variáveis adicionais (Pascal))
.....
.....
output < ip.evento >                            (interação de saída)
.....
end;
```

Uma vez declaradas as transições, é encerrada a especificação do corpo do módulo.

Estelle conta também com uma semântica bem definida, cuja descrição pode ser encontrada em [44], mas o fato de não empregar tipo abstratos de dados dificulta a especificação dos protocolos das camadas superiores do Modelo OSI (este trabalho propõe uma forma de superar este problema na seção III.5). Em contrapartida, o aprendizado dessa FDT é facilitado por possuir como suporte a linguagem de programação Pascal, que é bastante conhecida.

III.3.2 ASN.1, ROS e sua Aplicação em MHS

A Série X.400 1984 introduziu os conceitos de ASN.1 (Abstract Syntax Notation) e ROS (Remote Operations), onde ASN.1 era utilizada para descrição e codificação das Unidades de Dados dos Protocolos e o ROS era utilizado como suporte ao Protocolo de Acesso ao Sistema de Transferência de Mensagens (P3).

Com as Recomendações X.400 1988 os conceitos de ASN.1 e ROS foram estendidos, e agora são utilizados para definir de forma abstrata seu modelo, serviços e protocolos.

III.3.2.1 ASN.1

A Notação ASN.1 (Abstract Syntax Notation) [35,36] pode ser entendida como a parte de declaração de dados de uma linguagem de alto nível (como C ou Pascal). Seu objetivo é elevar a descrição de estruturas de dados complexas a um nível de abstração não alcançável por meios menos poderosos ou menos formais. ASN.1 é utilizada principalmente para definir as PDUs que dois sistemas abertos trocam a fim de realizar uma tarefa distribuída.

ASN.1 é suplementado com regras padrão para representação das estruturas de dados como sequência de octetos. Isto é necessário porque os detalhes de representação de PDUs não podem variar de implementação para implementação já que as PDUs são a base para a comunicação entre sistemas abertos.

Entre os conceitos fundamentais de ASN.1, estão os conceitos de "tipo" e "valor". Um valor é uma estrutura de dados particular ou um item de informação (e.g., o inteiro 3). Um tipo é um conjunto ou classe de um ou mais valores (e.g., todos os inteiros primos). A notação contém vários tipos de dados pré-definidos tais como inteiros (*Integer*), reais (*Real*), booleanos (*Boolean*), cadeias de bits (*Bit String*), vários tipos de cadeias de caracteres (*General*, *Graphic*, *IA5*, *Numeric*, *Printable*, etc.), data-hora (*Time*), identificador de objeto (*Object Identifier*), descritor de objeto (*Object Descriptor*), nulo (*Null*) e qualquer (*Any*).

ASN.1 conta também com tipos compostos, com os quais é possível definir outros tipos, são eles:

- O tipo *Enumerated*, que habilita o usuário a definir um tipo que contém um ou mais valores não definidos previamente, como o tipo "enum" da linguagem de programação C.
- O tipo *Choice*, que habilita o usuário a definir um tipo que corresponde à união de um ou mais tipos alternativos, como o tipo "union" da linguagem C.
- O tipo *Sequence*, com o qual o usuário pode definir um tipo que é uma coleção ordenada de tipos que podem ser distintos, como o tipo "struct" da linguagem C.
- O tipo *Sequence Of*, com o qual pode-se definir um tipo que é uma coleção ordenada de vários elementos do mesmo tipo, como o tipo "array" da linguagem C.
- Os tipos *Set* e *Set Of* são análogos aos tipos *Sequence* e *Sequence Of*, respectivamente, sendo que definem coleções não ordenadas de valores.

ASN.1 define ainda Regras de Básicas de Codificação (BER), como um meio para representação de qualquer valor de qualquer tipo como uma sequência de octetos. Estas BERs servem, principalmente, para definir uma sintaxe de transferência de valores, permitindo a comunicação entre sistemas abertos.

É associado a cada tipo ASN.1 um descritor chamado *tag*, que identifica unicamente aquele tipo. As BERs para cada tipo requerem a incorporação de sua *tag* para a codificação de seus valores.

```
ExtensionField ::= SEQUENCE {
    type           [0] ExtensionType,
    criticality    [1] Criticality DEFAULT { },
    value         [2] ANY DEFINED BY type DEFAULT NULL }

ExtensionType ::= INTEGER (0 .. ub-extension-types)

Criticality ::= BIT STRING {
    for-submission (0),
    for-transfer   (1),
    for-delivery   (2) } (SIZE (0 .. ub-bit-options))

ub-extension-types    INTEGER ::= 256
ub-bit-options        INTEGER ::= 16
```

Figura III.2: Exemplo ASN.1

Um exemplo do uso da notação ASN.1 para definir tanto tipo como valor encontra-se na Figura III.2. Neste exemplo, três tipos são associados aos nomes "ExtensionField", "ExtensionType" e "Criticality", respectivamente. Dois valores são associados aos nomes "ub-extension-types" e "ub-bit-options", respectivamente. Estas declarações definem a estrutura de um campo do envelope do Protocolo P1 das Recomendações X.400.

O tipo ExtensionField é do tipo Sequence, e possui três tipos membros, que estão associados aos nomes "type", "criticality" e "value", respectivamente. O primeiro identifica o serviço que o campo requer, e é obrigatório. O segundo indica se o serviço é crítico para submissão, transferência, entrega de mensagens, ou combinação destes; é opcional e seu valor default significa que o serviço não apresenta nenhum tipo de dificuldade. O terceiro parametriza o serviço conforme necessário; é opcional e tem como valor default, o valor do tipo Null. Os três elementos têm associados a eles tags específicas no contexto no qual residem. Estas tags são 0, 1 e 2 respectivamente. Seus tipos são ExtensionType, Criticality e Any, respectivamente. O tipo Any é identificado pelo valor do elemento type.

O tipo ExtensionType é um subtipo do tipo Integer no intervalo fechado entre zero e o valor de "ub-extension-types". O tipo Criticality é um subtipo do tipo Bit String compreendendo valores cujo comprimento em bits estão no intervalo fechado entre zero e o valor de "ub-bit-options", sendo que somente os primeiros três bits têm significância na versão corrente do Protocolo P1.

Os valores denominados "ub-extension-types" e "ub-bit-options" são do tipo Integer, e determinam precisamente os limites superiores impostos sobre os inteiros que identificam, respectivamente, o serviço solicitado e o comprimento da cadeia de bits que indica sua criticalidade.

Macros ASN.1 ASN.1 permite sua própria extensão. Isto é realizado habilitando o usuário a definir alternativamente, notação não-padrão para referenciar um tipo particular, seus valores ou ambos. Uma vez feito isto, o usuário pode utilizar esta notação não-padrão sempre que a notação padrão para aquele tipo ou um de seus valores for permitida.

A extensão é realizada definindo e utilizando *Macros*. Para definir uma macro, associa-se a ela um nome, um tipo, e especifica-se detalhadamente a notação não-padrão desejada para aquele tipo e seus valores utilizando Backus-Naur Form (BNF). A definição de uma macro consiste de duas produções principais BNF e zero ou mais produções de suporte. A primeira produção principal tem a frase "TYPE NOTATION" e define a notação não-padrão para o tipo. A segunda produção principal tem a frase "VALUE NOTATION" e define a notação não-padrão para o valor. As produções de suporte definem os termos utilizados nas produções principais ou de suporte.

Para utilizar a macro, aplica-se a notação assim definida em qualquer lugar onde a notação padrão para este tipo ou seus valores seja solicitada. Detalhes sobre a construção de macros ASN.1 podem ser encontrados em [49].

III.3.2.2 ROS

As Operações Remotas podem ser entendidas como a porção de declaração de procedimentos de uma linguagem de alto nível. Seu objetivo é elevar a descrição de interações complexas entre sistemas abertos a um nível de abstração não alcançável por meios menos poderosos ou menos formais. ROS pode ser utilizada para definição abstrata de protocolos de aplicação, e é particularmente apropriado para aplicação à protocolos interativos, mas pode ser também suportar outros tipos de protocolos.

ROS baseia-se no paradigma de comunicação "pedido/resposta", e seus conceitos fundamentais são os de *Operação Remota* e *Erro Remoto*. Uma Operação Remota é uma tarefa que uma entidade em um sistema aberto (entidade chamadora) solicita que seja executada por outra entidade em um sistema aberto remoto (entidade respondente). Um Erro Remoto é uma condição excepcional que pode terminar tal tarefa prematuramente.

Um dado em ASN.1 pode acompanhar a Operação Remota. Este dado é chamado "argumento" (argument) da Operação Remota. Se a operação é bem sucedida, a entidade respondente envia uma resposta, que pode ser acompanhada de um dado em ASN.1. Este dado é chamado "resultado" (result) da Operação Re-

mota. Da mesma forma, um dado ASN.1 pode ser retornado como "parâmetro" (parameter) de um Erro Remoto. Os tipos ASN.1 utilizados para argumento e resultado dependem de como a Operação Remota é definida.

Para que duas entidades possam executar Operações Remotas é necessário que esteja estabelecida uma "associação" entre elas. O estabelecimento e terminação de uma associação são também Operações Remotas de natureza especial. Uma Operação Remota cujo objetivo é estabelecer uma associação entre duas entidades é chamada *Operação Bind* (Bind Operation), o Erro Remoto associado a esta operação é chamado *Bind Error*. Uma Operação Remota cujo objetivo é terminar uma associação entre duas entidades é chamada *Operação Unbind* (Unbind Operation), o Erro Remoto associado a esta operação é chamado *Unbind Error*. Somente a entidade iniciadora de uma Operação Bind pode iniciar uma Operação Unbind.

ROS define ainda o conceito de *Operações Encadeadas* (Linked Operations). Operações Encadeadas são aquelas cuja execução na entidade respondente pode implicar a execução de uma ou mais outras operações pela entidade chamadora. Uma operação em cuja execução o iniciador pode ser envolvido desta forma é chamada *Operação Pai*, e as operações invocadas durante a execução desta Operação Pai são chamadas *Operações Filho*.

As especificações de ROS fornecem notação formal para referenciar tipos de Operações Remotas e Erros Remotos, e, operações e erros específicos. Isto é feito através de seis macros ASN.1 que as especificações ROS definem: BIND, UNBIND, OPERATION, ERROR, APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT. A descrição destas macros pode ser encontrada em [49]. Exemplos do uso destas macros serão vistos na próxima seção.

III.3.2.3 Aplicação de ASN.1 e ROS na definição do MHS

Através das macros ASN.1 e das macros especificadas pelo ROS as Recomendações X.400 1988 definem um Modelo Abstrato onde os processos de aplicação podem ser vistos como objetos que possuem portas, através das quais ocorrem troca de informação e realização de serviços. Um objeto, por sua vez, pode ser decomposto em vários outros objetos através de técnica de refinamento (utilizando macro ASN.1). Isto possibilita um nível maior de abstração e modularidade na definição do sistema.

Os objetos, portas, serviços e refinamentos são definidos através de macros em ASN.1. A definição abstrata dos protocolos é realizada através da macros

especificadas pelo ROS.

Entretanto, esta definição abstrata não cobre todos os aspectos relevantes que necessitam ser especificados em um protocolo de aplicação, tais como: a semântica das operações remotas; a especificação de que cada operação confirmada é síncrona ou assíncrona; em que prioridade os dados de pedido, resposta e rejeição de serviços são transferidos entre as entidades sistemas abertos remotos; se é garantido "exactly-once", "at-most-once", ou nenhuma semântica; o mapeamento das macros ROS em primitivas de serviço.

Nas Recomendações X.400 todos estes aspectos são descritos em linguagem natural. A modelagem Estelle visa suprir não somente estas deficiências, como também modelar o comportamento das entidades do MHS e permitir a aplicação de ferramentas para validação, implementação e teste dos protocolos. Porém, a validação da modelagem do MHS não faz parte dos objetivos deste trabalho.

Modelo Abstrato Segundo as Recomendações X.400, uma descrição macroscópica de um processo de informação de tarefas distribuídas e do ambiente onde estas atuam é chamada Modelo Abstrato [13]. Este modelo baseia-se nos conceitos de objetos, portas, serviços e refinamentos abstratos.

(i) **Objetos Abstratos** São entidades funcionais que interagem umas com as outras. Os vários objetos podem ser de tipos distintos, e seu tipo determina sua função e comportamento. Os objetos interagem entre si através de suas "portas abstratas".

Um Objeto é especificado através da macro OBJECT em ASN.1. Esta macro lista os tipos de portas abstratas utilizadas por este objeto na interação com outros objetos. A definição desta macro encontra-se em [11].

(ii) **Portas Abstratas** São os pontos através dos quais os objetos abstratos interagem entre si. Os diferentes tipos de Portas definem os tipos de interações nelas permitidas. Uma Porta é especificada através da macro PORT em ASN.1. Esta macro define as operações disponíveis na porta. A definição desta macro encontra-se em [11].

(iii) **Serviços Abstratos** Um Serviço Abstrato é o conjunto de facilidades que um objeto oferece a outro através de sua(s) porta(s). A especificação dos serviços abstratos é uma descrição microscópica de uma tarefa de processamento de informação distribuída. Esta especificação define como a tarefa é iniciada, controlada e terminada, e se baseia nos conceitos de ROS para definir

Operações Abstratas. As Operações Abstratas são definidas utilizando-se as macros de ROS e podem ser encontradas em [11].

(iv) Refinamentos Abstratos Um Refinamento Abstrato é uma decomposição funcional de um objeto em objetos menores. Isto pode facilitar e modularizar o entendimento e a definição de objetos complexos. Esta técnica pode ser aplicada recursivamente, podendo um objeto componente conter vários objetos internos a ele. O refinamento é especificado pela macro REFINE. Esta identifica o objeto cuja estrutura interna está sendo refinada e os objetos componentes usados em sua construção. A macro REFINE pode ser encontrada em [11].

III.3.2.4 Definição Abstrata de Protocolos

No ambiente OSI, os objetos são concretizados através de processos de aplicação (AP) e a comunicação entre APs em diferentes sistemas abertos é governada pelos protocolos de aplicação OSI nos seus contextos de aplicação (AC). Um contexto de aplicação específica como a associação deve ser estabelecida e liberada, e que elementos de serviço de aplicação (ASEs) estão envolvidos na comunicação.

Os serviços abstratos de um dado protocolo são agrupados de acordo com suas características comuns em um ou mais ASEs. A camada de aplicação fornece serviços que são comuns a vários protocolos de aplicação. Estes estão representados pelo CASE (Common Application Service Element) que contém três ASEs: o ROSE (Remote Operation Service Element), o RTSE (Reliable Operation Service Element) e o ACSE (Association Control Service Element). Estes elementos são chamados ASEs básicos, e é sobre eles que as operações abstratas são mapeadas e em seguida passadas para a camada de apresentação.

O modelo apresentado pelas Recomendações X.400 é abstrato. Isto é, não é sempre possível a um observador externo identificar as fronteiras entre objetos, ou decidir o momento ou os meios pelos quais as operações ocorrerão. Entretanto, em alguns casos o modelo pode ser "concretizado". Por exemplo, um par de objetos que se comunicam através de portas correspondentes podem estar localizado em diferentes sistemas abertos. Neste caso, a fronteira entre os objetos é visível, as portas são expostas e as operações podem ser suportadas por instâncias de comunicação OSI.

A modelagem desenvolvida neste trabalho descreve o comportamento dos objetos do MHS, define suas fronteiras e as condições sob as quais ocorrem as operações; mapeando não somente a definição abstrata do sistema contida nos padrões X.400

1988 mas também os demais aspectos relevantes dos protocolos e entidades do MHS nos elementos da Linguagem Estelle.

III.4 Modelagem do MHS

Esta seção visa, principalmente, apresentar a arquitetura da especificação desenvolvida para o MHS e a modelagem realizada para cada entidade e protocolo do MHS, destacando os aspectos importantes desta modelagem. A especificação completa pode ser encontrada em [55].

III.4.1 Arquitetura Geral do Modelo

A modelagem do comportamento dos componentes do MHS baseia-se principalmente nas características importantes da linguagem Estelle. As máquinas de estados finitas que representam a funcionalidade das entidades do sistema utilizam as seguintes construções: *any* (gerência de tabelas de pontos de interação), *when* (para recepção de eventos e respostas), *from* (estado corrente da máquina), *to* (o próximo estado da máquina), *provided* (teste de condição sobre o parâmetro de uma interação), *output* (para transmissão de uma interação através de um ponto de interação).

A arquitetura geral da especificação baseia-se no modelo funcional do MHS descrito na seção II.3. Esta organização geral é definida no momento da configuração e se mantém constante durante a execução da especificação. Para cada entidade funcional do MHS foi definido um tipo de módulo para o qual se pode gerar várias instâncias, utilizando o conceito de módulo Estelle.

O atributo "Activity" é utilizado para representar o comportamento distribuído do sistema. A especificação parte de um sistema único (atributo "Systemactivity") do qual todos os módulos filhos recebem o atributo "Activity". As principais razões que conduziram a esta escolha são:

Expressão de não-determinismo: a utilização destes atributos permitem, sob certas condições, atingir um alto grau de não-determinismo para a representação do sistema [52];

Abstração em relação às escolhas de implementação: A descrição do MHS a partir de um sistema único implica uma certa independência de sua configuração em relação à implementação [52].

A modelagem Estelle do MHS consiste basicamente de um módulo principal de atributo "Systemactivity" o qual é refinado em sub-módulos que modelam as enti-

dades funcionais do sistema. Este módulo principal chama-se MHS e sua função é definir os sub-módulos que representam as entidades funcionais do MHS (UA, MS, MTA) e realizar a configuração do sistema como um todo, refletindo a arquitetura geral da especificação (Figura III.3). Foi incluída nesta modelagem a especificação dos Elementos de Serviço Comuns da Camada de Aplicação (CASE). Isto possibilita uma visão global do funcionamento de toda a Camada de Aplicação.

Alguns destes sub-módulos são também refinados a fim de modularizar e permitir um bom entendimento da especificação.

A modelagem das Unidades de Acesso está fora do escopo deste trabalho por exigir conhecimentos específicos de cada serviço telemático ao qual a Unidade de Acesso se aplica. Porém, o protocolo do MHS através do qual estas Unidades de Acesso utilizam os serviços de manipulação de mensagens é modelado neste trabalho.

Os sub-módulos principais do módulo MHS são:

UA_TYPE : Possui atributo *activity* e modela a entidade UA do MHS. Tem como parâmetro de entrada um código referente ao protocolo de acesso utilizado. Coletivamente, as instâncias do sub-módulo UA_TYPE formam a modelagem do Protocolo P2. A definição deste módulo é mostrada abaixo.

```
module UA_TYPE activity (access_type: cod_prt);
  ip   USR:  usuario_interface (provider);
      P_AC:  acess_interface   (user);
end;
```

```
{%incluir UABODY}
```

MS_ACCESS_TYPE : Possui atributo *activity* e modela a parte do usuário do MS que faz acesso aos serviços desta entidade. Este módulo não corresponde à qualquer entidade do MHS e não é definido nas Recomendações X.400, sendo definido para modularidade da especificação de modo que possa ser conectado indistintamente a qualquer módulo que faça acesso ao MS. A definição deste módulo é mostrada abaixo.

```
module MS_ACESS_TYPE activity;
  ip    P_AC:  acess_interface  (provider);
        C11:  rose_interface   (user);
        C12:  acse_interface   (user);
end;

{$incluir MSAC}
```

MTS_ACCESS_TYPE : Possui atributo activity e modela a parte do usuário do MTS que faz acesso aos serviços deste sistema. Este módulo não corresponde à qualquer entidade do MHS e não é definido nas Recomendações X.400, sendo definido para modularidade da especificação de modo que possa ser conectado indistintamente a qualquer módulo que faça acesso ao MTS. A definição deste módulo é mostrada abaixo.

```
module MTS_ACESS_TYPE activity;
  ip    P_AC:  acess_interface  (provider);
        C11:  rose_interface   (user);
        C12:  acse_interface   (user);
end;

{$incluir MTSAC}
```

MS_TYPE : Possui atributo activity e modela a parte do provedor de serviços da entidade MS do MHS. Coletivamente, os sub-módulos MS_ACCESS_TYPE e MS_TYPE modelam o Protocolo P7. A definição deste módulo é mostrada abaixo.

```
module MS_TYPE activity;
  ip    P_AC:  acess_interface  (user);
        C11:  rose_interface   (user);
        C12:  acse_interface   (user);
end;

body MS_ACESS_BODY for MS_ACESS_TYPE;
```

```
{ $incluir MSBODY }
```

MTA_TYPE : Possui atributo *activity* e modela a entidade MTA do MHS. Este sub-módulo é, por sua vez, refinado em outros módulos: **P3_PROVIDER_TYPE**, **P1PM_TYPE** E **P1OL_TYPE**, que serão descritos na seção III.4. Em particular, coletivamente, os sub-módulos **P3_PROVIDER_TYPE** e **MTS_ACCESS_TYPE** modelam o Protocolo P3 e o sub-módulo **MTA_TYPE** modelam o Protocolo P1, e conseqüentemente, o MTS. A definição deste módulo é mostrada abaixo.

```
module MTA_TYPE activity;  
  ip  C11: array [1..max_usr] of ROSE_interface (user);  
      C12: array [1..max_usr] of ACSE_interface (user);  
      C2 : array [1..max_mta] of RTSE_interface (user);  
end;
```

```
{ $incluir MTABODY }
```

CASE_PROVIDER_TYPE : Possui atributo *activity* e modela parte de Elementos de Serviço Comuns da Camada de Aplicação (Common Application Service Elements - CASE), que são o ACSE, ROSE e RTSE. A definição deste módulo é mostrada abaixo.

```
module CASE_PROVIDER_TYPE activity;  
  ip  RT  : array [1..max_rtusr] of RTSE_interface (provider);  
      RS  : array [1..max_rsusr] of ROSE_interface (provider);  
      A   : array [1..max_ausr]  of ACSE_interface (provider);  
      PCEP: array [1..max_cnx]   of APRESENTACAO_interface (user);  
end;
```

```
body CASE_BODY for CASE_PROVIDER_TYPE;
```


PRES_SERVICE : Possui atributo `activity` e modela o serviço da Camada de Apresentação, através do qual as associações são criadas para que as entidades funcionais possam interagir. A definição deste módulo é mostrada abaixo.

```
module APRESENTACAO_TYPE activity;
    ip PCEP : array [1..max_pusr, 1..max_cnx] of
        APRESENTACAO_interface (provider);
end;

body APRESENTACAO_BODY for APRESENTACAO_TYPE;
```

A configuração do sistema consiste do seguinte:

- declaração e criação de instâncias chamadas `UA[i]`; estas instâncias têm o tipo `UA_TYPE` e têm o comportamento dado pelo corpo `UA_BODY`;
- declaração e criação de instâncias chamadas `MS[i]`; estas instâncias têm o tipo `MS_TYPE` e têm o comportamento dado pelo corpo `MS_BODY`;
- declaração e criação de instâncias chamadas `MTA[i]`; estas instâncias têm o tipo `MTA_TYPE` e têm o comportamento dado pelo corpo `MTA_BODY`;
- declaração e criação de instâncias chamadas `MS_ACS[i]`; estas instâncias têm o tipo `MS_ACCESS_TYPE` e têm o comportamento dado pelo corpo `MS_ACCESS_BODY`;
- declaração e criação de instâncias chamadas `MTS_ACS[i]`; estas instâncias têm o tipo `MTS_ACCESS_TYPE` e têm o comportamento dado pelo corpo `MTS_ACCESS_BODY`;
- declaração e criação de instâncias chamadas `CCASE[i]`; estas instâncias têm o tipo `CASE_PROVIDER_TYPE` e têm o comportamento dado pelo corpo `CASE_BODY`;
- declaração e criação de instâncias chamadas `PRES_SERVICE[i]`; estas instâncias têm o tipo `PRES_SERVICE_TYPE` e têm o comportamento dado pelo corpo `PRES_SERVICE_BODY`;

São definidas constantes que determinam o número de instâncias criadas de cada módulo. Estas constantes são:

- `max_usr`: Define o número máximo de usuários que o sistema pode suportar, conseqüentemente, esta constante define também o número de instâncias `UA[i]`.
- `max_ua1`: Define o número de instâncias `UA` que utilizarão os serviços de `MS`. Conseqüentemente, esta constante define também o número de instâncias `MS[i]` e `MS_ACS[i]`.
- `max_ua2`: Define o número de instâncias `UA` que utilizarão diretamente os serviços do `MTS`, logo, a soma das constantes `max_ua1` e `max_ua2` define o número de instâncias `MTS_ACS[i]`.
- `max_mta`: Define o número de instâncias do sub-módulo `MTA`.
- `max_cnx`: Define o número máximo de conexões um sub-módulo `MTA` pode suportar.
- `max_sa`: Define o número de sistemas abertos que constituem o sistema, conseqüentemente, define o número de instâncias do sub-módulo `CCASE`.

Após a declaração e criação das instâncias dos sub-módulos descritos acima, algumas instâncias `UA[i]` são conectadas às instâncias `MS_ACS[i]`, outras instâncias `UA[i]` são conectadas às instâncias `MTS_ACS[i]`, para que estas possam fazer acesso às instâncias `MS` e `MTA` respectivamente.

Esta conexão é realizada através da interface "access-interface". Esta interface não faz parte dos padrões X.400, foi criada para esta modelagem, todavia é bem definida e independente de implementação.

As instâncias `MS[i]` são conectadas às instâncias `MTS_ACS[j]`, para que este possa fazer acesso às instâncias `MTA`. As instâncias que representam as entidades funcionais do `MHS` assim formadas são então conectadas às instâncias `CCASE` que finalmente são conectadas ao sub-módulo `PRES_SERVICE`.

O corpo associado ao módulo `MHS` é vazio, ou seja, não possui transições. Portanto, o módulo `MHS` é constituído somente da parte de inicialização. A arquitetura da especificação é então estática e se mantém inalterada durante a execução da especificação.

Esta modelagem reflete um `MHS` completamente distribuído, ou seja, onde as entidades de acesso, armazenamento e transferência se encontram em sistemas

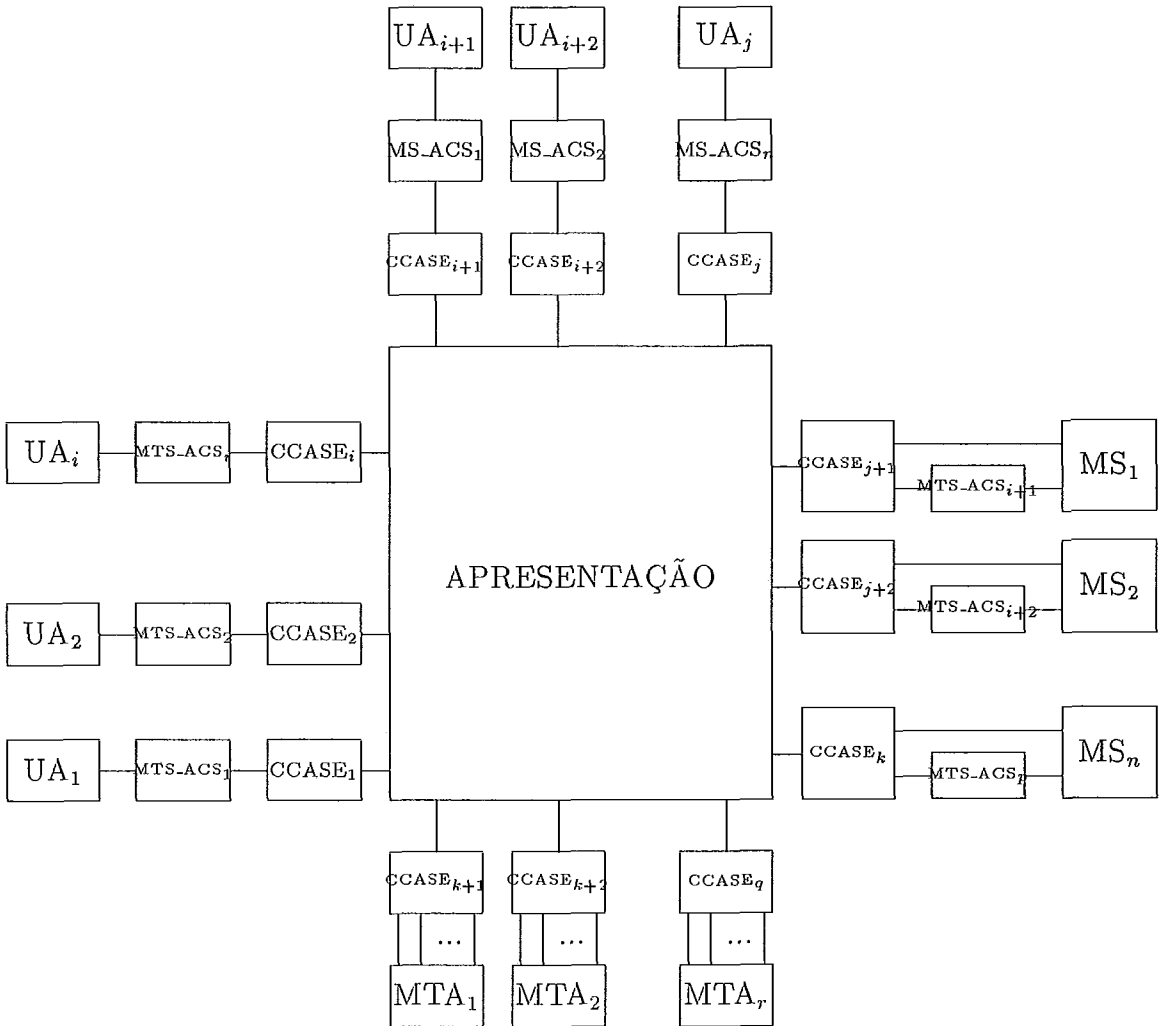


Figura III.3: Arquitetura Geral da Modelagem do MHS.

abertos distintos entre si. O fato de que existem inúmeras variantes para a configuração física dos componentes do MHS, se reflete em sua modelagem Estelle. Como a intenção foi modelar o sistema de forma mais distribuída possível, torna-se difícil descrever graficamente tal arquitetura deixando claro as relações entre os sub-módulos (Figura III.3).

A seguir a modelagem de cada sub-módulo é descrita separadamente.

III.4.2 Modelagem do Agente Usuário (UA)

III.4.2.1 O Agente Usuário

O Agente Usuário (UA) é a entidade funcional do MHS que tem por objetivo principal tornar os recursos de manipulação de mensagens disponíveis ao usuário. O UA auxilia o usuário na transmissão e recepção de mensagens e contém as funções necessárias a interação com o MTS (ou MS) na sua operação de submissão e entrega de mensagens.

Conforme visto no capítulo II, os UAs são agrupados em classes de acordo com o tipo de mensagens que manipulam. UAs de uma mesma classe são chamados "UAs cooperantes", já que devem cooperar para fornecer serviços e comunicação a seus usuários. No momento, somente uma classe de UAs cooperantes foi padronizada, a saber, a classe dos UAs que manipulam mensagens interpessoais (mensagens-IP), os IPM-UAs [18].

As regras de transmissão e mensagens-IP são definidas pelo Protocolo de Mensagens Interpessoais, também chamado Protocolo P2.

A cooperação entre IPM-UAs não envolve o estabelecimento e término de associação entre estas entidades. O UA estabelece associação somente com a entidade funcional através da qual ele utiliza os serviços de Transferência de Mensagens. Esta entidade pode ser o MS ou o MTA. Portanto, o UA deve utilizar o Protocolo de Acesso ao MS (P7) ou ao MTS (P3), respectivamente.

A modelagem Estelle do UA é descrita pelo sub-módulo UA_TYPE, e visa principalmente descrever formalmente os seguintes aspectos:

- O comportamento do UA na execução das operações.
- Sua interação com o usuário;
- Sua interação com o sub-módulo de protocolo de acesso utilizado.

São definidas quatro classes de operações que um IPM-UA deve realizar para seu usuário: operações de submissão, de recepção, de gerenciamento, e as operações automáticas.

Através das operações de submissão, um UA permite que seu usuário submeta mensagens, sondagens, e notificações de recebimento de mensagens para transferência pelo MTS. As regras para realização destas operações podem ser encontradas em [18].

Através das operações de recepção, um UA avisa a seu usuário da recepção de mensagens e notificações.

Através das operações de gerenciamento, o usuário pode alterar o valor de "variáveis de estado" mantidas no UA para a realização de ações que o UA pode realizar automaticamente. São definidas oito variáveis de estado:

auto-discard-expired-IPMs : Indica se as mensagens cujas datas de expiração passaram devem ser deletadas.

auto-discard-obsolete-IPMs : Indica se as mensagens cujas datas de obsolescência passaram devem ser deletadas.

auto-acknowledge-IPMs : Indica se deve ser gerada e enviada automaticamente notificação de recebimento para as mensagens recebidas.

auto-acknowledge-suppl-receipt-info : Indica valor de um campo do cabeçalho da notificação [18].

auto-forward-IPMs : Indica se as mensagens recebidas devem ser automaticamente redestinadas a outro(s) usuário(s).

auto-forward-recipients : Especifica o(s) usuário(s) para os quais as mensagens serão automaticamente redestinadas.

auto-forward-heading : Especifica o cabeçalho da mensagem que conterà a mensagem automaticamente redestinada.

auto-forward-comment : Especifica um comentário a ser inserido na notificação de não recebimento que será gerada para o originador da mensagem automaticamente redestinada.

Através das operações automáticas, o UA executa procedimentos internos de auto-deleção de mensagens, auto-confirmação de recepção de mensagens, auto-redirecionamento de mensagens recebidas a outros usuários.

O UA utiliza suas variáveis de estado para realizar estas operações.

III.4.2.2 Modelagem Estelle do UA

Os procedimentos do UA foram modelados em Estelle de forma modular. Consiste de um sub-módulo UA_TYPE com atributo *activity* e comportamento dado pelo corpo UA_BODY.

Para seu funcionamento, é necessário que este sub-módulo seja conectado a outro sub-módulo que modele seu protocolo de acesso. Este sub-módulo pode

ser do tipo `MTS_ACCESS_TYPE` ou do tipo `MS_TYPE`, conforme o protocolo de acesso utilizado seja o Protocolo P3 ou P7, respectivamente. Estes sub-módulos serão descritos nas seções seguintes.

O sub-módulo `UA_TYPE` tem como parâmetro de entrada, um código referente ao tipo de protocolo de acesso utilizado (P3 ou P7).

Por causa da distribuição modular desta modelagem, foram definidos canais para a comunicação entre os módulos que modelam o protocolo de acesso e o UA e entre este e seu usuário. O submódulo `UA_TYPE` possui dois pontos de interação para este fim, chamados, respectivamente, `P_AC` e `USR`.

```
channel usuario_interface(user,provider);
  by user :
    SendMsg          (arg:asn1_def); SendProbe      (arg:asn1_def);
    SendAdmOp        (arg:asn1_def); SendMsOp       (arg:asn1_def);
    ChangeAutoDiscard (arg:asn1_def); ChangeAutoAck (arg:asn1_def);
    ChangeAutoForward (arg:asn1_def);
    UnBind;
  by provider :
    RecIPM   (arg:asn1_def); RecIPN (arg:asn1_def);
    RecReport (arg:asn1_def);
    Res (arg:asn1_def);      Err (arg:asn1_def);
    Rju (arg:asn1_def);
    UnbindRes; UnbindAnormal;

channel access_interface(user,provider);
  by user, provider :
    Bind; BindRes; BindErr; Unbind; UnbindRes;
    Inv (OP:asn1_def); Res (RES:asn1_def); Err (ERR:asn1_def);
    Rju (RJU:asn1_def); Rjp;

module UA_TYPE activity (access_type:cod_prt);
  ip      USR:  usuario_interface (provider);
          P_AC: access_interface  (user);
end;

{$incluir UABODY}
```

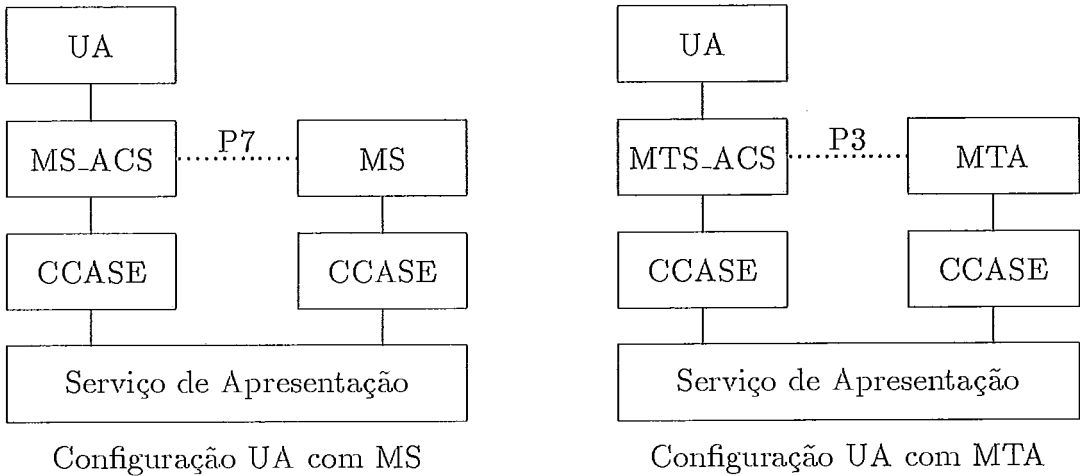


Figura III.4: Arquitetura da Modelagem Estelle do UA

O sub-módulo UA_TYPE é declarado no módulo principal da modelagem, MHS, onde é também criado e instanciado com o nome UA[i].

Devido a forma modular seguida em relação aos protocolos de acesso, o comportamento do sub-módulo UA foi modelado de forma simples, sem refinamentos. Este sub-módulo possui um canal interno chamado Intern. Este canal é utilizado pela próprio módulo na execução de ações automáticas tomadas na recepção de mensagens.

```
channel P2_INTERNAL (A,B);
  by A,B: Msg (arg: asn1_def);

ip
  Intern_in: P2_INTERNAL(A);
  Intern_out: P2_INTERNAL(B);
```

A arquitetura desta modelagem, pode assumir dois tipos em relação ao protocolo de acesso utilizado. Isto é mostrado na Figura III.4.

III.4.3 Modelagem do "Message Store" (MS)

III.4.3.1 O Message Store

O Message Store (MS) é modelado como um objeto atômico que age como um provedor de serviços para seu usuário (i.e., o UA) e um usuário dos serviços fornecidos pelo MTS. Conforme citado anteriormente, o MS age como um intermediário

entre o UA e o MTS. Sua função principal é aceitar a entrega de mensagens do MTS e armazená-las até que o UA as requisite, provendo serviços de recuperação de mensagens (retrieval). O MS também provê os serviços de submissão indireta e administração ao UA, que passa transparentemente ao MTS.

Como o UA, o MS age como provedor de serviços para um único usuário. A Figura III.5 mostra o modelo de interação entre UA-MS-MTS.

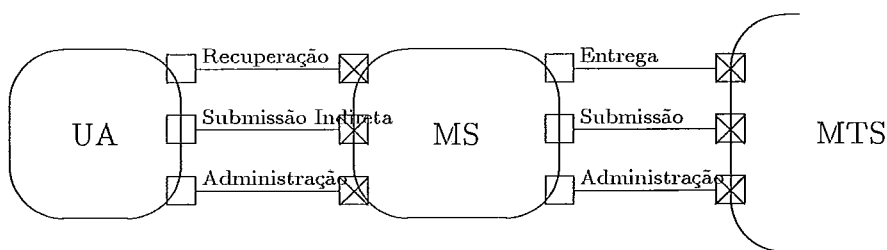


Figura III.5: Interação UA-MS-MTS

A recomendação X.407 define um modelo de armazenamento de informação para o MS. O MS armazena e mantém *bases de informação* que são constituídas de *entradas*, que, por sua vez, são constituídas de *atributos*. Esta recomendação define e descreve a *base de informação de mensagens armazenadas* (stored messages), que contém informação derivada da entrega de mensagens e notificações pelo MTS.

As normas MOTIS da ISO [20] definem mais dois tipos de bases de informação para armazenamento de informação sobre a entrada e saída de mensagens, chamadas *inlog* e *outlog* respectivamente. A definição destas bases de informação não faz parte do escopo das Recomendações X.400 do CCITT.

Cada base de informação é organizada como uma seqüência de entradas. Uma entrada na base de informação representa um único objeto, tal como uma mensagem

Cada entrada é identificada pelo seu *número de seqüência*, único em uma base de informação, e gerado pelo MS na criação de uma nova entrada. Em uma base de informação, o MS gera números de seqüência em ordem ascendente e sem reutilização.

Uma entrada consiste de um conjunto de atributos. Cada atributo provê uma parte de informação sobre, ou derivada do dado ao qual a entrada corresponde. Exemplos de tipos de atributos de uma entrada são: seu número de seqüência, sua data de criação e o envelope de uma mensagem.

Um atributo consiste de um *típo*, que identifica a classe da informação dada por ele, e um *valor* correspondente, que é instância particular daquela classe na entrada.

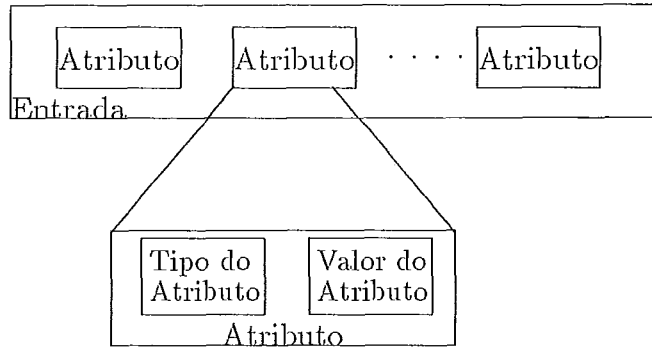


Figura III.6: Estrutura de uma Entrada do MS

Além de servir como um depósito de mensagens, o MS oferece operações sobre estas mensagens tais como: obter informações sobre as mensagens, recuperá-las ou deletá-las. Ao solicitar estas operações, o UA deve fornecer descritores das mensagens as quais a operação se aplica. Estes descritores consistem de estruturas de dados que permitem a seleção de mensagens e podem ser combinados para formar expressões complexas.

O MS provê serviços a seu usuário através do Protocolo de Acesso ao MS (Protocolo P7). Este protocolo constitui-se da comunicação entre o MS e seu usuário através de uma conexão de apresentação.

Para prover ao seu usuário os serviços de submissão indireta, o MS por sua vez, utiliza o Protocolo de Acesso ao MTS (Protocolo P3), cuja modelagem será descrita na próxima seção. Na modelagem Estelle, a Figura III.5 toma a forma da Figura III.7.

A Modelagem Estelle do MS visa descrever formalmente o protocolo de acesso à esta entidade e a execução de suas operações.

O UA pode fazer uso das várias funções do MS através das operações remotas do Protocolo P7 definidas em seu modelo abstrato. São elas as operações Summarize, List, Fetch, Delete, Register-MS.

A operação Sumarize permite que o UA obtenha um contador das entradas que contêm um dado conjunto de atributos. O número de mensagens identificadas pode ser restringido por um limite ou por seletores baseados em atributos. As informações retornadas por esta operações podem servir para formular consultas posteriores ao MS.

A operação List permite que o UA solicite um conjunto de atributos de mensagens retidas no MS. As mensagens a serem consideradas para listagem devem satisfazer a critérios de seleção como na operação Sumarize. Nem todos os atributos de mensagem podem ser obtidos com esta operação. Geralmente, os atributos

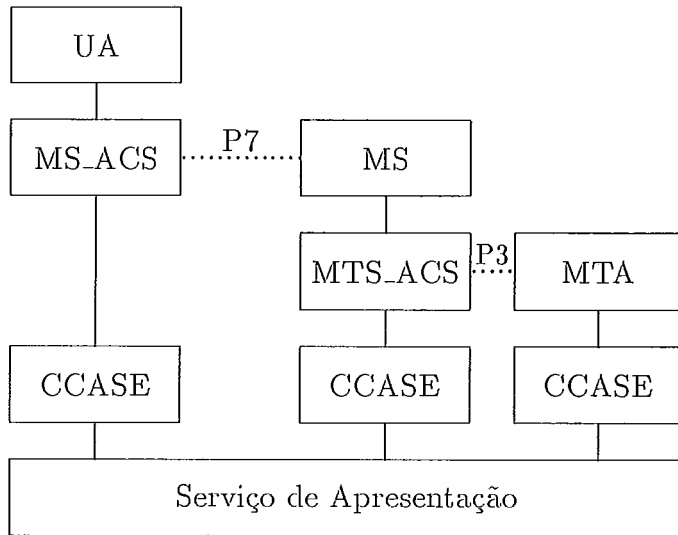


Figura III.7: Arquitetura da Modelagem do Protocolo P7

não permitidos nesta operação são aqueles que retem muita informação tais como, conteúdo e envelope de mensagens. Os resultados de uma operação List são tipicamente usados para selecionar mensagens para leitura ou ação posterior.

A operação Fetch age como a operação List mas no caso desta operação, somente uma mensagem é selecionada por vez e todos os atributos podem ser requisitados. Se mais de uma mensagem satisfizer ao critério de seleção, o MS retorna seus números de seqüencia para que o usuário possa formular operações posteriores.

A operação Delete permite que o UA delete mensagens (entradas) no MS a partir de seus números de seqüencia ou seletores baseados em atributos. O MS retornará um erro caso o UA tente deletar uma entrada cujos atributos ainda não foram consultados por uma operação List ou Fetch. Além disso, se o UA deleta uma entrada que contém uma mensagem-IP cujas partes de corpo ainda não foram todas lidas, o MS automaticamente retorna uma notificação de não-recebido ao originador da mensagem.

A operação Alert permite que o MS avise ao UA da entrega de mensagens ou notificações.

A operação Register-MS permite que o UA registre informação no MS que poderá ser utilizada "por default" na execução de operações e/ou solicite que o MS execute operações automáticas na entrega de mensagens tal como, redestinar mensagens a outro(s) destinatário(s).

Como citado acima, a classe de operações de submissão indireta permitem que

o UA envie mensagens e sondagens via o MS, que repassa transparentemente ao MTS. Opcionalmente, a pedido do usuário, o MS pode, durante a submissão, inserir no corpo da mensagem uma outra mensagem residente no MS.

O MS deve ainda realizar certas funções específicas ao tipo de conteúdo manipulado. Estas funções o distinguem de um MS genérico. Se o MS está associado a um IPM-UA, ele deve realizar algumas funções específicas para mensagens interpessoais. Estas funções se referem à criação de entradas, manutenção de atributos, execução de ações automáticas no MS. Estas funções são descritas no seção 9 da Recomendação X.420 [18].

III.4.3.2 O Protocolo P7

As recomendações X.413 e X.419 definem abstratamente o Protocolo P7 utilizando as macros do ROS. As macros MSBind e MSUnbind definem os parâmetros para estabelecimento e término da associação entre a entidade MS e seu usuário (UA). Os parâmetros das operações são também definidos utilizando macros ROS e tipos ASN.1 [16].

Cada classe de serviço foi agrupado em um ASE (Application Service Element): O MRSE (Message Retrieval Service Element), que corresponde aos serviços de recuperação, o MSSE (Message Submission Service Element), que corresponde aos serviços de submissão, o MASE (Message Administration Service Element), que corresponde aos serviços de administração. Estes ASEs unidos aos ASEs comuns aos protocolos de aplicação (ACSE, ROSE, RTSE) formam o conjunto de ASEs que definem os Contextos de Aplicação (ACs) do Protocolo P7. Existem dois contextos definidos para este protocolo:

- *ms-access*: composto dos ASEs: MRSE, MSSE, MASE, ACSE e ROSE. Este contexto de aplicação é mandatório.

```
ms-access  APPLICATION-CONTEXT
           APPLICATION SERVICE ELEMENT {aCSE}
           BIND          MSBind
           UNBIND        MSUnbind
           REMOTE OPERATIONS {rOSE}
           INITIATOR CONSUMER OF {mSSE, mRSE, mASE}
           ABSTRACT SYNTAXES {
               id-as-acse          -- of ACSE
               id-as-msse          -- of MSSE, including ROSE
```

```
id-as-mdse      -- of MRSE, including ROSE
id-as-mase      -- of MASE, including ROSE
id-as-ms        -- of MTSBind, MTSUnbind  --}
 ::= id-ac-ms-access
```

- *ms_reliable_access*: composto dos ASEs: MRSE, MSSE, MASE, ACSE, ROSE e RTSE. Este contexto de aplicação é opcional.

```
ms-reliable-access  APPLICATION-CONTEXT
APPLICATION SERVICE ELEMENT {acSE, rTSE}
BIND      MSBind
UNBIND    MSUnbind
REMOTE OPERATIONS {rOSE}
INITIATOR CONSUMER OF {mSSE, mRSE, mASE}
ABSTRACT SYNTAXES {
    id-as-acs      -- of ACSE
    id-as-mss      -- of MSSE, including ROSE
    id-as-mrs      -- of MRSE, including ROSE
    id-as-mas      -- of MASE, including ROSE
    id-as-ms       -- of MSBind, MSUnbind,
                    -- including RTSE }
 ::= id-ac-mts-reliable-access
```

O Protocolo P7 consiste de três etapas:

- Estabelecimento da Associação: Conforme definido pela macro MSBind. Os dados da macro contém a identificação da entidade remota e estabelece o contexto de segurança na qual as operações subseqüentes irão ocorrer. Ao fim da operação MSBind os sistemas fim haverão validado suas identificações mutuamente e conhecerão a(s) chave(s) a serem utilizadas em funções de segurança.

No Protocolo P7 somente o UA tem direito de iniciar o estabelecimento de associação.

- Execução de Operações: Havendo a associação, as entidades podem então solicitar a execução das operações. As operações do Protocolo P7 são executadas assincronamente sujeitas as seguintes condições: As operações Delete e Register-MS não devem ser executadas até que todas as operações pendentes

tenham sido completadas. Além disso, as operações são executadas na ordem em que foram solicitadas.

- **Término da Associação:** Se o UA não necessita mais dos serviços do MS e não existem "Results" pendentes, a associação é terminada conforme a macro MSUnbind.

Realização da Associação

A realização do serviço definido pela macro MSBind, ou seja, o estabelecimento de associação se dá através do serviço A-ASSOCIATE do ACSE, se o contexto de aplicação é "ms.access" ou através do serviço RT-OPEN do RTSE se o contexto de aplicação é o "ms-reliable-access".

Os parâmetros do serviço A-ASSOCIATE são utilizados da seguinte forma:

- **Mode:**

O parâmetro "mode" da primitiva A-ASSOCIATE.Request deve ter o valor "normal mode".

- **Application Context Name:**

O parâmetro "application-context-name" da primitiva A-ASSOCIATE.Request deve propor o contexto de aplicação ms.access através de seu identificador "id-ac-ms-access".

- **User Information:**

O argumento da macro MSBind é mapeado no parâmetro "user-information" da primitiva A-ASSOCIATE.Request.

- **Presentation Context Definition List:**

Este parâmetro contém a definição do contexto de apresentação para cada sintaxe abstrata citada no contexto de aplicação. Uma definição de contexto de apresentação é constituído de um identificador de contexto de apresentação e um nome de sintaxe abstrata para o ASE.

- **Quality of Service:**

Este parâmetro deve ser utilizado pelo iniciador da associação na primitiva A-ASSOCIATE.Request e pelo respondedor da associação na primitiva A-ASSOCIATE.Response. Os componentes "extended control" e "optimized dialogue transfer" devem ser usados com o valor "not required". Os demais componentes devem ser usados com os valores "default" definidos em [29,23].

Se o RTSE faz parte do contexto de aplicação, o RTSE é o único usuário dos serviços P-ACTIVITY-START, P-DATA, P-MINOR-SYNCRONIZE, P-ACTIVITY-END, P-ACTIVITY-INTERRUPT, P-ACTIVITY-DISCARD, P-U-EXCEPTION-REPORT, P-ACTIVITY-RESUME, P-P-EXCEPTION-REPORT, P-TOKEN-PLEASE e P-CONTROL-GIVE.

Uso dos Serviços de Sessão e Transporte

Se o RTSE não faz parte do contexto de aplicação, as unidades funcionais do Serviço de Sessão necessárias ao Protocolo P7 são: Kernel e Duplex.

Se o RTSE faz parte do contexto de aplicação, as unidades funcionais do Serviço de Sessão necessárias ao Protocolo P7 são: Kernel, Half-duplex, Exceptions. Minor-synchrone e Activity-management.

Quanto ao Serviço de Transporte, suporte à Classe 0 é obrigatório. Suporte à outras classes é opcional

III.4.3.3 Modelagem Estelle do MS

A modelagem Estelle do Protocolo P7 é constituída de uma instância do sub-módulo MS_ACS, do tipo MS_ACCESS_TYPE, uma instância do sub-módulo MS, do tipo MS_PROVIDER_TYPE, duas instâncias do sub-módulo CCASE, do tipo CASE_PROVIDER_TYPE, e a instância do sub-módulo PRES_SEVICE, com mostra a Figura III.7.

O sub-módulo MS_ACS funciona como a parte do usuário do MS que faz acesso as seus serviços e portanto é conectado ao sub-módulo UA através de uma interface simples, chamada "access_interface", não-padrão, porém independente de implementação (vide §III.UA). Esta divisão foi feita para efeito de modularidade da especificação, conforme visto na seção III.3.1. O módulo MS_ACCESS_TYPE é definido conform mostrado abaixo.

```
module MS_ACCESS_TYPE activity;
  ip   P_AC:  access_interface (provider);
      C11:  rose_interface   (user);
      C12:  acse_interface  (user);
end;
```

```
{${incluir MSAC}
```

Este sub-módulo tem três pontos de interação externos chamados P_AC, RS e AC. Através destes pontos de interação, este sub-módulo oferece os serviços de acesso ao MS, e utiliza os serviços do ROSE e do ACSE, respectivamente.

O corpo correspondente à instância do sub-módulo MS_ACS possui sete estados de controle (declaração state) cuja mudança é determinada pelos eventos detectados em seus pontos de interação e determina a transição a ser executada. A especificação de estados de controle não consta dos padrões X.400, foram definidos para esta modelagem.

Este sub-módulo possui um módulo filho que especifica um "timer", e que tem o tipo TIMER_TYPE e possui atributo activity. O comportamento do timer é dado pelo corpo TIMER_BODY. Este corpo qualificado como "external", ou seja, não foi especificado nesta modelagem.

```
module TIMER_TYPE activity;  
  ip T : TIMER_interface (provider);  
end;  
  
body TIMER_BODY for TIMER_TYPE; external;
```

São criadas duas instâncias do módulo TIMER_TYPE chamadas TIMER_OP e TIMER_REP. A instância TIMER_OP representa o timer relativo à execução das operações. Ele é ativado no envio de cada pedido de operação a entidade remota. A instância TIMER_REP representa o timer relativo ao estabelecimento de associação. Ele é ativado no envio do pedido de estabelecimento de associação; caso ocorra "time-out", o pedido é reinicializado até um certo limite local.

Associados a estas instâncias, estão pontos de interação através dos quais os timers são ativados e os time-outs são detectados.

```
ip TOP      : TIMER_interface (user);  
   TREP     : TIMER_interface (user);
```

O uso de timers para controle de aspectos do comportamento da entidade MS não consta dos padrões X.400. Portanto, estes foram definidos para esta modelagem.

Este sub-módulo possui um ponto de interação interno através do qual são ativadas transições que asseguram o bom funcionamento do protocolo quanto ao

tratamento de erros e asseguram que a associação não será terminada enquanto todas as operações não forem respondidas. Este ponto de interação é definido pelo canal de comunicação P7_INTERNAL mostrado abaixo.

```
channel P7_INTERNAL(A,B);
  by A,B : Error; PreRel;

ip
  Intern_in: P7_INTERNAL(A);
  Intern_out: P7_INTERNAL(A);
```

O sub-módulo MS modela o comportamento do próprio MS na realização de seus serviços e interação com seu usuário. Este sub-módulo tem três pontos de interação externos chamados P_AC, RS e AC. Através destes pontos de interação, este sub-módulo utiliza os serviços de acesso ao MTS, e utiliza os serviços do ROSE e do ACSE para comunicar-se com seu usuário remoto, respectivamente. O módulo MS_TYPE é definido conforme mostrado abaixo.

```
module MS_TYPE activity;
  ip P_AC: access_interface (user);
    C11: rose_interface (user);
    C12: acse_interface (user);
end;
```

```
{$incluir MSBODY}
```

O corpo correspondente à instância do sub-módulo MS_ACS possui dois estados de controle (declaração state) cuja mudança é determinada pelos eventos detectados em seus pontos de interação e determina a transição a ser executada. A especificação de estados de controle não consta dos padrões X.400, foram definidos para esta modelagem.

Como no sub-módulo MS_ACS, é definido um módulo TIMER_TYPE com corpo TIMER_BODY, e neste caso, são criadas duas instâncias chamadas TIMER_OP e TIMER_MTA. A instância TIMER_OP representa o timer relativo à execução das

- Session Requirements:

Este parâmetro deve ser utilizado pelo iniciador da associação na primitiva A-ASSOCIATE.Request e pelo respondedor da associação na primitiva A-ASSOCIATE.Response; e deve especificar as seguintes unidades funcionais:

- a) Kernel;
- b) Duplex.

A realização do serviço definido pela macro MSUnbind, ou seja, término da associação se dá através do serviço A-RELEASE do ACSE, se o contexto de aplicação é "ms_access", ou através do serviço RT-CLOSE do RTSE se o contexto de aplicação é "ms_reliable_access".

Os parâmetros do serviço A-RELEASE são utilizados da seguinte forma:

- Result:

Este parâmetro deve ter o valor "affirmative".

O ROSE é o usuário dos demais serviços A-ABORT, A-P-ABORT do ACSE.

Realização de Operações

A realização dos serviços dos ASEs MRSE, MSSE e MASE, ou seja, as operações de Recuperação, Submissão e Administração se dá através dos serviços RO-INVOKE, RO-RESULT, RO-ERROR do ROSE de acordo com [30,25].

Ao se solicitar uma operação, o argumento desta será mapeada no parâmetro "user-data" do serviço RO-INVOKE do ROSE. A entidade provedora da operação enviará o resultado (ou erro) da operação em questão no parâmetro "user-data" do serviço RO-RESULT (RO-ERROR) do ROSE, que dependendo do contexto de aplicação utilizará o serviço P-DATA (ms_access) ou o serviço RT-TRANSFER (ms_reliable_access) que utilizará o serviço P-DATA.

Uso dos serviços de Apresentação

O Protocolo P7 utiliza os serviços da Camada de Apresentação em modo normal. Neste modo, um diferente contexto de apresentação (presentation-context) é utilizado para cada sintaxe abstrata incluída no contexto de aplicação.

O ACSE é o único usuário dos serviços P-CONNECT, P-RELEASE, P-U-ABORT e P-P-ABORT.

Se o RTSE não faz parte do contexto de aplicação, o ROSE é o único usuário do serviço P-DATA.

operações. Ele é ativado no envio de cada pedido de operação à entidade remota. A instância `TIMER_MTA` é utilizado na associação entre MS e MTS quando o MS é o iniciador da associação a fim de determinar o término da associação. Este timer é ativado quando todas as operações solicitadas ao MTS houverem sido respondidas.

Os sub-módulos `CCASE` modelam os elementos de serviço comuns da Camada de Aplicação (`CASE`), e o sub-módulo `PRES_SERVICE` provê o serviço da Camada de Apresentação.

O sub-módulo `CCASE` faz uso do serviço de apresentação através do ponto de interação `PCEP`.

As instâncias dos sub-módulos `MS_ACS`, `MS`, `CCASE` e `PRES_SERVICE` são criados (*init*) e conectados (*connect*) na parte de inicialização (*initialize*) do módulo principal `MHS` da especificação.

III.4.4 Modelagem do Sistema de Transferência de Mensagens (MTS)

III.4.4.1 O Sistema de Transferência de Mensagens

O Sistema de Transferência de Mensagens (MTS) é o meio para troca de mensagens na base armazena-envia. A mensagem submetida por um usuário (originador) é transferida através do MTS e entregue a um ou mais usuários (destinatários).

As recomendações X.400 [15] descrevem o MTS utilizando um modelo abstrato a fim de definir os serviços fornecidos pelo MTS como um todo - o serviço abstrato do MTS. Este modelo é, mais tarde, refinado de forma a definir sua estrutura interna.

O MTS é modelado como um objeto, cujo comportamento geral pode ser descrito sem referência à sua estrutura interna. Os serviços oferecidos pelo objeto MTS se tornam disponíveis em suas *portas*. Um tipo de porta representa uma classe de serviços oferecidos pelo objeto MTS.

O usuário do MTS é também modelado como um objeto, que obtém os serviços oferecidos pelo MTS através de uma porta correspondente a uma porta do MTS do mesmo tipo. Um tipo de porta corresponde a um conjunto de *operações abstratas*; aquelas que podem ser invocadas pelo objeto MTS (e executadas pelo usuário do MTS) e aquelas que podem ser invocadas pelo usuário do MTS (e executadas pelo MTS).

Antes que objetos possam solicitar operações, é preciso que haja entre eles uma

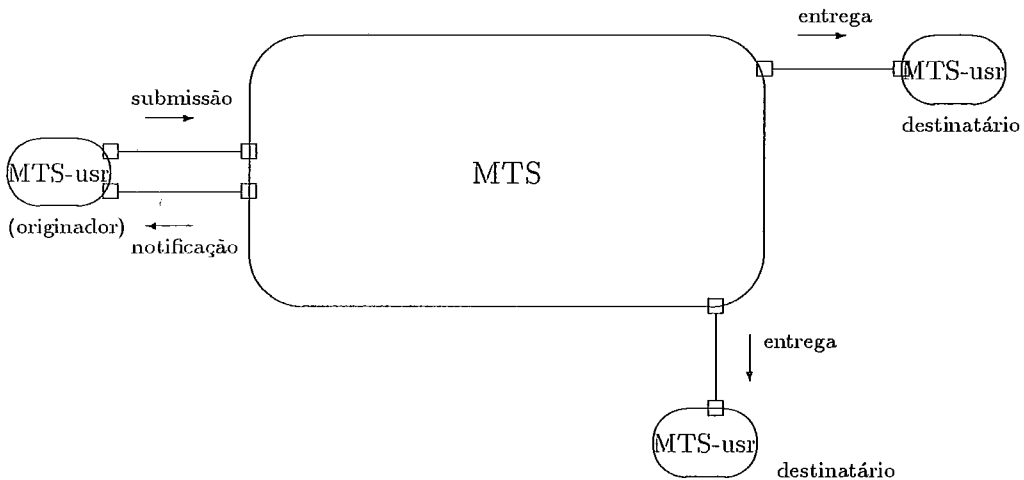


Figura III.8: Modelo do MTS

associação abstrata. O estabelecimento de uma associação entre objetos define o relacionamento entre eles até que a associação seja terminada. Uma associação é sempre terminada pelo iniciador da associação. Durante o estabelecimento da associação, os objetos validam as identificações (credenciais) um do outro e são estabelecidos o *contextos de aplicação* e o *contexto de segurança* da associação.

O objeto MTS suporta três classes de operações (e, portanto, três tipos de portas) a seus usuários: Operações de Submissão (Submission), de Entrega (Delivery) e Administração (Administration). Através destas portas é realizado o Protocolo P3.

As operações de submissão habilitam o usuário do MTS a submeter mensagens ao MTS para transferência e entrega a um ou mais usuários destinatários e a enviar sondagens que testam a entregabilidade de uma certa mensagem.

As operações de entrega habilitam o usuário do MTS a entregar a seu usuário mensagens e notificações de entrega e de não-entrega relativas a mensagens e sondagens.

As operações de administração habilitam ao usuário do MTS a alterar informações a seu respeito ou relativas à sua capacidade de recepção de mensagens, e habilitam tanto MTS quanto seu usuário a alterarem suas credenciais.

Uma mensagem submetida por um usuário do MTS via operação de sub-

missão, normalmente será entregue a um ou mais usuários via operação de entrega. O usuário originador pode escolher ser notificado da entrega da mensagem via operação de entrega.

III.4.4.2 Refinamento do MTS

O MTS é constituído de uma coleção de Agentes de Transferência de Mensagens (MTA), que no modelo abstrato são chamados "objetos MTA". Os MTAs cooperam entre si para formar o MTS e oferecer o serviço de transferência de mensagens a seus usuários. São os MTAs que executam as funções do MTS, isto é, transferem mensagens, sondagens notificações, geram notificações e realizam conversões de conteúdo.

Os objetos MTA também possuem portas, algumas das quais são exatamente aquelas visíveis na fronteira do objeto MTS, isto é, portas de submissão, entrega e administração. Entretanto, os MTAs possuem outro tipo de porta, a porta de transferência, através da qual ocorrem as operações de transferência e está relacionada à distribuição do serviço abstrato do MTS entre MTAs, e não são visíveis na fronteira do objeto MTS.

As operações de transferência habilitam um MTA a transferir mensagens, sondagens e notificações a outro MTA. Em geral, uma mensagem, sondagem ou notificação pode ser transferida várias vezes entre diferentes MTAs para alcançar seu destino. Através das portas de transferência é realizado o Protocolo P1.

III.4.4.3 Funcionamento do MTA

A descrição dos procedimentos de um MTA é baseado no modelo mostrado nas Figuras III.10 a III.13 a seguir. Deve-se notar que o modelo é incluído apenas para fins de exposição, não pretendendo impor um modelo para a implementação de um MTA.

O funcionamento do MTA é descrito utilizando-se módulos e procedimentos através dos quais é definido sua interação com usuários, sua interação com outros MTAs e sua operação distribuída.

O funcionamento do MTA é orientado a eventos e este se mantém inativo até que um evento seja detectado em uma de suas portas.

Os módulos que definem o funcionamento do MTA são mostrados na Figura III.11.

O módulo "deferred-delivery" é invocado pelos módulos "message-submission" e "message-in", que passam uma mensagem vinda do usuário ou de outro MTA,

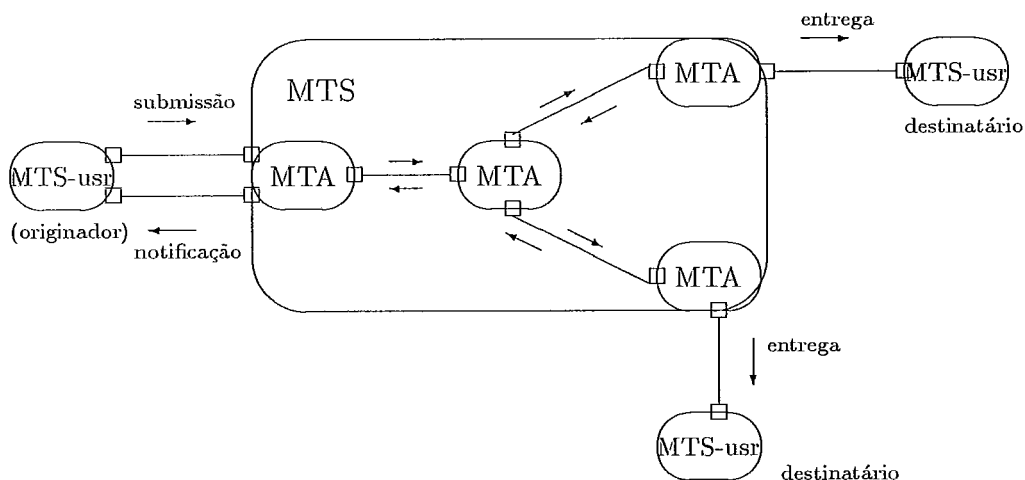


Figura III.9: Refinamento do MTS

respectivamente. É verificado se a mensagem solicita entrega pré-datada, podendo a mensagem ser retida para entrega na data determinada. Terminado o processamento, este módulo invoca o módulo "main", passando a mensagem.

O módulo "main" realiza processamento de mensagens ou sondagens que chegam ao MTA. A Figura III.11 mostra o relacionamento entre este módulo e os módulos que o invocam ou por ele invocados. Este módulo está sujeito à invocação pelos seguintes módulos:

- módulo "probe-in", que passa uma sondagem;
- módulo "deferred-delivery, que passa uma mensagem;
- módulo "probe", que passa uma sondagem;

Em caso de erro ou de necessidade de envio de notificação de entrega, o módulo "main" também pode ser invocado pelo seguintes módulos:

- módulo "message-out", que passa uma mensagem indicando o problema encontrado;
- módulo "probe-out", que passa uma sondagem indicando o problema encontrado;

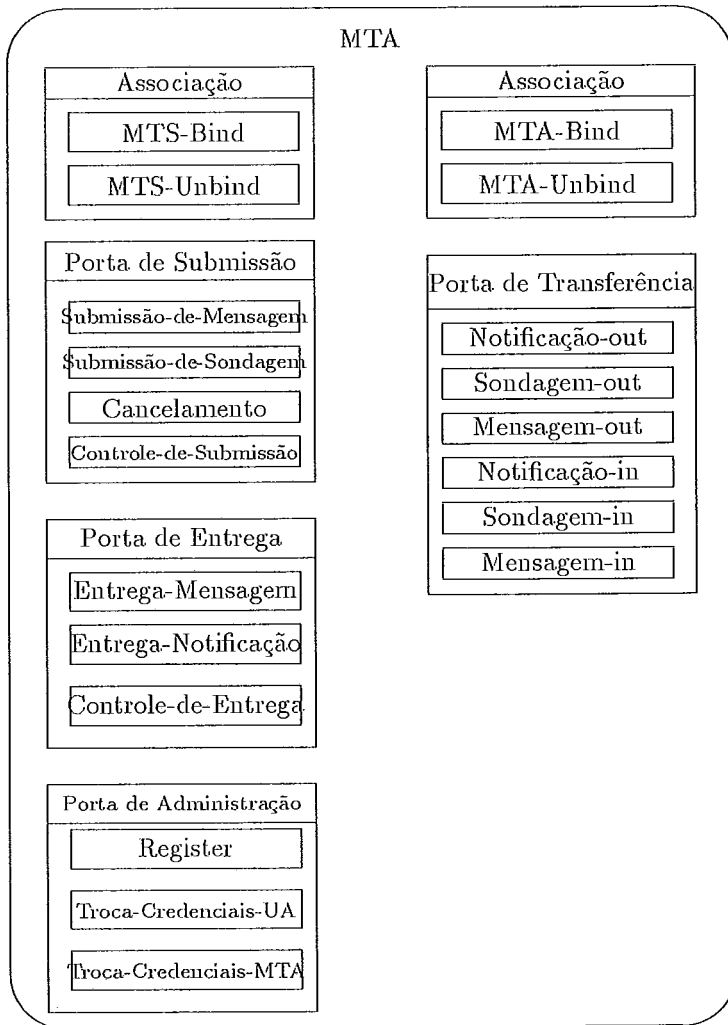


Figura III.10: Portas do MTA

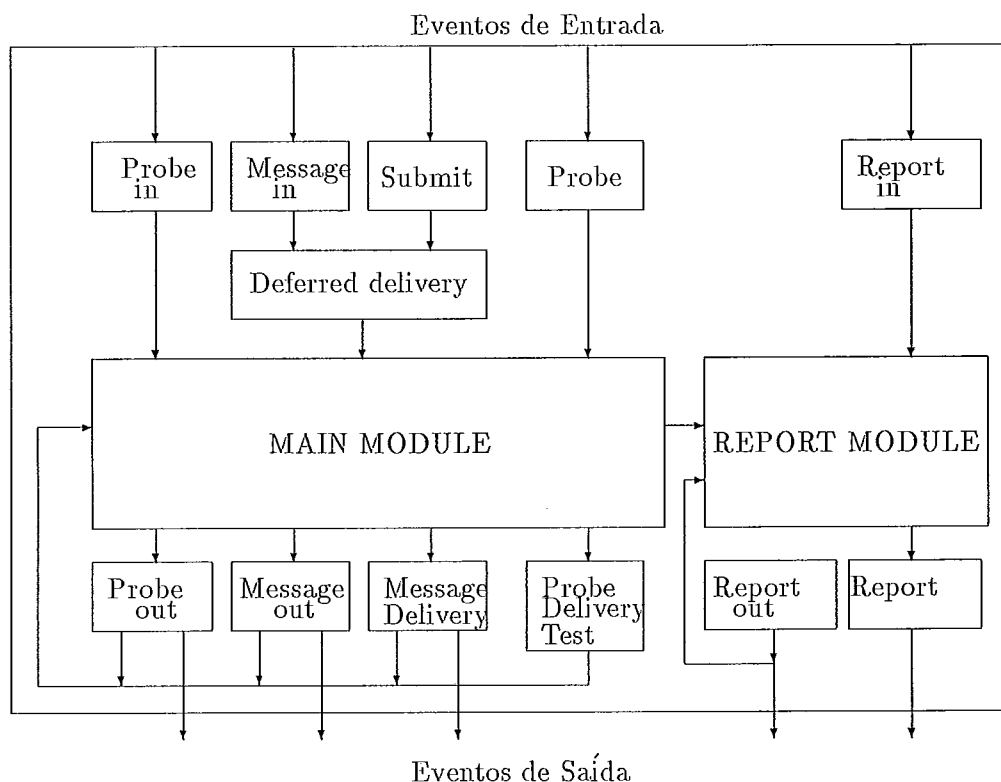


Figura III.11: Funcionamento do MTA

- módulo "message-delivery", que passa uma mensagem indicando os problemas encontrados para a geração de uma notificação de não-entrega;
- módulo "probe-delivery-test", que passa uma sondagem indicando os problemas encontrados para a geração de uma notificação de não-entrega.

O módulo "main" contém procedimentos que, coletivamente, suportam as seguintes funções:

- Processamento de "trace" [15];
- Roteamento;
- Redirecionamento de mensagens;
- Conversão de Conteúdo;
- Expansão de Listas de Distribuição;
- Replicação de Mensagens;
- Resolução de Endereços.

Como resultado deste módulo, uma ou mais réplicas da mensagem são criadas juntamente com instruções apropriadas. Dependendo da natureza destas instruções, o módulo "main" então invoca:

- módulo "message-out", para o qual é passado cada mensagem com instrução de transferência.
- módulo "probe-out", para o qual é passado cada sondagem com instrução de transferência.
- módulo "message-delivery", para o qual é passado cada mensagem com instrução de entrega para cada destinatário local.
- módulo "probe-delivery-test", para o qual é passado cada mensagem com instrução de entrega para cada destinatário.
- módulo "report", para o qual é passado cada mensagem ou sondagem com instrução globais à mensagem para geração de notificação e/ou instruções específicas para cada destinatário.

Para cada mensagem ou sondagem, o módulo "main" executa os seguintes procedimentos:

- (1) Front-End: Este procedimento inicializa as informações de "trace" e realiza várias checagens na mensagem. Por exemplo, verifica expiração da mensagem e detecta loops de roteamento.

Em caso de erro, o procedimento gera instruções para a geração de uma notificação de não-entrega e o processamento continua no procedimento 9. Caso contrário, continua no procedimento 2.

- (2) Decisão de Roteamento e Conversão: Gera instruções para roteamento e conversão para cada destinatário da mensagem. São instruções que dirigirão a mensagem (sondagem) pelos demais procedimentos.

Se é detectada a necessidade de redirecionamento da mensagem, isto é, a mensagem não pode ser entregue para um determinado destinatário e existe um destinatário alternativo para o qual a mensagem pode ser enviada, então o processamento continua no procedimento 3.

- (3) Redirecionamento: Verifica se existe loop de redirecionamento, caso contrário, altera o envelope da mensagem apropriadamente e o processamento continua no procedimento 2. Em caso de loop, continua no procedimento 8.

- (4) Despachante: Verifica as instruções geradas pelos procedimentos anteriores e passa o controle para o primeiro dos seguintes procedimentos que for aplicável:

- replicante (procedimento 5);
- conversão (procedimento 6);
- expansão de DL (procedimento 7);
- saída (procedimento 10).

- (5) Replicante: Replica a mensagem conforme solicitado pelas instruções geradas para cada destinatário no procedimento 2. Para cada réplica, o processamento continua individualmente no procedimento 4.
- (6) Conversão: é executado para cada mensagem ou sondagem que necessite. Em caso de sucesso, o processamento continua no procedimento 4. Caso contrário, é gerada instrução indicando erro de conversão e o processamento continua no procedimento 8.
- (7) Expansão de DL: Executa expansão de Listas de Distribuição. Em caso de sucesso, o processamento continua no procedimento 2 para que os destinatários adicionados à mensagem possam ser tratados apropriadamente.
- Se é gerada uma cópia da mensagem com instrução para geração de uma notificação de entrega, seu processamento continua no procedimento 9.
- No caso de sondagem, esta terá instrução para geração de uma de entrega, seu processamento continua no procedimento 9. Em caso de erro, o processamento continua no procedimento 8 com instruções para geração de uma notificação de entrega.
- (8) Processamento de Erros: Este procedimento é solicitado quando é detetada a impossibilidade de entrega de uma mensagem/sondagem. Este procedimento tenta determinar outro método de entrega ou um destinatário alternativo. Em caso de sucesso, o processamento continua no procedimento 2, ou no procedimento 9 se é verificada a possibilidade de redirecionamento.
- (9) Saída para Report: Os procedimentos do módulo main terminam neste e o módulo "report" é chamado para a geração de notificação para a mensagem ou sondagem.
- (10) Saída: Ponto de saída do módulo "main".

A Figura III.12 mostra o fluxo de informação no módulo "main".

O módulo "report" pode ser invocado pelos seguintes módulos:

- módulo "report-in", que passa uma notificação;

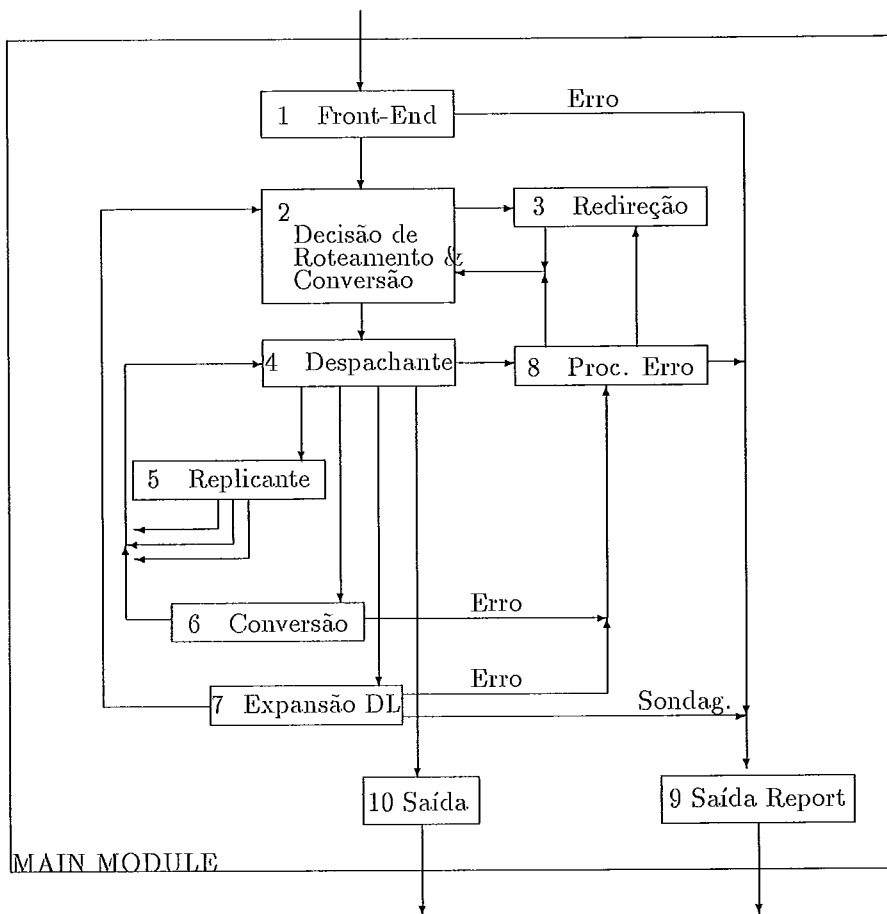


Figura III.12: Funcionamento do Módulo Principal

- módulo "main", que passa uma mensagem ou notificação com instruções para geração de notificação;
- módulo "report-out", que passa uma notificação com problemas;

Se os procedimentos internos deste módulo encontram uma condição de erro, não é gerada qualquer saída. Em caso de sucesso, este módulo invoca o módulo "report-out" ou "report-delivery", passando uma notificação a ser transferida a outro MTA ou entregue a um usuário, respectivamente.

O módulo "report" executa os seguintes procedimentos:

- (1) Front-End: é o primeiro procedimento executado neste módulo para notificações vindas do módulo "report-in". Este procedimento realiza inicialização das informações de "trace" e várias outras verificações iniciais [15].

Em caso de erro, a notificação é descartada e o procedimento termina. Caso contrário, o processamento continua no procedimento 3.

- (2) Geração de Notificação: é o primeiro procedimento executado neste módulo para mensagens e sondagens. Este procedimento gera a notificação de entrega ou não entrega conforme as instruções trazidas pela mensagem ou sondagem.

Em caso de erro, a mensagem ou sondagem é descartada e o procedimento termina. Caso contrário, o processamento continua no procedimento 3.

- (3) Roteamento de Notificação: Gera instruções de roteamento para a notificação. Em caso de erro, a notificação é descartada e o procedimento termina. Caso contrário, as informações de trace são ativadas para indicarem a passagem por este MTA e o processamento termina.

A Figura III.13 mostra o fluxo de informação no módulo "report".

Os demais módulos (probe-in, message-in, submit, probe, reportin, probe-out, message-out, message-delivery, probe-deliver-test, report-out e report-delivery) correspondem às portas de entrada e saída de mensagens, sondagens e notificações e realizam os Protocolos P3 e P1. Estes protocolos são descritos a seguir.

III.4.4.4 O Protocolo P3

As recomendações X.411 e X.419 definem abstratamente o Protocolo P3 utilizando as macros do ROS. As macros MTSBind e MTSUnbind definem os parâmetros para estabelecimento e término da associação entre a entidade MTA e seu usuário (UA

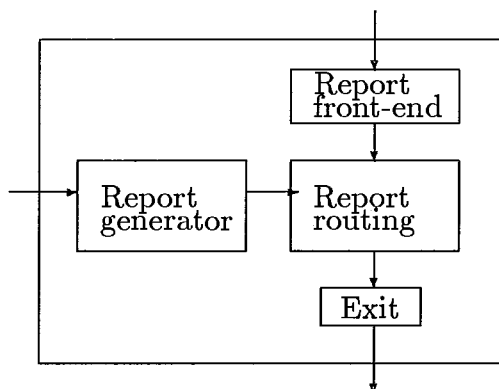


Figura III.13: Funcionamento do Módulo Report

ou MS). Os parâmetros das operações são também definidos utilizando macros ROS e tipos ASN.1 [15].

Cada classe de serviço foi agrupado em um ASE (Application Service Element): o MSSE (Message Submission Service Element) , que corresponde aos serviços de submissão de mensagens , o MDSE (Message Delivery Service Element), que corresponde aos serviços de entrega de mensagens , o MASE (Message Administration Service Element), que corresponde aos serviços de administração. Estes ASEs unidos aos ASEs comuns aos protocolos de aplicação (ACSE, ROSE, RTSE) formam o conjunto de ASEs que definem os Contextos de Aplicação (ACs) do Protocolo P3. Existem quatro contextos definidos neste protocolo:

- *mts-access*: composto dos ASEs: MSSE, MDSE, MASE, ACSE e ROSE. Este contexto de aplicação é obrigatório.

```
mts-access APPLICATION-CONTEXT
APPLICATION SERVICE ELEMENT {acse}
BIND MTSBind
UNBIND MTSUnbind
REMOTE OPERATIONS {rose}
INITIATOR CONSUMER OF {msse, mdse, mase}
ABSTRACT SYNTAXES {
    id-as-acse -- of ACSE
    id-as-msse -- of MSSE, including ROSE
    id-as-mdse -- of MDSE, including ROSE
    id-as-ms -- of MTSBind, MTSUnbind --}
::= id-ac-mts-access
```

- *mts-forced-access*: composto dos ASEs: MSSE, MDSE, MASE, ACSE e ROSE. Este contexto de aplicação é obrigatório.

```
mts-forced-access APPLICATION-CONTEXT
  APPLICATION SERVICE ELEMENT {aCSE}
  BIND          MTSBind
  UNBIND        MTSUnbind
  REMOTE OPERATIONS {rOSE}
  RESPONDER CONSUMER OF {mSSE, mDSE, mASE}
  ABSTRACT SYNTAXES {
    id-as-acse      -- of ACSE
    id-as-msse      -- of MSSE, including ROSE
    id-as-mdse      -- of MDSE, including ROSE
    id-as-mase      -- of MASE, including ROSE
    id-as-ms        -- of MTSBind, MTSUnbind --}
  ::= id-ac-mts-forced-access
```

- *mts-reliable-access*: composto dos ASEs: MSSE, MDSE, MASE, ACSE, ROSE e RTSE. Este contexto de aplicação é opcional.

```
mts-reliable-access APPLICATION-CONTEXT
  APPLICATION SERVICE ELEMENT {aCSE, rTSE}
  BIND          MTSBind
  UNBIND        MTSUnbind
  INITIATOR CONSUMER OF {mSSE, mRSE, mASE}
  ABSTRACT SYNTAXES {
    id-as-acse      -- of ACSE
    id-as-msse      -- of MSSE, including ROSE
    id-as-mrse      -- of MRSE, including ROSE
    id-as-mase      -- of MASE, including ROSE
    id-as-ms        -- of MSBind, MSUnbind,
                    -- including RTSE }
  ::= id-ac-mts-reliable-access
```

- *mts-forced-reliable-access*: composto dos ASEs: MSSE, MDSE, MASE, ACSE, ROSE e RTSE. Este contexto de aplicação é opcional.

```
mts-forced-reliable-access APPLICATION-CONTEXT
```

```
APPLICATION SERVICE ELEMENTS {aCSE, rTSE}
BIND          MTSBind
UNBIND       MTSUnbind
RESPONDER    CONSUMER OF {mSSE, mRSE, mASE}
ABSTRACT SYNTAXES {
    id-as-acse      -- of ACSE
    id-as-msse     -- of MSSE, including ROSE
    id-as-mrse     -- of MRSE, including ROSE
    id-as-mase     -- of MASE, including ROSE
    id-as-ms       -- of MSBind, MSUnbind,
                    -- including RTSE }
 ::= id-ac-mts-forced-reliable-access
```

O Protocolo P3 consiste de três etapas:

- Estabelecimento da Associação: Conforme definido pela macro MTSBind. Os dados da macro contem a identificação da entidade remota e estabelece o contexto de segurança no qual as operações subseqüentes irão ocorrer. Ao fim da operação MTSBind os sistemas fim haverão validado a identificação mutuamente e conhecerão a(s) chave(s) a serem utilizadas em funções de segurança. No Protocolo P3 tanto o UA como o MTA tem direito de iniciar o estabelecimento de associação.
- Execução de Operações: Havendo a associação, as entidades podem então solicitar a execução das operações. As operações do Protocolo P3 são executadas assincronamente.
- Término da Associação: Se a entidade iniciadora da associação não necessita mais dos serviços da entidade remota, e operações a serem respondidas, ou seja, "Results" pendentes, a associação é terminada conforme a macro MTSUnbind. A realização dos serviços definidos pelas macros de ROS para o Protocolo P3 do ACSE, ROSE e RTSE e o uso dos parâmetros dos serviços destes ASEs é idêntico ao Protocolo P7 descrito em III.3.3.

III.4.4.5 Protocolo P1

As recomendações X.411 e X.419 definem abstratamente o Protocolo P1 utilizando as macros do ROS. As macros MTABind e MTAUnbind definem os parâmetros para estabelecimento e término da associação entre entidades MTA. Os parâmetros das operações são também definidos utilizando macros ROS e tipos ASN.1 [15].

Os serviços de transferência foram agrupados em um ASE (Application Service Element): o MTSE (Message Transfer Service Element). Este ASE unido aos ASEs comuns aos protocolos de aplicação ACSE e RTSE formam o conjunto de ASEs que definem os Contextos de Aplicação (ACs) do Protocolo P1. Existem três contextos definidos neste protocolo:

- *mts-transfer-protocol-access*: Este AC é definido para intercooperação com implementações de Protocolo P1 em conformidade com a Recomendação X.411 1984 [7]. Neste AC, a sintaxe abstrata do MTSE deve ser restrita à sintaxe definida na Recomendação X.411 1984. As diferenças entre as duas recomendações são destacadas na Recomendação X.411 1988 e listadas no Anexo C desta mesma recomendação [15]. Este AC é suportado pelo RTSE em modo "X410-1984". O suporte à este AC é obrigatório.

```
mts-transfer-protocol-1984 INTEGER ::= 12
```

- *mts-transfer-protocol*: Este AC é definido para intercooperação com implementações que suportam a funcionalidade do Protocolo P1 1988 via sistemas em conformidade com a Recomendação X.411 1984. Este AC é suportado pelo RTSE em modo "X410-1984". O suporte à este AC é obrigatório.

```
mts-transfer-protocol INTEGER ::= 12
```

- *mts-transfer*: O suporte deste AC é opcional e definido para as implementações em conformidade a Recomendação X.411 1988, sem qualquer imposição quanto ao suporte à sintaxe definida na Recomendação X.411 1984. Acredita-se que ao longo do tempo, a maioria dos sistemas migrarão para o suporte deste AC. Uma futura versão das Recomendações X.400 provavelmente tornará o suporte a este AC obrigatório como parte de uma estratégia de migração que permita suporte à funcionalidade estendida deste AC e maximize a intercooperação entre os sistemas.

```
mts-transfer APPLICATION-CONTEXT
```

```
APPLICATION SERVICE ELEMENTS {acse, rtse, mtse}
```

```
BIND MTABind
```

```
UNBIND MTAUnbind
```

```
ABSTRACT SYNTAXES {
```

```
id-as-acse -- of ACSE
```

```
id-as-mts-rtse    -- of MTABind, MTAUnbind e RTSE
id-as-mtse        -- of MTSE }
 ::= id-ac-mts-transfer
```

O Protocolo P1 consiste de três etapas:

- Estabelecimento da Associação: Conforme definido pela macro MTABind. Os dados da macro contem a identificação da entidade remota e estabelece o contexto de segurança no qual as operações subseqüentes irão ocorrer. Ao fim da operação MTABind os sistemas fim haverão validado a identificação mutuamente e conhecerão a(s) chave(s) a serem utilizadas em funções de segurança. Associações entre dois MTAs são estabelecidas seguindo acordo bilateral sobre os seguintes aspectos:
 - a) O número máximo de associações que pode existir simultâneamente;
 - b) Se a comunicação entre MTA é do tipo "monólogo" ou "diálogo alternado";
 - c) Qual o contexto de aplicação utilizado;
 - d) Qual MTA tem a responsabilidade de estabelecer associações;
 - e) Se haverão associações permanentemente estabelecidas, ou se serão estabelecidas e terminadas conforme necessário.
- Execução de Operações: Havendo a associação, as entidades podem então solicitar a execução das operações de transferência.
- Término da Associação: Se a entidade iniciadora da associação não necessita mais dos serviços da entidade remota, e não existem operações a serem respondidas, a associação é terminada conforme a macro MTAUnbind.

Realização das Operações do Protocolo P1

A realização de todas as operações do Protocolo P1 é suportada pelo RTSE definido em [24,31], incluindo o estabelecimento e término de associação. O RTSE provê a transferência confiável de Unidades de Dados dos Protocolos de Aplicação (APDUs). Ele assegura que cada APDU é completamente transferida ou que o transmissor é avisado em caso de condições de exceção. O RTSE recupera falhas e minimiza a quantidade de retransmissões necessárias à recuperação.

O RTSE utiliza os serviços do ACSE e da Camada de Apresentação. O uso do modo X.410-1984 do RTSE implica o uso do ACSE em modo X.410-1984 e o uso da Camada de Apresentação no mesmo modo. O mesmo ocorre se o RTSE é utilizado em modo normal.

A realização do serviço definido pela macro `MTABind` se dá através do serviço `RT-OPEN`. Os parâmetros deste serviço são utilizados da seguinte forma:

- **Mode:**

Se o RTSE é utilizado em modo X.410-1984, o parâmetro "mode" da primitiva `RT-OPEN.Request` deve ter o valor "X.410-1894 mode". Se o RTSE é utilizado em modo normal, o parâmetro deve ter o valor "normal mode".

- **Application Protocol:**

O parâmetro "application-protocol" primitiva `RT-OPEN.Request` deve propor o contexto de aplicação. Se o RTSE é utilizado em modo X.410-1984, o parâmetro deve ter o valor do identificador do contexto de aplicação "mts-transfer-protocol" ou o valor do identificador do contexto de aplicação "mts-transfer-protocol-1984". Se o RTSE é utilizado em modo normal, o parâmetro deve ter o valor do identificador do contexto de aplicação "mts-transfer".

- **User Data:**

O valor do tipo definido na cláusula `ARGUMENT` da macro `MTABind` é mapeado no parâmetro "user-data" da primitiva `RT-OPEN.Request` pelo iniciador da associação.

Se o respondedor fornece o valor "accepted" no parâmetro "result" da primitiva `RT-OPEN.Response`, então o valor do tipo definido na cláusula `RESULT` da macro `MTABind` é mapeado no parâmetro "userdata" da primitiva `RT-OPEN.Response`.

Em caso de erro, o respondedor fornece o valor "reject (permanent)" ou "reject (transient)" no parâmetro "result" da primitiva `RT-OPEN.Response`. Em caso de "reject (permanent)", o parâmetro "user-data" da primitiva `RT-OPEN.Response` deve ter o valor "authentication-error" ou "unacceptable-dialogue-mode".

A realização do serviço definido pela macro `MTAUnbind` se dá através do serviço `RT-CLOSE`. Em modo X.410-1984, este serviço não possui parâmetros. Em modo normal, os parâmetros deste serviço não são utilizados.

A realização dos serviços de transferência de mensagem, sondagem e notificação se dá através do serviço `RT-TRANSFER`. O RTSE somente pode disparar uma primitiva `RT-TRANSFER.Request` se possui a vez de transmissão e se não existem primitivas `RT-TRANSFER.Confirm` pendentes. Os parâmetros deste serviço são utilizados da seguinte forma:

- APDU:

O valor da MTS-APDU deve ser mapeado no parâmetro "APDU" da primitiva RT-TRANSFER.Request pelo transmissor.

Para o serviço de transferência de mensagem, a MTS-APDU é uma mensagem, que é definida pela cláusula ARGUMENT da macro que define a operação MessageTransfer [15].

Para o serviço de transferência de sondagem, a MTS-APDU é uma sondagem, que é definida pela cláusula ARGUMENT da macro que define a operação ProbeTransfer [15].

Para o serviço de transferência de notificação, a MTS-APDU é uma notificação, que é definida pela cláusula ARGUMENT da macro que define a operação ReportTransfer [15].

- Transfer Time:

O valor este parâmetro é especificado por regras locais do transmissor. Pode estar relacionado à prioridade da APDU.

O gerenciamento da vez de transmissão é feito através dos serviços RT-TURN-PLEASE e RT-TURN-GIVE. O MTSE deve possuir a vez antes do uso do serviço RT-TRANSFER.

O MTSE sem a vez pode disparar uma primitiva RT-TURNPLEASE.Request com o parâmetro "priority" que reflete a prioridade mais alta de APDU aguardando transferência.

O MTSE com a vez pode disparar uma primitiva RT-TURN-GIVE.Request quando não tem mais APDUs a transferir. Ele deve disparar uma primitiva RT-TURN-GIVE.Request em resposta a uma RT-TURNPLEASE.Indication quando não tem mais APDUs a transferir com prioridade igual ao menor àquela indicada na primitiva RT-TURNPLEASE.Indication. Se ele ainda tem APDUs de menor prioridade a transferir, então ele deve disparar uma primitiva RT-TURNPLEASE.Request indicando a mais alta prioridade destas APDUs.

Um MTSE dispara a primitiva RT-TURN-PLEASE.Request somente se não possui a vez. Se o iniciador da associação propôs comunicação do tipo "monólogo" e a vez inicial é do iniciador (parâmetro "initial-turn"), o serviço RT-TURN-PLEASE não deve ser utilizado. Os parâmetros deste serviço são utilizados da seguinte forma:

- Priority:

Reflete a prioridade mais alta de APDU aguardando transferência.

A mais alta prioridade é a de valor zero, e é reservada para a ação de término de associação pelo iniciador.

A prioridade de valor "um" deve ser designada a mensagens cujo campo prioridade (do envelope [15]) tem o valor "urgente". Este valor de prioridade também pode ser designado a sondagens e a notificações.

A prioridade de valor "dois" deve ser associada a mensagens cujo campo prioridade (envelope) tem o valor "normal".

A prioridade de valor "três" deve ser associada a mensagens cujo campo prioridade (envelope) tem o valor "não-urgente".

Se existe mais de uma associação estabelecida entre dois MTAs, as MTS-APDUS podem ser atribuídas às associações de acordo com suas prioridades. Várias associações podem ser utilizadas para transferir MTS-APDUs de mesma prioridade. Em qualquer associação, as MTS-APDUs de maior prioridades são transferidas antes das de menor prioridade. MTS-APDUs de mesma prioridade são enviadas segundo a política "first-in first-out".

Um MTSE dispara a primitiva RT-TURN-GIVE.Request para obter a vez de seu par. Isto somente pode ser feito se ele não possui a vez. Se o iniciador da associação propôs comunicação do tipo "monólogo" e a vez inicial é do iniciador (parâmetro "initial-turn"), o serviço RT-TURN-GIVE não deve ser utilizado.

Este serviço não possui parâmetros.

O processo de aplicação é o usuário dos demais serviços do RTSE, RT-P-ABORT e RT-U-ABORT.

Através do serviço RT-P-ABORT, o processo de aplicação é avisado que a associação não pode ser mantida. Este serviço não possui parâmetros

Através do serviço RT-U-ABORT, o processo de aplicação pode abortar uma associação de aplicação. Este serviço pode ser utilizado tanto pelo iniciador como pelo respondedor da associação. Os parâmetros deste serviço não são utilizados em modo normal.

O serviço RT-U-ABORT não está disponível em modo X.410-1984.

Uso dos serviços de Apresentação

Como visto acima, dependendo do contexto de aplicação, o Protocolo P1 utiliza os serviços da Camada de Apresentação em modo X.410-1984 ou normal.

Em modo X.410-1984, um único contexto de apresentação ("default") é utilizado para a conexão de apresentação. Este contexto de apresentação inclui uma única sintaxe abstrata para todos os ASEs incluídos no contexto de aplicação.

Em modo normal, um diferente contexto de apresentação (presentation-context) é utilizado para cada sintaxe abstrata incluída no contexto de aplicação.

O endereçamento da camada de Apresentação não é usado para o Protocolo P1 quando o RTSE é utilizado em modo X.410-1984.

O ACSE é o único usuário dos serviços P-CONNECT, P-RELEASE, P-UABORT e P-P-ABORT.

O RTSE é o único usuário dos serviços P-ACTIVITY-START, P-DATA, P-MINOR-SYNCRONIZE, P-ACTIVITY-END, P-ACTIVITY-INTERRUPT, P-ACTIVITY-DISCARD, P-U-EXCEPTION-REPORT, P-ACTIVITY-RESUME, P-P-EXCEPTION-REPORT, P-TOKEN-PLEASE e P-CONTROL-GIVE.

Uso dos Serviços de Sessão e Transporte

O uso do RTSE requer as seguintes unidades funcionais do Serviço de Sessão: Kernel, Half-duplex, Exceptions. Minor-synchronize e Activity-management.

O endereçamento da camada de Apresentação não é usado para o Protocolo P1 quando o RTSE é utilizado em modo X.410-1984. Isto é, não é passado um endereço de sessão na "Connect SPDU" da camada de Sessão.

Quanto ao Serviço de Transporte, suporte à Classe 0 é obrigatório. Suporte à outras classes é opcional.

III.4.4.6 Modelagem Estelle do MTA

Na modelagem Estelle, as instâncias MTA são conectadas aos vários pontos de interação do sub-módulo CCASE, que por sua vez é conectado ao sub-módulo PRES-SERVICE, que modela o serviço de apresentação. São necessários vários pontos de interação do sub-módulo CCASE por que, em sua operação distribuída, um MTA pode se comunicar com vários outros e com várias entidades usuárias de seus serviços (UA e MS). Nesta modelagem, o módulo MTA é refinado em três sub-módulos:

P3_PROVIDER_TYPE : Possui o atributo *activity* e modela a interação do MTA com seu usuário. O usuário do MTA tem conectado a si um sub-módulo do tipo MTS_ACCESS_TYPE que provê o acesso ao MTA. Logo, coletivamente, os módulos P3_PROVIDER_TYPE e MTS_ACCESS_TYPE

modelam o Protocolo P3. A definição do módulo P3_PROVIDER_TYPE é mostrada abaixo.

```
module P3_PROVIDER_TYPE activity (n: integer);
  ip      P:  OI_P3_interface (p3);
          C11: ROSE_interface (user);
          C12: ACSE_interface (user);
  export Release_me: boolean; num: integer;
  end;

body P3_PROVIDER_BODY for P3_PROVIDER_TYPE;
```

Este sub-módulo possui um parâmetro de entrada, n, do tipo inteiro que identifica o módulo, e que corresponde á identificação do usuário para o MTA. Além disso, o sub-módulo exporta duas variáveis chamadas "num" e "release_me". A variável num é do tipo inteiro e exporta a identificação do sub-módulo passada como parâmetro de entrada. A variável release_me é do tipo booleano e exporta uma indicação de que o sub-módulo não é mais necessário e pode ser descartado (declaração release); ou seja, a entidade usuária não está mais conectada ao MTA.

P10I_PROVIDER_TYPE : Possui o atributo *activity* e modela a operação interna do MTA no processamento de mensagens, sondagens e notificações que transitam por ele. Este módulo compreende todos os procedimentos internos do MTA descritos em III.3.4.3. A definição deste módulo é mostrada abaixo.

```
module P10I_TYPE activity;
  ip      U: OI_P3_interface (oi);
          P: OI_P1_interface (oi);
  end;

body p10i_body for p10i_type;
```

Este módulo tem dois pontos de interação externos chamados U (do tipo OI_P3_interface) e P (do tipo OI_P1_interface), através dos quais, este sub-

módulo recebe e envia mensagens do/para os sub-módulos P3_PROVIDER_-TYPE e P1_PM_TYPE, respectivamente. A definição destas interfaces é mostrada abaixo.

```
channel OI_P3_interface (oi, p3);
  by oi:   ToUser      (arg: asn1_def);
  by p3:   FromUser    (arg: asn1_def);

channel OI_P1_interface (oi, p1);
  by oi:   TransfOut   (arg: asn1_def);
  by p1:   TransfIn    (arg: asn1_def);
          TransfOutErr (arg: asn1_def);
```

Cada transição deste sub-módulo descreve um conjunto dos procedimentos do modelo apresentado acima. Este sub-módulo possui uma ponto de interação interno cujas interações definem qual conjunto de procedimentos será executado. A definição deste ponto de interação é mostrada abaixo.

```
channel P1OI_Internal (A,B);
  by A,B:
    Main (arg: asn1_def);
    ProbeTst (arg: asn1_def);
    RoutingDecision (arg:asn1_def);
    ConversionDecision (arg: asn1_def);
    ErrorProcessing (arg: asn1_def; rec:asn1_def);
    Redirection (arg: asn1_def; rec: asn1_def);
    Dispatcher (arg: asn1_def);
    Splitter (arg: asn1_def);
    Conversion (arg: asn1_def);
    Dl_Expansion (arg: asn1_def);
    Report (arg: asn1_def);
    ReportFrontEnd (arg: asn1_def);
    ReportRouting (arg: asn1_def);
```

ip

```
Intern_in: P1OI_Internal (A);  
Intern_out: P1OI_Internal (B);
```

P1PM_TYPE : Possui o atributo *activity* e modela a máquina de estados do Protocolo P1. Este sub-módulo possui dois pontos de interação externos chamados P (do tipo OI_P1_interface) e RT (do tipo RTSE_interface), respectivamente. Estes pontos de interação realizam as interfaces com os sub-módulos P1OI e CCASE respectivamente.

```
module P1PM_TYPE activity (n: integer);  
  ip    P:  OI_P1_interface (p1);  
        C2: RTSE_interface  (user);  
  export Release_me: boolean; num: integer;  
  end;  
  
  body  p1pm_body for p1pm_type;
```

Este sub-módulo possui um parâmetro de entrada, *n*, do tipo inteiro que identifica o módulo, e que corresponde à identificação do outro sub-módulo MTA que está se comunicando com este. Além disso, o sub-módulo exporta duas variáveis chamadas "num" e "release_me". A variável num é do tipo inteiro e exporta a identificação do sub-módulo passada como parâmetro de entrada. A variável release_me é do tipo booleano e exporta uma indicação de que o sub-módulo não é mais necessário e pode ser descartado (declaração release); ou seja, a entidade MTA remota não mais está conectada.

```
trans  
  provided exist P1PM: P1PM_TYPE  
  suchthat (P1PM.Release_me)  
  begin  
    forone P1PM: P1PM_TYPE suchthat (P1PM.Release_me) do  
      begin  
        release P1PM;  
      end;  
    end;  
  end;
```

O sub-módulo MTA possui vários pontos de interação internos que permitem o diálogo entre as instâncias de módulos filhos. Estes pontos de interação são conectados (declaração connect) dinamicamente aos pontos de interação externos das instâncias de sub-módulos filhos durante a execução da especificação, e assim permitem que o módulo pai (MTA) tenha acesso aos eventos por eles disparados. A declaração dos pontos de interação internos é mostrada a seguir.

```
ip
  IDLE_OIH: OI_P3_interface (p3);
  IDLE_P3: array [1..max_usr] of OI_P3_interface (oi);

  IDLE_OIL: OI_P1_interface (p1);
  IDLE_P1: array [1..max_mta] of OI_P1_interface (oi);

  IDLE_C1: array [1..max_usr] of ACSE_interface (provider);
  IDLE_C2: array [1..max_mta] of RTSE_interface (provider);
```

Durante a inicialização do MTA somente a instância P1OI é criada e seus pontos de interação externos são conectados a dois pontos de interação do MTA para que este possa ter acesso a seus eventos de saída. As várias instâncias dos módulos P3_PROVIDER_TYPE e P1PM_TYPE, isto é, P3PR[i] e P1PM[i], respectivamente, são criadas dinamicamente de acordo com os eventos detectados nos pontos de interação externos do modulo MTA.

A Figura III.14 ilustra a arquitetura da modelagem Estelle do MTA.

Quando uma instância P3PR[i] é criada, seus pontos de interação externos AC[i] e RO[i] são ligados (attach) aos pontos de interação externos do MTA AC[i] e RO[i], e seu ponto de interação P é conectado (connect) ao ponto de interação do MTA IDLE_P3[i] para que este possa ter acesso aos eventos provenientes deste módulo filho. A criação de uma instância do módulo P3PR é mostrada abaixo.

```
trans
```

```
  when IDLE_OIH.ToUser
```

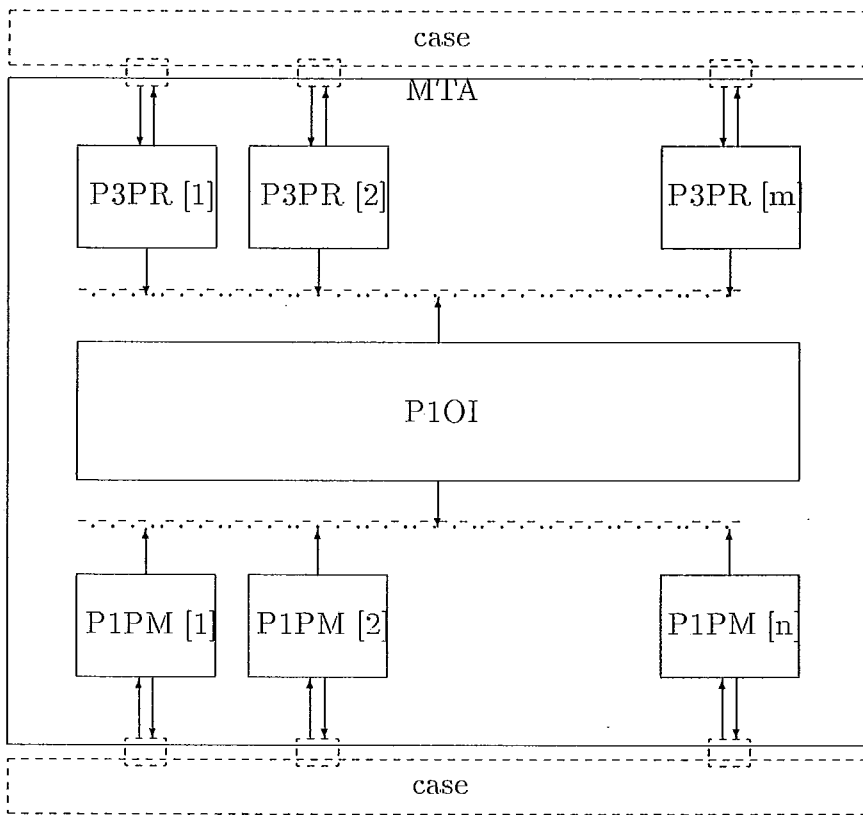



Figura III.14: Arquitetura da Modelagem Estelle do MTA

```
provided exist P3PR: P3_PROVIDER_TYPE
suchthat (P3PR.num = Get_n(arg, n))
begin
    output IDLE_P3[n].ToUser (arg);
end;

provided otherwise
begin
    init    P3PR[n]      with P3_PROVIDER_BODY (n);
    connect IDLE_P3[n]  to  P3PR[n].P;
    output  IDLE_P3[n].ToUser (arg);
    attach  C11[n]      to  P3PR[n].C11;
    attach  C12[n]      to  P3PR[n].C12;
end;
```

Analogamente, quando uma instância P1PM[i] é criada, seus pontos de interação externos RT[i] são ligados (attach) ao ponto de interação externo do MTA RT[i], e seu ponto de interação P é conectado (connect) ao ponto de interação do MTA IDLE_P1[i] para que este possa ter acesso aos eventos provenientes deste módulo filho. A criação de uma instância do módulo P1PM é mostrada abaixo.

trans

```
when IDLE_OIL.TransfOut
```

```
provided exist P1PM: P1PM_TYPE
suchthat (P1PM.num = Get_n(arg, n))
begin
    output IDLE_P1[n].TransfOut (arg);
end;
```

```
provided otherwise
```

```
begin
    init    P1PM[n]      with P1PM_BODY (n);
    connect IDLE_P1[n]  to  P1PM[n].P;
    output  IDLE_P1[n].TransfOut (arg);
```

```
attach C2[n] to P1PM[n].C2;  
end;
```

III.4.5 Modelagem dos Serviços Comuns de Aplicação (CASE)

Em um ambiente OSI, um processo de aplicação é representado por uma ou mais "entidades de aplicação". Cada entidade de aplicação representa um aspecto relevante do comportamento empreendido na comunicação entre os processos de aplicação. Por sua vez, uma entidade de aplicação é constituída por um Elemento de Usuário e uma série de Elementos de Serviço de Aplicação.

O elemento de usuário representa a parte do processo de aplicação que provê acesso aos serviços de aplicação. Os elementos de serviço de aplicação podem interagir entre si e utilizar os serviços de aplicação para realizar sua função. Os elementos de serviço de aplicação se dividem em duas categorias:

- Elementos de Serviço Comuns de Aplicação (CASE - Common Application Service Elements); e
- Elementos de Serviço Específicos de Aplicação (SASE - Specific Application Service Elements).

Os elementos de serviço específicos de aplicação fornecem funcionalidade específica. Nesta categoria se enquadram o MHS e seus ASEs descritos nas seções anteriores.

Os elementos de serviço comuns de aplicação fornecem suporte às diversas aplicações. Conforme visto anteriormente, são definidos três elementos de serviço comuns, a saber, o ACSE (Association Control Service Element), o ROSE (Remote Operations Service Element), e o RTSE (Reliable Transfer Service Element).

III.4.5.1 O Elemento de Serviço de Controle de Associação (ACSE)

O objetivo do Elemento de Serviço de Controle de Associação (ACSE) é prover facilidades básicas para o estabelecimento e controle de uma associação de aplicação entre duas entidades de aplicação que se comunicam através de uma conexão de apresentação. O ACSE é definido em [23,29].

O serviço do ACSE é definido por quatro primitivas de serviço:

- A-ASSOCIATE – Estabelece uma associação de aplicação.

- A-RELEASE – Causa o término de uma associação de aplicação sem perda de informação em trânsito.
- A-ABORT – Causa o término anormal de uma associação de aplicação com a possibilidade de perda de informação em trânsito.
- A-P-ABORT – Indica o término anormal de uma associação de aplicação pelo ACSE com a possibilidade de perda de informação em trânsito.

Os serviços do ACSE operam dois modos: modo "normal" ou modo "X.410-1984".

O modo de operação normal permite que o usuário do ACSE utilize toda a funcionalidade oferecida pelo ACSE e pelo serviço de apresentação. Neste modo o ACSE transfere sua semântica utilizando o serviço de apresentação em modo normal.

O modo de operação X.410-1984 permite que o usuário do ACSE se comunique com o seu par utilizando o protocolo especificado na Recomendação X.410-1984. Neste modo, o ACSE não transfere qualquer semântica própria e usa o serviço de apresentação no modo X.410-1984.

III.4.5.2 O Elemento de Serviço de Operações Remotas (ROSE)

O objetivo do Elemento de Serviço de Operações Remotas (ROSE) é permitir a execução de operações em sistemas remotos. O sistema local envia chamadas ao sistema remoto desejado, contendo as operações que ele deseja executar. Os resultados ou erros provenientes da execução da operação, ou falhas detectadas na utilização do serviço podem ser retornados.

O serviço do ROSE é definido por cinco primitivas de serviço:

- RO-INVOKE – Solicita a execução de uma operação.
- RO-RESULT – Retorna o resultado de uma operação bem sucedida.
- RO-ERROR – Retorna erros da execução de uma operação.
- RO-REJECT-U – Retorna falha em uma das primitivas acima detectada no sistema remoto.
- RO-REJECT-P – Retorna falha em uma primitiva detectada pelo ROSE.

As operações podem ser categorizadas em cinco classes. Operações da classe 1 são do tipo síncronas, ou seja, o iniciador de uma operação necessita receber uma resposta antes de solicitar outra operação. Operações da classe 2 a 5 são assíncronas, ou seja, o usuário pode solicitar operações sem que seja necessário aguardar respostas. As operações de classe 1 e 2 retornam resultados e erros, as de classe 3 retornam somente falhas, as de classe 4 retornam somente resultados, e as de classe 5 não retornam nem resultados nem condições de erro.

A execução de operações é geralmente independente, mas o ROSE oferece a possibilidade de "encadear operações". Isto é realizado através das Operações Encadeadas (Linked Operations), que consistem de uma operação pai e uma ou mais operações filho. A entidade respondedora da operação pai invoca uma ou mais operações filho na entidade originariamente iniciadora. Uma operação filho pode ser operação pai de um outro conjunto de operações encadeadas.

Deve sempre existir uma associação de aplicação quando o ROSE é iniciado, e o ROSE continua sua execução até que a associação seja externamente terminada.

Existem três classes de associações para a operação do ROSE. Esta classificação determina que usuário pode invocar operações: na classe 1 somente o iniciador da associação pode invocar operações; na classe 2 somente o respondedor da associação pode invocar operações; e na classe 3 ambos podem invocar operações. O operações encadeadas requerem que a associação entre os usuários seja de classe 3. O controle da associação de aplicação é realizado pelo ACSE ou pelo RTSE.

O processo de aplicação realiza o mapeamento das operações disponíveis ao elemento de usuário em serviços do ACSE, ou do RTSE; e do ROSE, dependendo do contexto de aplicação.

Se o contexto de aplicação inclui o RTSE, a função do processo de aplicação que realiza o mapeamento é usuária do RTSE e do ROSE, o ROSE é usuário do RTSE, e o RTSE é usuário do ACSE e do serviço de apresentação.

Se o contexto de aplicação não inclui o RTSE, a função do processo de aplicação que realiza o mapeamento é usuária do ROSE e do ACSE, e o ROSE e o ACSE são usuários do serviço de apresentação.

III.4.5.3 O Elemento de Serviço de Transferência Confiável (RTSE)

O objetivo do Elemento de Serviço de Transferência Confiável (RTSE) é assegurar que cada Unidade de Dados de Protocolo de Aplicação (APDU) é completamente transferida entre AEs exatamente uma vez, ou a AE transmissora será avisada em caso de uma exceção. A Transferência Confiável recupera falhas na comu-

nicação e no sistema-fim e minimiza a quantidade de retransmissões necessárias a recuperação.

A transferência confiável é realizada no contexto de uma associação de aplicação. O estabelecimento e controle desta associação é realizada pelo RTSE através do serviço do ACSE.

São definidos cinco serviços para o RTSE:

- RT-OPEN – Estabelece uma associação de aplicação com outra AE.
- RT-CLOSE – Termina uma associação de aplicação.
- RT-TRANSFER – Transfere uma APDU.
- RT-TURN-PLEASE – Solicita a vez de transmissão a AE par.
- RT-TURN-GIVE – Cede a vez de transmissão a AE par.
- RT-P-ABORT – Indica ao usuário que a associação de aplicação não mais pode ser mantida.
- RT-U-ABORT – Solicita término anormal da associação de aplicação.

Os serviços do RTSE operam dois modos: modo "normal" ou modo "X.410-1984".

O modo de operação normal permite que o usuário do ACSE utilize toda a funcionalidade oferecida pelo RTSE. Neste modo o RTSE deve utilizar os serviços do ACSE e de apresentação em modo normal.

O modo de operação X.410-1984 permite que o usuário do ACSE se comunique com o seu par utilizando o protocolo especificado na Recomendação X.410-1984. Este modo, implica certas restrições no uso dos serviços do RTSE, e implica o uso dos serviços do ACSE e de apresentação no modo X.410-1984.

III.4.5.4 Modelagem Estelle do CASE

Na modelagem Estelle, o CASE foi especificado de forma que uma só instância da especificação possa atender os vários processos de aplicação de um mesmo sistema aberto.

A modelagem do CASE consiste do sub-módulo CASE_PROVIDER_TYPE, que possui atributo activity e comportamento dado pelo corpo CASE_BODY. Este sub-módulo possui quatro vetores (arrays) de pontos de interação externos. Um vetor para cada Elemento de Serviço Comum e outro para a comunicação com a camada

de apresentação. Estes vetores são limitados por constantes que definem o número máximo de usuários que cada Elemento de Serviço Comum pode suportar. O vetor referente a comunicação com a camada de apresentação é limitado por uma constante que define o número máximo de conexões que a camada de apresentação pode suportar.

```
module CASE_PROVIDER_TYPE activity;
  ip RT:  array [1..max_rtusr] of RTSE_interface (provider);
  RO:    array [1..max_rousr] of ROSE_interface (provider);
  AC:    array [1..max_acusr] of RTSE_interface (provider);
  PCEP:  array [1..max_cnx] of APRESENTACAO_interface (provider);
end;

body CASE_BODY for CASE_PROVIDER_TYPE;
```

O módulo CASE_PROVIDER_TYPE é declarado no módulo principal da modelagem, MHS, onde é também criado e instanciado com o nome CCASE[i].

O comportamento do sub-módulo CCASE é orientado pelos eventos detectados em seus pontos de interação externos, ou seja, eventos provocados pelos usuários ou pelo serviço de apresentação.

O sub-módulo CCASE possui três módulos filhos que modelam cada Elemento de Serviço Comum.

ACSE_PROVIDER_TYPE : Possui atributo activity e modela a máquina de transição de estados do protocolo do ACSE, que estabelece e controla uma associação de aplicação. Seu comportamento é dado pelo corpo ACSE_BODY, e sua definição é mostrada abaixo.

```
module ACSE_PROVIDER_TYPE activity (tipo_usr: cod_usr);
  ip A: ACSE_interface (provider);
  I: ACRS_interface (provider);
  P: APRESENTACAO_interface(user);
  export Release_me: boolean;
end;
```

```
body ACSE_BODY for ACSE_PROVIDER_TYPE;
```

Este módulo filho possui um parâmetro de entrada, "cod_usr", do tipo "tipo_usr", que indica qual o tipo de contexto de aplicação do processo usuário. Isto é necessário para que os Elementos de Serviço Comuns interajam de maneira coordenada. Por exemplo, se o ROSE pertence ao contexto de aplicação, ele precisa ser sinalizado de que a associação já existe para que possa prover serviços ao usuário.

Este módulo filho exporta uma variável chamada "release_me". A variável release_me é do tipo booleano e exporta uma indicação de que o sub-módulo não é mais necessário e pode ser descartado (declaração release), ou seja, a associação foi terminada.

Este módulo filho possui três pontos de interação externos chamados A (do tipo ACSE_interface), I (do tipo ACRO_interface), e P (do tipo APRESENTACAO_interface). Através destes pontos de interação este módulo dispara eventos para seu usuário, para o ROSE (somente se este faz parte do contexto de aplicação), e para o serviço de apresentação, respectivamente.

ROSE_PROVIDER_TYPE : Possui atributo activity e modela o protocolo de operações remotas do ROSE. Seu comportamento é dado pelo corpo ROSE_BODY, e sua definição é mostrada abaixo.

```
module ROSE_PROVIDER_TYPE activity;
  ip R: ROSE_interface (provider);
  I: ACRS_interface (user);
  P: APRESENTACAO_interface (user);
  export Release_me: boolean;
end;

body ROSE_PROVIDER_BODY for ROSE_PROVIDER_TYPE;
```

Este módulo filho exporta uma variável chamada "release_me". A variável release_me é do tipo booleano e exporta uma indicação de que o sub-módulo não é mais necessário e pode ser descartado (declaração release), ou seja, a

associação foi terminada e o usuário não mais deseja a execução de operações remotas.

Este módulo filho possui três pontos de interação externos chamados R (do tipo ROSE_interface), I (do tipo ACRO_interface), e P (do tipo APRESENTACAO_interface). Através destes pontos de interação este módulo dispara eventos para seu usuário, recebe eventos do ACSE, e dispara eventos para o serviço de apresentação, respectivamente.

RTSE_PROVIDER_TYPE : Possui atributo activity e modela a máquina de transição de estados do protocolo do RTSE, que executa transferência confiável de dados. Seu comportamento é dado pelo corpo RTSE_BODY.

```
module RTSE_PROVIDER_TYPE    activity;
    ip  R: RTSE_interface    (provider);
        A: ACSE_interface    (user);
        P: APRESENTACAO_interface (user);
    export Release_me: boolean;
end;

body    rtse_body for rtse_provider_type; external;
```

Este módulo filho exporta uma variável chamada "release_me". A variável release_me é do tipo booleano e exporta uma indicação de que o sub-módulo não é mais necessário e pode ser descartado (declaração release), ou seja, a associação foi terminada e o usuário não mais deseja a execução de operações remotas.

Este módulo filho possui três pontos de interação externos chamados R (do tipo RTSE_interface), A (do tipo ACSE_interface), e P (do tipo APRESENTACAO_interface). Através destes pontos de interação este módulo dispara eventos para seu usuário, utiliza os serviços do ACSE, e do serviço de apresentação, respectivamente.

O sub-módulo CCASE possui seis vetores de pontos de interação internos que permitem o diálogo entre as instâncias de módulos filhos. Estes pontos de interação são conectados (declaração connect) dinamicamente aos pontos de interação externos das instâncias de sub-módulos filhos durante a execução da especificação,

e assim permitem que o módulo pai (CCASE) tenha acesso aos eventos por eles disparados.

ip

```
IDLE_RT: array [1..maxrtusr] of RTSE_interface (user);
IDLE_A:  array [1..maxausr]  of ACSE_interface (user);
IDLE_P:  array [1..maxcnx]   of APRESENTACAO_interface (provider);

A_RT: array [1..maxrtusr] of APRESENTACAO_interface (provider);
A_RS: array [1..maxrsusr] of APRESENTACAO_interface (provider);
A_A:  array [1..maxausr]  of APRESENTACAO_interface (provider);
```

As várias instâncias dos módulos filhos são declaradas pelo comando "modvar" como mostrado abaixo.

modvar

```
RTSE: array [1..max_rtusr] of RTSE_PROVIDER_TYPE;
ROSE: array [1..max_rsusr] of ROSE_PROVIDER_TYPE;
ACSE: array [1..max_ausr] of ACSE_PROVIDER_TYPE;
end;
```

Durante a inicialização do CCASE nenhum comando é executado. As várias instâncias dos módulos filhos ACSE_PROVIDER_TYPE, ROSE_PROVIDER_TYPE e RTSE_PROVIDER_TYPE, isto é, ACSE[i], ROSE[i] e RTSE[i], respectivamente, são criadas dinamicamente de acordo com os eventos detectados nos pontos de interação externos do modulo CASE.

Quando o sub-módulo CCASE recebe do usuário i um evento "A_AssReq", através de um ponto de interação AC[i], os módulos ACSE[i] e ROSE[i] são criados, o ponto de interação A do módulo ACSE é conectado ao ponto de interação interno IDLE_AC[i], através do qual este módulo envia a interação recebida ao ACSE[i]. Logo após este envio, este ponto de interação interno é desconectado do ponto de interação A do ACSE, que é ligado ao ponto de interação externo A[i] para que o ACSE possa se comunicar diretamente com o usuário. Da mesma forma, o ponto

de interação interno R do ROSE é ligado ao ponto de interação externo R[i] do CCASE. Os pontos de interação externos I do ACSE e ROSE são conectados.

É assumido que um usuário que utiliza diretamente os serviços do ACSE para estabelecimento de associação, utilizará o ROSE para transferência de dados já que um usuário do RTSE o utiliza para estabelecimento de associação, e este, por sua vez utiliza o ACSE.

Para comunicação com o serviço de apresentação os pontos de interação externos P do ACSE e ROSE são conectados aos pontos de interação internos do CCASE A_AC[i] e A_RO[i], respectivamente. Assim, o módulo pai (CCASE) receberá os eventos que os módulos filhos desejam enviar ao serviço de apresentação e os enviará pelo ponto de interação externo PCEP[i], que é alocado para esta conexão de apresentação. A criação dos módulos ACSE[i] e ROSE[i] é mostrada abaixo.

```
any i: 1..max_rsusr do      (* Rose Users *)
  when A[i].A_AssReq
    begin
      init      ROSE[i]      with ROSE_PROVIDER_BODY;
      init      ACSE[i]      with ACSE_PROVIDER_BODY (rose_usr);
      connect   ACSE[i].A    to   IDLE_A[i];
      output    IDLE_A[i].A_AssReq (Udata);
      disconnect ACSE[i].A;
      attach    RS[i]        to   ROSE[i].R;
      connect   ROSE[i].I    to   ACSE[i].I;
      attach    A[i]         to   ACSE[i].A;
      connect   A_RS[i]      to   ROSE[i].P;
      connect   A_A[i]       to   ACSE[i].P;
    end;
```

Quando o sub-módulo CCASE recebe do usuário i um evento "RT-OpReq", através de um ponto de interação RT[i], os módulos RTSE[i] e ACSE[i] são criados, o ponto de interação R do módulo ACSE é conectado ao ponto de interação interno do CCASE IDLE_RT[i], através do qual este módulo envia a interação recebida ao RTSE[i]. Logo após este envio, este ponto de interação interno é desconectado do ponto de interação RT do RTSE, que é ligado ao ponto de interação externo do CCASE R[i] para que o RTSE possa se comunicar diretamente com o usuário. Os pontos de interação externos A do RTSE e ACSE são conectados.

Para comunicação com o serviço de apresentação os pontos de interação externos P do RTSE e ACSE são conectados aos pontos de interação internos do CCASE A_RT[i] e A_AC[i], respectivamente. Assim, o módulo pai (CCASE) receberá os eventos que os módulos filhos desejam enviar ao serviço de apresentação e os enviará pelo ponto de interação externo PCEP[i], que é alocado para esta conexão de apresentação. A criação dos módulos ACSE[i] e RTSE[i] causada pela recepção de um evento "RT_OpReq" é mostrada abaixo.

```
any i: 1..max_rtusr do      (* Rtse Users *)
  when RT[i].RT_OpReq
    begin
      init      RTSE[i]      with RTSE_PROVIDER_BODY;
      init      ACSE[i]      with ACSE_PROVIDER_BODY (rtse_usr);
      connect   RTSE[i].R to  IDLE_RT[i];
      output    IDLE_RT[i].RT_OpReq (Udata);
      disconnect RTSE[i].R;
      attach    RT[i]        to  RTSE[i].R;
      connect   RTSE[i].A to  ACSE[i].A;
      connect   A_RT[i]      to  RTSE[i].P;
      connect   A_A[i]       to  ACSE[i].P;
    end;
```

Quando o sub-módulo CCASE recebe do usuário i um evento "P_ConInd", através de um ponto de interação PCEP[i], uma função chamada "IsRtseUsr" é executada para determinar se o processo de aplicação ao qual o pedido de conexão se refere é usuário do RTSE. A partir desta determinação, um procedimento análogo a um dos dois acima descritos é executado, sendo que neste caso, o ponto de interação interno P do ACSE é conectado ao ponto de interação externo IDLE_P[i] do CCASE; através deste é enviada a primitiva recebida do serviço de apresentação e o ponto de interação é desconectado. A criação dos módulos ACSE[i] e ROSE[i] causada pela recepção de um evento "P_ConInd" é mostrada abaixo.

```
trans
  any i: 1..max_usr do
    when PCEP[i].P_ConInd
      begin
```

```
if IsRtseUsr then
  begin
    init    RTSE[i]    with  RTSE_PROVIDER_BODY;
    init    ACSE[i]    with  ACSE_PROVIDER_BODY (rtse_usr);
  end
else (* IsRoseUsr *)
  begin
    init    ROSE[i]    with  ROSE_PROVIDER_BODY;
    init    ACSE[i]    with  ACSE_PROVIDER_BODY (rose_usr);
  end;
connect    ACSE[i].P    to    IDLE_P[i];
output     IDLE_P[i].P_ConInd (APDU);
disconnect ACSE[i].P;
if IsRtseUsr then
  begin
    attach  RT[i]      to  RTSE[i].R;
    connect RTSE[i].A  to  ACSE[i].A;
    connect A_RT[i]    to  RTSE[i].P;
  end
else
  begin
    attach  RS[i]      to  ROSE[i].R;
    attach  A[i]       to  ACSE[i].A;
    connect ROSE[i].I  to  ACSE[i].I;
    connect A_RS[i]    to  ROSE[i].P;
  end;
connect    A_A[i]     to  ACSE[i].P;
end;
```

Os módulos filhos assim criados são destruídos (release) quando exportam a variável "release_me" é exportada com valor "TRUE". A transição abaixo é realizada para cada módulo filho. Um exemplo de destruição dos módulos filhos é mostrado abaixo.

```
trans
  provided exist  ACSE: ACSE_PROVIDER_TYPE
```

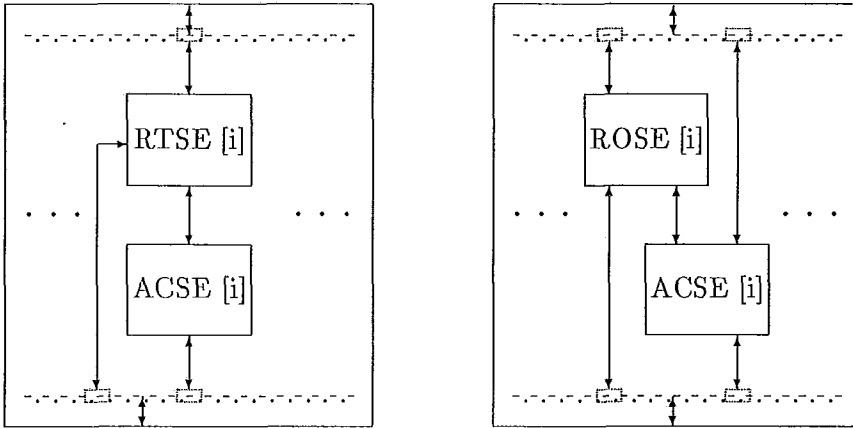


Figura III.15: Arquitetura da Modelagem Estelle do CASE

```
suchthat (ACSE.Release_me)
begin
  forone ACSE: ACSE_PROVIDER_TYPE
  suchthat (ACSE.Release_me) do
    begin
      release ACSE;
    end;
  end;
end;
```

As demais transições do sub-módulo CCASE fazem a transferência de interações recebidas dos pontos de interação PCEP[i] para os módulos filhos e vice-versa [55].

A Figura III.15 mostra as possíveis configurações da modelagem descrita acima.

III.5 O Sistema de Auxílio para Projetos em Estelle

Um dos principais objetivos do uso de técnicas formais para a modelagem de sistemas é a possibilidade de utilização de ferramentas para auxílio no desenvolvimento do projeto, aumentando a qualidade do sistema resultante.

Para a modelagem descrita neste trabalho foi utilizado Sistema de Auxílio ao Projeto de Protocolos de Comunicação em desenvolvimento no Programa de Engenharia Elétrica da COPPE/UFRJ [48]. Este Sistema tem a linguagem Estelle como formalismo para a descrição do software a ser tratado, e tem por objetivo

facilitar o desenvolvimento de protocolos de comunicação para redes de computadores.

A arquitetura deste Sistema inclui um compilador, um simulador, um gerador de seqüências de testes, um módulo implementador, um sistema de edição e uma interface homem-máquina compondo um ambiente integrado.

O compilador faz a tradução de um fonte, constituído por uma especificação Estelle, gerando uma Forma Intermediária, onde todos os conceitos da linguagem se encontram definidos de forma estruturada adequada, e que serve como entrada padronizada dos demais módulos de tratamento de uma especificação tais como o simulador de protocolos, o gerador de seqüências de teste, e o implementador.

O simulador executa o modelo de Estelle permitindo amplo acesso a todos os objetos que representam a dinâmica do protocolo simulado. Ele exercita o conjunto de módulos cooperantes que constituem o sistema especificado em Estelle. Ao longo de uma sessão de simulação são criadas instâncias de módulos que podem ser eventualmente destruídas e recriadas novamente. A criação de uma instância de módulo corresponde à criação de um conjunto de instâncias de objetos associados ao módulo dado, tais como as filas correspondentes aos pontos de interação, as variáveis e os módulos declarados em seu interior.

O gerador de seqüência de testes gera as seqüências de testes a serem usadas durante as fases de simulação e de teste final da implementação de um dado protocolo. O modelo de entrada aceito pelo gerador de seqüência de testes é constituído pelo conjunto de máquinas de estados finitas obtidas de um texto em Estelle.

O módulo implementador é encarregado de gerar código executável em uma dada máquina alvo, partindo das informações fornecidas pela forma intermediária. Devem ser portanto considerados o sistema operacional e uma linguagem de programação disponíveis.

Durante a elaboração deste trabalho foi utilizado o módulo compilador deste Sistema de Auxílio. A modelagem descrita neste trabalho resultou em 7200 linhas compiladas. Não foi objetivo do trabalho utilizar os demais módulos do Sistema (simulador, gerador de testes e implementador). Entretanto, este trabalho tem sido continuado através de Projetos de Iniciação Científica, que tem o objetivo de simular as diferentes partes desta especificação Estelle.

A modelagem não pode ser simulada por inteiro neste Sistema por razões de limitação do simulador quanto ao tamanho da especificação. Porém, seria interessante tomar para simulação, estruturas menores da modelagem, tais como:

(i) O módulo CASE: Com isto se pode simular de forma independente os

protocolos dos Elementos de Serviço Comuns da Camada de Aplicação.

- (ii) Os módulos `UA_TYPE`, `MS_ACCESS_TYPE` e `MS_TYPE`: Com isto se pode simular o protocolo P7 e o comportamento UA em relação a este protocolo.
- (iii) Os módulos `UA_TYPE`, `MTS_ACCESS_TYPE` e `P3_PROVIDER_TYPE`: Com isto se pode simular o protocolo P3 e o comportamento UA em relação a este protocolo.
- (iv) Os módulos `UA_TYPE`, `MS_ACCESS_TYPE`, `MS_TYPE`, `MTS_ACCESS_TYPE` e `P3_PROVIDER_TYPE`: Com isto se pode simular o comportamento do MS executando os protocolos P7 e P3 simultaneamente, ou seja, a interação UA-MS-MTA.

Deve-se notar que a simulação das partes (ii) a (iv) incluem o uso do módulo CASE já que cada entidade estaria em sistemas abertos distintos.

- (v) O módulo `MTA_TYPE`: Com isto se pode simular o protocolo P7 e a operação distribuída do MTS.

III.6 Comentários e Propostas a Estelle

A escolha, definição e suporte de uma metodologia coerente para especificação, validação e implementação de protocolos reais e sistemas distribuídos de grande porte apresenta algumas dificuldades.

A primeira dificuldade está relacionada ao fato de que a metodologia deve, simultaneamente e de forma coerente, permitir uma especificação de alto nível, independente de detalhes e escolhas de implementação, e, dentro do mesmo ambiente, traduzir a especificação para uma implementação. Deve-se observar que a implementação, no contexto de sistemas abertos, deve ser definida para diferentes tipos de ambientes e sistemas operacionais, o que implica a manipulação e uso de mecanismos de linguagens e de sistemas operacionais que suportarão as possíveis implementações reais.

Além disso, a metodologia deve permitir eficiência no projeto e implementação, e facilidade de validação e de geração de seqüências de testes.

A modelagem descrita na seção anterior mostra o esforço de se descrever formalmente as entidades e protocolos de um sistema distribuído.

A importância desta modelagem está na definição precisa do sistema em suas entidades e protocolos, e da utilização dos serviços que servem de suporte ao MHS.

O modelo abstrato com o qual o MHS é originariamente descrito em suas Recomendações, não permite a visualização das fronteiras entre os objetos, não especifica o comportamento das entidades, nem mostra o mapeamento do modelo nos serviços de suporte. Estes aspectos são descritos em linguagem natural. São também descritos em linguagem natural, outros aspectos importantes de protocolos tais como, a especificação de que as operações são síncronas ou não; a prioridade com que os pedidos de operações, resultados ou erros são enviados entre as entidades.

Já a especificação Estelle força a especificação das interfaces entre módulos, através da especificação de canais e pontos de interação; além de forçar a especificação precisa dos predicados que devem ser satisfeitos de forma a ativar transições.

Quanto ao comportamento das entidades, isto é facilitado pelo fato de que Estelle é uma linguagem algorítmica, portanto; as sentenças que descrevem os procedimentos tiveram de ser traduzidas para as declarações Estelle e da linguagem de programação Pascal.

A representação do sistema utilizando os conceitos Estelle de tipos de módulos, corpos de módulos e instâncias apresenta as seguintes interessantes características:

- Permite uma separação explícita entre a configuração do sistema e a programação dos processos (descrição do comportamento de cada componente);
- A especificação torna-se independente do número de instâncias de tipo e corpo de módulo ("module type" e "module body", respectivamente).

Dois tipos de dificuldade se apresentaram no desenvolvimento da modelagem do MHS em Estelle. A primeira envolve a questão da representação de dados. Em Estelle isto é realizado através dos tipos Pascal, ou seja, Estelle não provê ferramentas suficientes para a especificação de dados de protocolos de comunicação, o que é de grande importância para os protocolos da camada de aplicação.

As recomendações X.400 utilizam extensivamente a notação ASN.1 para a descrição dos dados dos protocolos além de usá-las para o desenvolvimento de seu modelo abstrato como visto em III.3.2.3. Uma solução seria traduzir a declaração de dados de ASN.1 para Pascal na especificação Estelle. Porém, este modo de proceder não seria eficiente pois não existe uma correspondência direta entre os tipos básicos ASN.1 e os tipos Pascal, e isto poderia dar origem a erros.

Além disso, ASN.1 já é uma linguagem de declaração de dados e, traz consigo um esquema de codificação padrão [35,36] que tem sido adotado pelos protocolos

do nível de aplicação OSI. Baseando-se na informação contida na definição ASN.1 da estrutura de uma PDU, este esquema determina completamente a codificação e decodificação de maneira sistemática.

Partindo-se destes princípios, seria interessante que a linguagem Estelle suportasse a declaração e manipulação de dados em ASN.1, para que as especificações se tornassem ainda mais eficientes e completas. Isto poderia ser feito através da definição de uma "extensão ASN.1 para Estelle". Com esta extensão, Estelle incorporaria os tipos ASN.1 e poderia ter a forma abaixo.

```
{ $ASN1
    . . . .
    (* Declaracao de variaveis ASN.1 *)
    . . . .
}
```

Neste espaço seriam inseridos os dados de protocolo tal como se apresentam nas recomendações. E poderia ser permitido o acesso a estas variáveis da mesma forma que são acessado e manipulados variáveis do tipo "record" Pascal. Isto facilitaria a especificação de protocolos e serviços da camada de aplicação e permitiria a aplicação de ferramentas para a geração automática de programas de codificação/decodificação de PDUs como em [45].

Outro ponto se deve ao fato de que as restrições ou extensões da linguagem Pascal na definição Estelle não são sempre muito coerentes. Por exemplo, é possível utilizar ponteiros para manipulação de dados, mas não é permitido utiliza-los como parâmetros de interações de módulos e como variáveis exportadas. A linguagem Estelle também não dispõe das funções de entrada e saída (por exemplo, write, read).

A segunda dificuldade está relacionada ao mapeamento do modelo abstrato utilizado pelas Recomendações (III.3.2.3), que segue uma visão objeto-orientada do sistema, na modelagem Estelle.

Uma definição objeto-orientada de uma linguagem representa as construções e conceitos da linguagem como classes de objetos com comportamento e estrutura comuns. Um programa ou especificação então, é definido por um conjunto de

objetos destas classes. Seu comportamento é definido pelos membros da classe e define, por sua vez, a semântica do programa ou especificação.

Dado forte impacto desta forma de modelagem em OSI [49], seria interessante que fosse desenvolvida uma definição objeto-orientada da semântica de Estelle que permitisse a visão de uma instância de módulo como um objeto que pode ser ativado enviando-se a ele mensagens. Este objeto pode por sua vez enviar mensagens a outros objetos.

A fim de obter uma completa descrição da semântica de Estelle, esta abordagem deve ser completada com a identificação e descrição comportamental de todos os tipos de objetos definidos por Estelle. A estrutura de um objeto e seus relacionamentos com outros objetos seriam derivados da definição e semântica Estelle do conceito correspondente. O envio de mensagens para as classes permitiriam a criação de definições e declarações específicas. As mensagens respondidas pelas instâncias criadas definiriam os comportamentos dos conceitos Estelle correspondentes [46].

Capítulo IV

Proposta de Migração de um MHS-84 para um MHS-88

IV.1 Introdução

Este capítulo tem por objetivo descrever uma proposta de migração de um MHS de acordo com as Recomendações X.400 1984 para um MHS em conformidade com as Recomendações de 1988 baseado na modelagem Estelle descrita no Capítulo III.

Na seção IV.2 deste capítulo analisam-se as razões da migração levantando-se os aspectos insuficientemente definidos ou não resolvidos nas Recomendações X.400 1984. Na seção IV.3 é a proposto um processo de migração de um MHS baseado nos padrões X.400 1984 para um MHS baseado nos padrões de 1988. A seção IV.4 descreve a implementação da primeira fase da proposta de migração apresentada na Seção IV.3. A seção IV.5 aborda aspectos relacionados ao desenvolvimento da segunda fase do processo de migração apresentado. A seção IV.6 tece comentários sobre todo o capítulo.

IV.2 A Evolução dos Padrões X.400: Por que Migrar?

Existem alguns aspectos do MHS que o padrão X.400 1984 deixa explícita ou implicitamente como questões de decisão local, para que o implementador resolva. A partir do momento em que cada equipe de implementação escolhe sua própria solução, pode-se gerar problemas de interoperabilidade entre os sistemas.

Estes aspectos incluem a disponibilidade de usuário remoto para a entrega de mensagens, o redirecionamento de mensagens, as ações relacionadas à entrega

de mensagens, a supressão e detecção de loops, os serviços de diretório, listas de distribuição, a estrutura do dispositivo de armazenamento de mensagens, o roteamento de mensagens e a interface com o usuário. Estes aspectos serão discutidos nos itens que se seguem.

A versão 1988 das Recomendações X.400 é uma versão estendida e compatível do padrão predecessor. Esta nova versão esclarece alguns dos aspectos omitidos ou confusos descritos anteriormente, e soma, de forma significativa, funcionalidade a nível de protocolos e serviços. Outros aspectos, entretanto, continuam a cargo do implementador.

Esta seção descreve e discute estes aspectos conflitantes do padrão sob o ponto de vista de implementação, mostrando como eles se apresentam tanto na série X.400 1984 (MHS-84), como na série X.400 1988 (MHS-88), e apresenta as extensões do novo padrão para a migração de um sistema baseado em X400-84 para um sistema baseado em X400-88.

IV.2.1 Entrega de Mensagens e a Configuração Física do UA

Quanto à configuração física, um UA pode ser co-residente ou remoto ao MTA que o atende para transmissão e recepção de mensagens. As Recomendações X.400 especificam que tanto o UA como o MTA podem inicializar comunicação para entrega de mensagens. O problema está na disponibilidade do UA em receber mensagens do MTA e esta disponibilidade está relacionada com a localização física do UA e do MTA.

Se o UA é remoto ao MTA, ele é geralmente implementado em PCs, e portanto, pode não estar disponível quando o MTA deseja se comunicar com ele para entrega de mensagens, usando o protocolo P3. Com isso, mensagens podem expirar e/ou podem ser consideradas sem possibilidade de entrega se o usuário não está apto a acessar seu MTA com suficiente frequência.

Quando o UA e o MTA são co-residentes, o processo UA pode não estar ativo sempre que o processo MTA está. A nível de implementação, se o sistema operacional permitir que a área de mensagens do UA seja acessível ao MTA, este poderia realizar entrega sem a participação ativa do UA.

Os padrões X.400 1988 oferecem uma solução para o problema. Apesar de conter o protocolo P3, pois os padrões devem ser compatíveis, uma nova entidade funcional, o Message Store, e um novo protocolo, P7, superam o problema de disponibilidade do UA. Um MS pode co-residir com o MTA, para receber as men-

sagens no lugar do UA que ele representa. O UA remoto pode então estabelecer comunicação com o MS conforme desejado, utilizando o protocolo P7.

O MS co-residente com o MTA, ocupa-se não somente do serviço de entrega, mas também do serviço de submissão e de administração do MTA, além de permitir maior controle ao UA quanto à entrega de mensagens do que com o uso do protocolo P3. De acordo com o protocolo P7, o UA recupera mensagens do MS sob o controle do usuário, enquanto com o protocolo P3 o UA é forçado a aceitar a entrega de essencialmente todas as mensagens no momento em que abre conexão com o MTA.

IV.2.2 Redirecionamento de Mensagens

O MHS-84 permite que um originador rotule uma mensagem com um qualificador de "permissão para destinatário alternativo", indicando que é permitida a entrega da mensagem a outro destinatário caso ela seja considerada sem possibilidade de entrega para os destinatários originais. Os padrões X.400 1984 não definem qualquer mecanismo para a designação deste destinatário alternativo.

A nível de implementação este problema pode ser resolvido permitindo-se que o usuário especifique um destinatário alternativo no momento em que ele se registra no sistema. Este destinatário alternativo pode ser alterado posteriormente pelo usuário caso deseje.

O MHS-88 permite que o originador especifique, para cada destinatário, um destinatário alternativo para o qual a mensagem será redirecionada caso a entrega para o destinatário original falhe. Além disso, um usuário, do ponto de vista do destinatário, pode especificar (no seu registro com o MTS) que mensagens para ele sejam redirecionadas para outro usuário. Existe ainda um serviço, através do qual o originador proíbe a reatribuição pelo destinatário.

IV.2.3 Ações automáticas relacionadas à recepção de mensagens ("auto-actions")

O Protocolo P2 do MHS permite que o originador de uma mensagem rotule uma mensagem com qualificadores tais como importância, segurança, data de expiração, mensagens obsoletas, indicação de referência cruzada, e de que a mensagem foi automaticamente redestinada a outro usuário. Estes qualificadores transportam informação sobre a mensagem ao usuário destinatário. Quando alguma ação é tomada automaticamente pelo UA na recepção destas informações (tais como

desabilita a listagem de uma mensagem confidencial em impressora pública ou descartar automaticamente mensagens expiradas), ou quando ação automática é tomada para gerar mensagens (tal como redestinar automaticamente uma mensagem sob algum critério), estes qualificadores podem ser de grande valor.

Os padrões de 84 não especificam quaisquer ações automáticas para gerar ou responder aos qualificadores de uma mensagem. Tais ações são ditas "questões locais". Os novos padrões, entretanto, esclarecem e dão suporte à implementação de tais auto-ações. Em particular, permite o registro de pedidos para descartar automaticamente mensagens expiradas e para auto-redestinar mensagens quando, por exemplo, um usuário precisa redestinar certas mensagens para outro usuário durante sua ausência.

Isto é operacionalizado através das "variáveis de estado" definidas para o UA e descritas na seção II.4.1.

Outras ações, tais como resposta automática a indicação de importância, segurança, continuam como questões locais de implementação.

IV.2.4 Prevenção e Detecção de "Loops"

Uma importante função da entidade MTA é a prevenção e detecção de "loops" no roteamento de uma mensagem.

O MHS-84 descreve um algoritmo para a prevenção e detecção de "loop", baseado na análise de informações contidas no envelope da mensagem, chamadas informação de "trace". Este elemento de protocolo contém informação sobre a passagem da mensagem através dos sistemas. Porém, esta descrição é confusa, podendo levar a interpretações ambíguas. O MHS-84 também não provê técnicas de supressão e detecção de loops dentro de um domínio de gerenciamento.

O MHS-88 traz a descrição de um algoritmo muito mais claro e completo, apesar de seu uso não ser obrigatório. Além disso, os novos padrões definem uma informação de "trace" interna, para ser utilizada na supressão e detecção de loops dentro de um domínio do MHS.

IV.2.5 Uso dos Serviços de Diretório

Em todo MHS, os usuários são identificados pelos chamados Nomes O/R ("Originator/Recipient Names"). Os nomes O/R são compostos de uma lista de atributos escolhidos de uma lista de atributos padrão, podendo conter, opcionalmente, atributos definidos internamente ao domínio de gerenciamento. Nomes O/R que

contêm atributos que indicam a localização do usuário são "endereços O/R".

No MHS-84, todo nome O/R tem características de endereço, ou seja, é um endereço O/R. Endereços O/R são de difícil memorização por geralmente conter uma longa lista de atributos. Então, se o implementador não fornece mecanismos que possam facilitar a identificação do usuário, este tem de aprender e lembrar dos endereços O/R de seus destinatários.

Uma solução seria implementar um diretório pessoal de modo a facilitar o endereçamento dos destinatários de suas mensagens. Cada usuário escolhe nomes amigáveis para seus correspondentes e estes nomes são armazenados com os correspondentes nomes O/R. No momento da submissão da mensagem, o usuário especifica os nomes associados aos endereços dos destinatários e o UA se encarrega de mapeá-los nos endereços O/R que eles representam.

No MHS-88 o nome O/R passa a ser chamado endereço O/R. O termo nome O/R é usado para conter um "nome de Diretório" e/ou um "endereço O/R". Este nome de Diretório permite o armazenamento e recuperação de endereços O/R pelo uso de um Sistema de Diretório padronizado e definido pelo CCITT (Recomendações X500). O nome de Diretório é utilizado para obter endereços O/R no Sistema de Diretório.

IV.2.6 Listas de Distribuição

É geralmente útil que um usuário de MHS possa especificar convenientemente um grupo de destinatários para uma mensagem, de forma que o usuário originador especifique somente um nome para o grupo (group name), e este nome seja mapeado em uma lista de nomes O/R. Esta lista de nomes O/R associada a um nome é chamada "Lista de Distribuição" (vide II.3.7).

As normas de 1984 não definem suporte para listas de distribuição. Geralmente, implementações de MHS-84 oferecem sistemas de diretório locais, não padronizados, para associar "group names" á listas de distribuição. Neste caso, o mapeamento é feito no UA originador.

O MHS-88 padroniza um método para manipulação de listas de distribuição (DLs). Cada DL tem um endereço O/R que habilita o MTS a rotear a mensagem ao seu "ponto de expansão". Conforme visto em II.3.7, um ponto de expansão é simplesmente um MTA capaz de mapear este endereço O/R na lista de membros correspondente e então distribuir a mensagem a cada endereço da lista. Este processo é chamado "expansão de uma DL". No MHS-88, a expansão de listas é realizada no Sistema de Transferência de Mensagens e pode ser suportada pelo uso

do sistema padronizado de Diretório ou pode usar métodos locais no seu ponto de expansão.

IV.2.7 Armazenamento de Mensagens

Um MHS deve oferecer a cada usuário um dispositivo de armazenamento para suas mensagens, tipo "caixa postal". Não existe necessidade de padrões para armazenamento e recuperação local de mensagens. O implementador deve desenvolvê-lo de acordo com o ambiente e as necessidades da comunidade de usuários.

Uma caixa postal pode ser implementada como um ou mais arquivos. Geralmente, uma entrada na caixa postal contém: a mensagem propriamente dita, o envelope que acompanha a mensagem, e informações sobre o estado da mensagem (ex. lida, não lida). Quanto aos tipos de acesso à caixa postal, os sistemas geralmente suportam: acesso a todas as mensagens, listagem em impressora, e seleção de mensagens segundo algum critério [61].

O MHS-88 define uma estrutura de armazenamento de mensagens como parte da nova entidade funcional Message Store (MS). O principal objetivo do MS é prover um dispositivo de armazenamento remoto para receber mensagens em lugar do UA (veja seção 3.1.1), podendo alternativamente, ser utilizado para outros objetivos.

Conforme visto em III.3.3, conceitualmente, o Message Store consiste de três "bases de informação". Cada base de informação é uma coleção de entradas (cada uma das quais é um objeto tal como uma mensagem entregue). As entradas são rotuladas com número de seqüência; entradas podem referir-se a outras entradas através destes números de seqüência. Uma caixa postal eficiente em termos de espaço pode ser implementada explorando-se este esquema de referência entre números de seqüência. Uma entrada pode ser logicamente duplicada referenciando-a em outra entrada e mantendo-se contadores de referência a fim de controlar as referências múltiplas. Desta forma, um simples MS pode implementar múltiplos mailboxes lógicos, e compartilhar uma única cópia de mensagem entregue entre todos os destinatários locais daquela mensagem [51].

IV.2.8 Aspectos não Abordados pelos Padrões X.400

IV.2.8.1 Roteamento

No MHS, os endereços O/R são a base para a seleção de rotas. Dado um endereço O/R, o MTS deve determinar a rota para o UA identificado por aquele endereço.

Os padrões de 1984 consideram somente o roteamento entre domínios de gerenciamento, o roteamento dentro de um domínio de gerenciamento está fora do escopo das Recomendações.

O MHS-84 inicialmente especifica que um esquema de roteamento incremental deve ser utilizado. Isto é, cada domínio de gerenciamento ao longo da rota da mensagem determina o MTA no próximo domínio de gerenciamento para o qual a mensagem deve ser transferida, mas não toma qualquer decisão além desta determinação. Não deve ser feita qualquer tentativa de estabelecimento de uma rota completa para a mensagem. Apesar da importância de um esquema deste tipo em um sistema com transferência de dados do tipo armazena-envia, o MHS-84 não padroniza e não faz qualquer sugestão sobre o método de escolha do próximo domínio de gerenciamento para o qual a mensagem será enviada.

Os padrões de 88 continuam deixando o esquema de roteamento a cargo da implementação mas sugerem alguns princípios de roteamento externo (entre domínios de gerenciamento). O roteamento interno continua completamente a cargo do implementador.

Uma solução que deve ser adotada pelos novos sistemas é o uso do Sistema de Diretório padronizado pela Recomendações X.500. O serviço de Diretório pode ser usado para armazenar informação relacionada às decisões de roteamento. Isto pode ser feito associando endereço O/R aos nomes dos MTAs através dos quais eles podem ser alcançados, com informação de conexão necessária para alcançá-los.

IV.2.8.2 Interface com Usuário

Não é trivial projetar uma interface com usuário para um sistema que dispõe de tantos serviços como o MHS. Criar, acessar e manipular mensagens contendo várias partes de diferentes tipos de codificação e ainda, tornar, tanto quanto possível, agradável a seleção das múltiplas opções por mensagem e por destinatário é uma difícil tarefa. Além disso, a interface deve atender as necessidades particulares da comunidade que utiliza o sistema. Portanto, os aspectos relativos a interface usuário-MHS estão fora do escopo de padronização tanto em 1984 como em 1988.

Como variados tipos de interface com usuário podem ser construídas sobre a entidade UA, que presta serviços padronizados, é desejável que a interface entre o UA e a interface com o usuário seja bem definida.

Um aspecto da interface com usuário que poderia se beneficiar da padronização, é a forma de especificação e apresentação de nomes O/R.

A nível de implementação, o sistema poderia oferecer ao usuário um diretório

pessoal que realizasse um mapeamento entre chaves definidas pelo usuário e nomes O/R correspondentes [62]. Este mapeamento seria feito tanto para mensagens em submissão como para mensagens recebidas. Uma forma reduzida de nome também pode ser utilizada para a apresentação do nome O/R. Um "nome-forma-livre" ("free-form name") pode opcionalmente acompanhar um nome O/R dentro de uma mensagem. Apresentar somente o nome-forma-livre, caso esteja presente, ou somente os atributos de usuário do nome O/R tal como nome pessoal, pode tornar a visão da mensagem mais agradável.

O formato interno de um nome O/R é estritamente definido pelos padrões X.400, apesar de uma grande variedade de estilos serem permitidos. O formato externo, ou seja, como estes nome aparecem ao usuário humano, é deixado como uma "questão local". A extensa variedade de nomes permitidos e a falta de uma representação comum para estes nomes cooperam para desafiar o implementador e confundir o usuário.

IV.2.9 Extensões aos Protocolos

IV.2.9.1 Protocolo P1

No caso dos Protocolos P1 e P3, a Recomendação X.411 diferencia as alterações sublinhando os elementos de protocolos adicionados ou alterados. As alterações são as seguintes:

(1) Restrições de Tamanho

Foi inserido restrições nos novos padrões para o comprimento dos tipos "string", para o número de itens nos tipos SET OF e SEQUENCE OF, para a extensão de valores do tipo inteiro. Estas restrições se aplicam a todos os parâmetros descritos na Recomendação X.411, exceto conteúdo de mensagens.

(2) Alterações nos Parâmetros Básicos

Os parâmetros "O/R name", "content-type" e "encoded-information-type", que são tipos comuns aos elementos do Protocolo P1, foram estendidos como segue:

(i) *O/R name*: Dois novos parâmetros opcionais foram adicionados a este parâmetro. O primeiro deles é um conjunto de atributos de extensão (extensions-attributes) que fornecem um meio para: utilizar caracteres teletex nos componentes "standard-attributes" e "domain-defined-attributes"; especificar um endereço postal para entrega física de mensagens; e especificar

um endereço de terminal pertencente a um endereço de rede estendido.

(ii) *Content-type*: Foi adicionada a opção de utilizar um identificador de objeto, ao invés de um inteiro, para identificar o tipo de conteúdo. Além disso, três novos valores foram definidos: "undefined", "external" e "interpersonal-messaging-1988".

(iii) *Encoded-information-types*: Foi adicionada a opção de especificar um conjunto de tipos de informação codificada definido externamente.

(3) Associação

Segundo a Recomendação X.411 1984, durante o estabelecimento de associação eram trocadas "senhas" do tipo ASN.1 "ANY". Na Recomendação X.411 1988, estas senhas são chamadas "credenciais" e podem ser "credenciais-simples", do tipo IA5String ou OCTECTSTRING), ou podem ser "credenciais-fortes", cujo tipo se baseia em técnicas de cifragem.

Um parâmetro opcional foi adicionado para especificar um contexto de segurança sob o qual estará toda a associação. Foi também adicionado um novo tipo de erro para indicar "contexto de segurança inaceitável".

(4) Parâmetro para Extensões

O parâmetro "extensions" foi adicionado para incluir a maioria das extensões feitas ao Protocolo P1. Este parâmetro está presente no envelope de transferência de mensagens, no envelope de transferência de sondagens e no envelope e conteúdo de transferência de notificações. Os componentes deste parâmetro particulares a cada um destes itens acima são definidos na Recomendação X.411 [15].

(5) Outras Diferenças

Dois parâmetros opcionais foram adicionados aos campos do envelope de transferência de notificações, a saber: "original-encoded-information-types" e "content-type".

Um parâmetro opcional para identificação de domínio privado (PRMD) foi adicionado ao parâmetro de "informação bilateral por domínio" pertencentes aos envelopes de transferência de mensagens e sondagens. Isto permite que a informação bilateral por domínio seja enviada não somente a ADMDs como também a PRMDs.

Um componente opcional para ("other-actions") foi adicionado ao parâmetro de informação de "trace". Este novo componente indica se o domínio de

gerenciamento (MD) redirecionou a mensagem e/ou expandiu uma Lista de Distribuição da mensagem.

IV.2.9.2 Protocolo P3

As alterações descritas nos itens (1), (2) e (3) do Protocolo P1 acima também se aplicam ao Protocolo P3. Este protocolo também sofreu alteração em suas operações remotas como segue:

(1) Submissão de Mensagens e Sondagens

Os parâmetros "original-encoded-information-types" e "explicit-conversion" se tornaram opcionais.

(2) Controle de Submissão

Foi adicionado ao argumento da operação o parâmetro opcional "permissible-security-context".

Foi adicionado ao resultado da operação o parâmetro opcional "waiting-content-types" de modo a indicar os tipos de conteúdo das mensagens que ficaram retidas devido a esta operação.

Foi adicionado o tipo de erro "security-error".

(3) Entrega de Mensagens

Os parâmetros "original-encoded-information-types" e "delivery-flags" se tornaram opcionais no envelope de entrega de mensagens, e um parâmetro opcional "content-identifier" foi adicionado a este envelope.

Esta operação se tornou confirmada através do retorno de seu resultado (RESULT). Este resultado contém dois parâmetros opcionais: "recipient-certificate" e "proof-of-delivery".

Foi adicionado o tipo de erro "security-error".

(4) Entrega de Notificações

Foram adicionados os parâmetros "original-encoded-information-types" e "content-type" ao envelope de entrega de notificações.

Foram definidos cinco novos códigos de "razões para não entrega de uma mensagem/sondagem", e trinta e cinco novos códigos de diagnóstico para não entrega de uma mensagem/sondagem.

Foram adicionados cinco novos valores para o parâmetro "type-of-MTS-user".

Esta operação se tornou confirmada através do retorno de seu resultado (RESULT). Este resultado não contém parâmetros.

(5) Controle de Entrega

Foram adicionados ao argumento da operação os parâmetros opcionais "permissible-content-types" e "permissible-security-context".

Foi adicionado ao resultado da operação o parâmetro opcional "waiting-content-types" de modo a indicar os tipos de conteúdo das mensagens que ficaram retidas devido a esta operação.

Foram adicionados os tipos de erro "control-violates-registration" e "security-error".

(6) Registro

Foram adicionados dois novos parâmetros ao argumento desta operação: "deliverable-content-types" e "labels-and-redirections".

Os parâmetros que indicam os tipos de operações permitidas e o comprimento máximo de conteúdo permitido foram alterados. O parâmetro "permissible-content-types" foi adicionado.

(7) Troca de Credenciais

Nas Recomendações de 1984 esta operação se chamava Troca de Senha. O tipo definido para as credenciais sofreram a alteração descrita no item (3) do Protocolo P1 acima. Foi imposta a restrição de que os tipos nos parâmetros "old-credentials" e "new-credentials" devem ser os mesmos.

IV.2.9.3 Protocolo P2

As alterações relevantes ao protocolo P2 trazidas pelos padrões X.400 1988 são as seguintes:

- (1) O tipo de conteúdo associado ao protocolo P2 foi alterado. Nos padrões X.400 1984, este tipo de conteúdo era identificado pelo valor inteiro "2". Nos novos padrões este tipo de conteúdo é identificado pelos valores "2" ou "22", dependendo da funcionalidade utilizada. O valor 2 é utilizado se o conteúdo segue a definição da Recomendação X.420 1984, ou seja, se o conteúdo é construído seguindo as seguintes restrições:

- (i) O cabeçalho (de uma mensagem-IP) não contém o campo "extensions".

(ii) O corpo (de uma mensagem-IP) não contém partes definidas externamente.

(iii) Os parâmetros de uma parte de corpo do tipo videotexto não contêm o componente "syntax".

(iv) Cada componente do conteúdo com tipo de dado definido no serviço do MTS (e.g., nome O/R) satisfaz a definição da Recomendação X.411 1984.

(v) O parâmetro "data" de qualquer parte de corpo satisfaz a estas restrições (recursivamente).

O valor 22 é utilizado se o conteúdo segue a definição da Recomendação X.420 1988 sem obedecer às restrições acima.

- (2) A omissão do componente "user" no parâmetro "IPMIdentifier" do cabeçalho da mensagem-IP é agora ignorado.
- (3) O parâmetro "extensions" foi adicionado ao cabeçalho da mensagem-IP. Seu grau é opcional.
- (4) Os tipos de parte de corpo "telex" e "simple formattable document" foram eliminados.
- (5) O parâmetro "syntax" foi adicionado à parte de corpo do tipo "videotext". Seu grau é opcional.
- (6) A presença do componente "delivery-time" no parâmetro "MessageParameters", pertencente ao tipo de parte de corpo "Message", é agora ignorado na ausência do componente "delivery-envelope" do mesmo parâmetro. Este tipo de corpo equivale ao chamado "ForwardedMessage" da Recomendação X.420 1984.
- (7) Os tipos de parte de corpo "bilaterally-defined" e "externally-defined" foram acrescentados.
- (8) Os seguintes elementos de protocolo definidos na Recomendação X.411 e incorporados aos elementos do Protocolo P2 foram alterados:
 - (i) O/R Name;
 - (ii) O/R Address;
 - (iii) MessageDeliveryEnvelope;
 - (iv) Encoded Information Types;
 - (v) Supplementary Information.

- (9) Atribuir um valor de comprimento zero a qualquer um dos seguintes tipos abaixo é agora ignorado:
- (i) LocalIPMIdentifier;
 - (ii) FreeFormName;
 - (iii) TelephoneNumber;
 - (iv) SubjectField;
 - (v) AutoForwardComment.
- (10) Foram impostos limites superiores sobre certos elementos de protocolo de comprimento variável. Estes limites foram obtidos da versão 6 do "X.400-series Implementor's Guide" [38].

IV.3 Proposta de Migração

Uma vez delineadas as vantagens que o uso do novo padrão traz em relação ao padrão de 1984, o próximo passo refere-se à implantação de uma nova configuração, que permita a implementação das extensões advindas com os padrões de 1988.

Na prática, estratégias de migração bem sucedidas são baseadas em um planejamento cuidadoso e em uma progressão gradual até o objetivo final, onde cada passo consiste da alteração de somente uma porção lógica do ambiente. Isto tende a isolar o efeito de conseqüências inesperadas, e minimiza o impacto sobre os usuários do sistema. Este trabalho sugere uma migração em duas fases claramente distintas.

Este processo de migração parte de uma implementação MHS de acordo com as Recomendações X.400 1984 que consiste de entidades de acesso (UA) e de transferência (MTA) centralizadas em um mesmo ambiente, e pretende obter uma configuração distribuída onde as entidades de acesso (UA) e de transferência (MTA) se localizam em ambientes distintos. Nesta configuração, as entidades UA utilizam entidades de armazenamento (MS) remotas e co-residentes com o MTA, e, a entidade MTA utilizará os serviços de Diretório padronizados pelas Recomendações X.500 do CCITT [21].

Em uma primeira versão, enquanto os padrões X.400 1988 não estão extensivamente disseminados e implementados, a entidade de transferência (MTA) se comunicará com outras de acordo com o Protocolo P1 1984. Isto permitirá a utilização local da funcionalidade estendida dos padrões X.400 1988 e da capacidade

de interoperabilidade dos sistemas X.400 1984 já implantados. As fases propostas para a migração serão descritas a seguir.

IV.3.1 Primeira Fase: Distribuição das Entidades de Acesso do MHS (UA) e Implantação da Entidade de Armazenamento (MS)

O primeiro objetivo desta migração é desenvolver a distribuição das entidades UA partindo de uma implementação onde estas entidades estão centralizadas em um mesmo ambiente. Estudos mostram que uma implementação distribuída oferece vantagens sobre uma implementação centralizada [40].

Esta fase consiste da implantação de entidades Message Store (MS) co-residentes com a entidade MTA já implantada em equipamento de grande porte de acordo com os padrões X.400 1984, e da implementação de UAs remotos em equipamento de pequeno porte de acordo com os padrões X.400 1988. Os MSs se comunicarão com os UAs através do Protocolo de Acesso ao MS (P7). Isto permite que UAs sejam implementados em microcomputadores do tipo PC sem o problema de disponibilidade do UA para a entrega de mensagens e os problemas relacionados ao Protocolo P3 (vide seção IV.2). A ênfase sobre o equipamento PC é devida ao seu baixo custo e sua grande disponibilidade em universidades, escritórios e companhias em geral. Com o progresso da tecnologia e a queda de custos será interessante que UAs sejam implementados também em estações de trabalho.

Nesta configuração, o usuário teria no MS as mensagens entregues pelo MTS, e adicionalmente, poderia ter no próprio UA uma outra área de mensagens para manter cópias de mensagens submetidas ou recebidas, conforme sua necessidade. Os comandos para gerenciamento de mensagens poderiam ser os mesmos para sua área de mensagens local e para o MS; o UA determinaria implicitamente quando um comando do usuário requer uma conexão com o MS.

A realização desta fase deve seguir os seguintes critérios:

- Modificar os comandos do usuário tão pouco quanto possível de modo que migração seja transparente ao usuário;
- Minimizar as modificações do software existente;
- Minimizar a quantidade de novo software a ser escrito e a complexidade do protocolo a ser utilizado entre UA e o MS.

Esta fase pode ser refinada como segue:

- Implementação da interface homem-máquina no equipamento remoto:

Os comandos de usuário já existentes devem ser mantidos tanto quanto possível, e a interface deve ser ampliada no sentido de tornar disponível ao usuário os novos serviços oferecidos, a saber, os serviços do MS e os novos serviços do UA.

- Construção do UA de acordo com a Recomendação X.420 1988 no equipamento remoto:

Isto inclui a implementação das ações automáticas e variáveis de estado descritas na seção III.3.2 e a alteração das unidades de dados do Protocolo P2 de acordo com os parâmetros particulares de projeto e a necessidades da comunidade de usuários.

- Implementação do Protocolo de Acesso ao MS, o Protocolo P7:

A parte do protocolo devida ao usuário deve ser implementada juntamente com o UA. A conexão com o MS é transparente ao usuário o UA estabelece a conexão com o MS implicitamente. A parte do protocolo devida ao provedor de serviços é implementada no MS.

- Implementação do MS no equipamento de grande porte onde já se encontra a implementação do MTA (84).
- Alteração da interface com o MTA, que antes era voltada para um UA (84), para que esta funcione com um MS (88). Já que o MS estará co-residente com o MTA, não é necessário o Protocolo P3, mas a interface entre estas entidades deve ser bem definida e deve seguir a filosofia daquele protocolo. Como neste caso já existe um MTA (84) e, portanto, também existe uma interface MTA/UA, esta interface deve ser alterada para que funcione com um MS (88).

Do ponto de vista da modelagem Estelle, a realização desta fase equivale a implementação dos sub-módulos UA_TYPE, MS_ACCESS_TYPE e MS_PROVIDER_TYPE da modelagem Estelle descrita na seção III.4 e detalhada no relatório técnico [55].

IV.3.2 Segunda Fase: Uso dos Serviços de Diretório e das Extensões do Protocolo P1

O MHS é projetado para ser um sistema de mensagens distribuído. Um importante aspecto de sistemas deste tipo é que operações freqüentes são sistematizadas.

A necessidade de serviços de diretório surge da constante evolução do sistema (ou de uma rede em geral), da necessidade de isolar os usuários de alterações na configuração do sistema, e do desejo de prover ao usuário uma visão mais amigável do sistema.

Um Sistema de Diretório (DS) pode prover serviços no âmbito de uma única rede ou de várias redes interconectadas. Para o MHS, o DS oferece um "servidor de nomes" que provê:

- Informações de Roteamento.
- Suporte à expansão de Listas de Distribuição.
- Especificação amigável de endereços (nomes O/R) para usuários.

A importância e o impacto do DS no MHS e dos aspectos acima citados foram descritos em II.3.6, IV.2.5, IV.2.6 e IV.2.8.

Esta fase pode ser refinada como segue:

- Implantação do Sistema de Diretório (DS) padronizado X.500 [21].
- Implantação da estrutura UA/DUA e MTA/DUA (descrito em II.3.6). Implementação das interfaces MHS/DS.
- Utilização da estrutura estabelecida acima para a resolução de endereços. Isto implica na alteração da estrutura do nome O/R para estrutura X.400 1988. Entretanto, deve-se notar que o escopo desta alteração limita-se a interface entre o MTA e o usuário de seus serviços, e a interface entre UA e usuário final, já que em um primeiro momento o MTA não se comunicará com outros com o Protocolo P1 1988.
- Implantação das Listas de Distribuição com expansão no MTA local.
- Utilização do DS para obtenção de informações de roteamento.

IV.4 Implementação da Fase 1

A seguir é descrita uma proposta para a implementação da primeira fase de migração apresentada na seção anterior. São descritas a arquitetura e estruturas de dados para a implementação de um UA a ser realizada no ambiente PC/DOS utilizando a linguagem C e para a implementação de um MS a ser realizada no ambiente VAX/VMS utilizando a linguagem C.

IV.4.1 Arquitetura e Estruturas de Dados do Agente Usuário

IV.4.1.1 Arquitetura Interna

A arquitetura interna do UA divide suas funções em quatro módulos principais: o módulo **Interface-Usr**, o módulo **UA-Gerente**, o módulo **UA-Msg**, e o módulo **UA-P7**. Esta arquitetura é ilustrada na Figura IV.1.

O módulo **Interface-Usr** é o ponto de contato entre o sistema de mensagens e o usuário. Este módulo apresenta ao usuário toda a gama de recursos de manipulação de mensagens disponível, além de fornecer facilidades locais para edição, composição e envio de mensagens bem como a seleção de mensagens previamente recebidas.

Os serviços do módulo **Interface-Usr** são oferecidos através de comandos selecionados pelo usuário através de menus e helps. Estes comandos podem ativar dois tipos de funções: funções executadas no próprio módulo ou funções executadas no módulo **Interface-usr** ou no módulo **UA-Gerente**. As funções próprias do módulo **Interface-usr** são aquelas de âmbito local e correspondem as facilidades estritamente locais, não sendo objeto de padronização. As funções executadas no módulo **UA-Gerente** são aquelas que requerem interação com o MS remoto. Estas funções correspondem aos elementos de serviço de IPM, MS e MT oferecidos pelo sistema.

O módulo **UA-Gerente** é responsável por mapear solicitações do módulo interface em unidades de dados do Protocolo P2, operações remotas do Protocolo P7 e primitivas de serviço para o CASE e ativar a máquina de estados do UA. Por outro lado, este módulo torna disponível para o usuário qualquer resposta à operações previamente submetidas. Este módulo também ativa/desativa ações automáticas do UA e realiza operações de gerenciamento que visam controlar as interações entre UA e MS.

O módulo de **UA-Msg** opera a parte de armazenamento e seleção local de mensagens. As operações de armazenamento permitem que mensagens possam ser preparadas previamente e armazenadas já codificadas em sintaxe padrão. As operações de seleção auxiliam o usuário na busca de mensagens armazenadas. Estas operações correspondem a facilidades locais, não sendo objeto de padronização.

O módulo **UA-P7** processa a máquina lógica que opera a transição de estados do UA, comunicando-se com a MS remoto através do Protocolo de Acesso ao MS, o Protocolo P7. Este módulo utiliza os serviços dos Elementos de Serviço Comuns da Camada de Aplicação (CASE), a saber: o ACSE, que estabelece e controla a

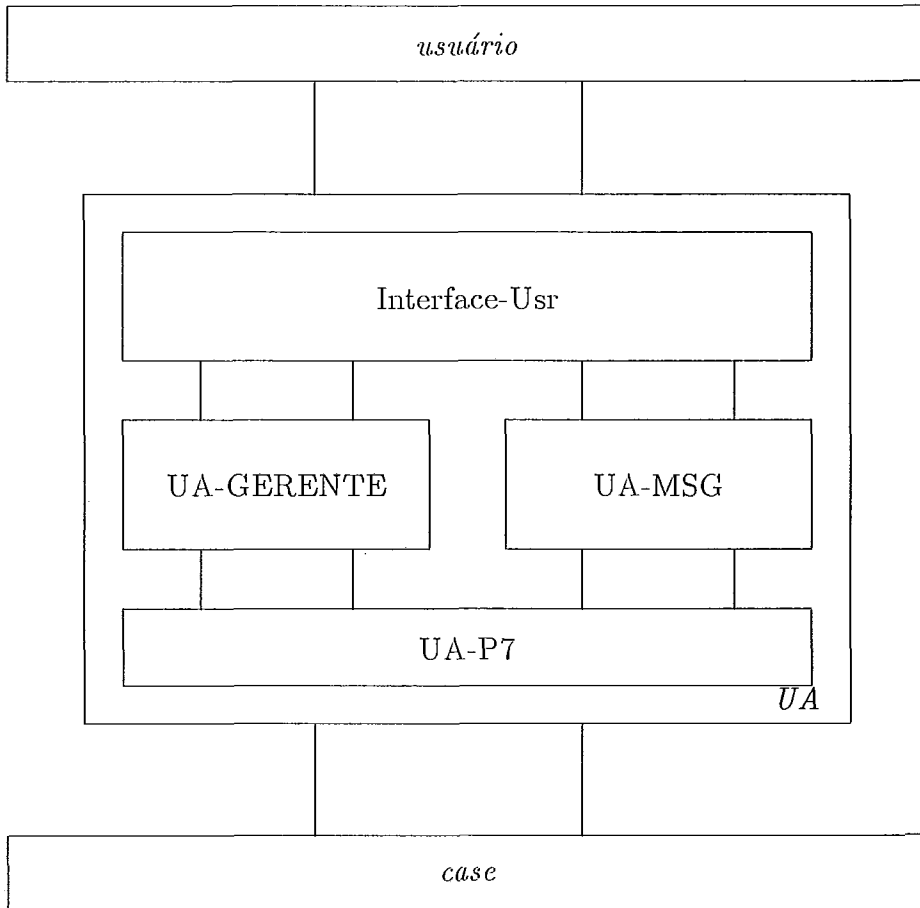


Figura IV.1: Arquitetura de Implementação do UA

associação entre as entidades e o ROSE, que realiza as operações remotas. Este módulo estabelece o mecanismo que rege a transição de estados do protocolo P7 baseada na ocorrência de eventos.

IV.4.1.2 Estruturas de Dados

Quanto às estruturas de dados, o modelo de implementação propõe que o UA opere sobre áreas de dados, tabelas de informação, filas de distribuição de eventos e listas.

A área de dados é composta das áreas das mensagens recebidas da entidade MS remota.

A cada UA estão associadas tabelas de informações e filas de distribuição de eventos.

As filas de distribuição de eventos são do tipo FIFO (First-in, First-out) e contém apontadores de eventos esperando para serem tratados pela máquina de

transição de estados. Estas filas são utilizadas enquanto a associação com o MS remoto ainda não foi completada.

O UA mantém duas estruturas de dados para armazenamento de informações relativas às operações sendo realizadas. Estas estruturas são listas encadeadas que são criadas e manipuladas dinamicamente em memória conforme as operações são solicitadas e realizadas.

As tabelas armazenam dados relativos a mensagens, as operações remotas, ao UA e ao MS; seus campos podem ser alterados pelo UA durante uma sessão de diálogo entre UA e MS de acordo com as operações invocadas por ambas as partes. As tabelas são armazenadas em disco e manipuladas em memória, para onde são copiadas em tempo de inicialização. As tabelas e listas são detalhadas a seguir.

(i) Tabelas

TAB_MSG_PRB (Tabela Relativa à Mensagem) Esta tabela contém informações necessárias ao UA relativas a:

- sondagens submetidas;
- mensagens que aguardam algum tipo de notificação;
- mensagens submetidas para entrega pré-datada.

Para cada mensagem é alocado dinamicamente um registro na tabela. O registro é deletado da tabela quando a UAE recebeu todas as notificações esperadas para a mensagem.

Cada registro contém os seguintes campos:

Tipo	UA_Cont_Id	Event_Id	Data_Sub	N_Notif	UA_Id_Original
------	------------	----------	----------	---------	----------------

Onde:

- Tipo: Indica se o registro se refere a mensagem ou a sondagem.
- UA_Cont_Id: Identificador único da mensagem submetida, gerado pelo UA. Este identificador é também transportado ao MS na operação de Submissão de Mensagens.
- Event_Id: Identificador único do evento de submissão de mensagem ou sondagem, gerado pelo MTA. Este identificador é retornado pelo resultado da operação de Submissão de Mensagens.

- Data_Sub: Indica data e hora da submissão da mensagem ao MTA. Este parâmetro é retornado pelo resultado da operação de Submissão de Mensagens.
- N_Notif: Indica o número de notificações aguardadas pela mensagem ou sondagem.
- UA_Id_Original: Este campo é utilizado por mensagens que possuem destinatários cópia cega e armazena o identificador da mensagem original.

TAB_UA (Tabela Relativa ao UA) Esta tabela armazena todas as informações pertinentes ao UA tais como, credenciais de acesso ao MS, nome O/R do usuário, variáveis de estado, capacidades de submissão de mensagens e informações de registro do UAE, o estado da UAE. Estes valores de registro também são mantidos no MTA. Esta tabela contém somente um registro cujos campos podem ser alterados pelo UA e são descritos abaixo.

Registro-UA	Credenciais	Subm_Ctrl	State_Var	Restrições	Estado-UA	Nome_Livre
-------------	-------------	-----------	-----------	------------	-----------	------------

Onde,

- Registro-UA: Armazena as informações de registro do UA que também são mantidas no MTA. Este campo contém seis sub-campos:

Tipo_Cod	Tam_Max	Prior_Min	Flg_Msg	Flg_Prpb	Nome_OR
----------	---------	-----------	---------	----------	---------

Onde,

- Tipo_Cod: Indica os tipos de informação codificada que o MTA pode transmitir ao UA
- Tam_Max: Indica o comprimento máximo do conteúdo que o MTA pode transmitir ao UA
- Prior_Min: Indica a prioridade mínima da mensagem que a MTA pode transmitir a UA
- Flg_Msg: Indica se a MTA pode ou não entregar mensagens ao UA
- Flg_Not: Indica se a MTA pode ou não entregar notificações ao UA

- Nome_OR: Indica o nome O/R do usuário (UA)
- Credenciais: Indica as credenciais de acesso do usuário (UA) ao MS
- Subm_Ctrl: Contém os valores correntes de controle de submissão de mensagens/sondagens. Estes valores podem sofrer alterações durante um diálogo entre UA e MS e se subdivide em cinco sub_campos:

Tipo_Cod	Tam_Max	Prior_Min	Flg_Msg	Flg_Prb
----------	---------	-----------	---------	---------

- Tipo_Cod : Indica os tipos de codificação permitidos nas mensagens/sondagens que o UA submete.
 - Tam_Max : Indica o comprimento máximo do conteúdo que o UA pode submeter.
 - Prior_Min: Indica a prioridade mínima permitida para mensagens/sondagens que o UA submete.
 - Flg_Msg : Indica se o UA pode ou não submeter mensagens.
 - Flg_Prb : Indica se o UA pode ou não submeter sondagens.
- State_Var: Armazena os valores das variáveis de estado que definem as ações automáticas que o UA deve realizar para seu usuário. Este campo se subdivide em dois sub-campos:

Auto_Discard	Auto_Ack
--------------	----------

Onde,

- Auto_Discard: Armazena informações pertinentes à ação de descarte automático de mensagens obsoletas e/ou expiradas, conforme definido na Recomendação X.420 [18].
 - Auto_Ack: Armazena informações pertinentes à ação de confirmação automática de mensagens recebidas, conforme definido na Recomendação X.420 [18].
- Restrições: Armazena as restrições correntes sobre a operação de "Fetch" do Message Store. Este campo se subdivide em três sub-campos:

Cont.Types	EIT	Cont.Lenght
------------	-----	-------------

Onde,

- Cont.Types: Indica os tipos de conteúdos que podem ser solicitados em um operação de fetch.
- EIT: Indica os tipos de informação codificada que podem ser solicitados em um operação de fetch.
- Cont.Lenght: Indica o comprimento máximo de conteúdo que pode ser solicitado em um operação de fetch.
- Estado.UA: Indica o estado em que se encontra a máquina de estados do UA.
- Nome.Livre: Indica um nome amigável para identificação local do usuário.

TAB_MS (Tabela Relativa ao MS) Esta tabela armazena todas as informações pertinentes ao MS ao qual o UA está associado. As informações desta tabela reflete a configuração do MS e podem ser alteradas pela administração do sistema ou por operações remotas disparadas pelo UA. Esta tabela contém somente um registro cujos campos são descritos abaixo.

Credenciais	Registro_MS	Auto_Act_Avail	Att_Type_Avail	Contents_Type
-------------	-------------	----------------	----------------	---------------

Onde,

- Credenciais: Armazena credenciais do MS que são validadas pelo UA durante o estabelecimento de uma associação.
- Registro_MS: Armazena informações mantidas no MS que são utilizadas na execução de certas operações como mostram seus sub-campos abaixo descritos. Estas informações são configuradas pelo usuário através da operação "Register_ms" do Message Store.

Auto_Act_Reg	Fetch_Att	List_Att
--------------	-----------	----------

Onde,

- Auto_Act_Reg: Indica quais as ações automáticas que estão correntemente ativas no MS.
- Fetch_Att: Indica os atributos que são utilizados em uma operação "fetch" no caso em que o usuário não fornece estes parâmetros.
- List_Att: Indica os atributos que são utilizados em uma operação "list" no caso em que o usuário não fornece estes parâmetros.

- Auto_Act_Avail: Indica os tipos de ações automáticas disponíveis no MS remoto.
- Att_Type_Avail: Indica os tipos de atributos disponíveis nas entradas do MS remoto.
- Contents_Type: Indica os tipos de conteúdo de mensagens que podem ser manipulados pelo MS.

(ii) Listas

LIST_INV (Lista Relativa à Operações Remotas) Esta lista é utilizada para armazenar informação sobre quaisquer operações solicitadas pelo UA ao MS remoto. É criado um item da lista para cada operação. Cada item possui os seguintes campos:

Id	Ação_Usr	Wait_Result
----	----------	-------------

Onde,

- Id: Contém o identificador da operação submetida para transferência pelo ROSE. É idêntico ao parâmetro "invoke-id" da primitiva RO-INVOKE do ROSE.
- Ação_Usr: Indica se a operação é fruto de ação explícita do usuário ou é fruto de uma ação automática do UA.
- Wait_Result: Indica se a operação em questão está aguardando um resultado.

LIST_SUMM (Lista Referente à Operação Summarize) Esta lista é utilizada para armazenar informação sobre a operação "summarize" do MS solicitada pelo usuário. Esta informação é usada posteriormente para correlacionar o resultado da operação com o pedido do usuário. É criado um item para cada operação summarize solicitada pelo usuário. Cada item possui os seguintes campos:

Id	Summ
----	------

Onde,

- Id: contém o identificador da operação submetida para transferência pelo ROSE. É idêntico ao parâmetro "invoke-id" da primitiva RO-INVOKE do ROSE.
- Summ: Armazena a informação solicitada pelo usuário na operação summarize. Esta informação é usada posteriormente para correlacionar o resultado da operação com o pedido do usuário.

Listas de Mensagens e Sondagens Retidas O UA utiliza duas listas encadeadas e dois registros utilizados como "cabeças" destas duas listas. Uma lista é utilizada para armazenar mensagens retidas e outras para sondagens retidas para posterior submissão.

Os registros de cabeça de lista, contêm apontadores para o início e fim de cada lista.

Formato dos registros de cabeça de lista:

Tipo	iptr	fptr
------	------	------

Onde,

- Tipo: Indica se a lista é de mensagens ou de sondagens;
- iptr: Indica o início da lista;
- fptr: Indica o fim da lista.

São necessários dois registros para a inserção de uma mensagem/sondagem na lista de mensagem/sondagem retidas para o MTA. O primeiro conteria informações sobre a última mensagem/sondagem da lista e o segundo conteria informações sobre a mensagem/sondagem a ser inserida. Esse último registro é também utilizado para a leitura de uma mensagem/sondagem da lista.

Os registros das duas listas contêm os mesmos campos da operação "Message-Submission na lista de mensagens ou da operação ProbeSubmission na lista de sondagens.

Os registros apresentam o seguinte formato:

nLst	tam	prior	eventId	oper
------	-----	-------	---------	------

Onde,

- nLst : Aponta para o próximo da lista;
- tam : Tamanho da mensagem;
- prior : Prioridade de mensagem;
- oper : Contém o formato da operação MessageSubmission/ProbeSubmission pelo UA para o MS.

As listas são armazenadas e manipuladas em disco.

IV.4.1.3 As Interfaces do UA

Uma Entidade de Agente Usuário deve se comunicar-se ora com o usuário, para prestar-lhe os serviços de manipulação de mensagens, ora com os elementos de serviço comuns da camada de aplicação (CASE), os quais efetuam estabelecimento de conexão e transferência de dados ao MS remoto. As duas interfaces através das quais o UA se comunica são: a interface com o módulo Interface e a interface com o módulo CASE.

Interface UA/CASE A interface entre UA e MTA engloba todas as primitivas de serviço que são oferecidas pelo ACSE [29,23] e ROSE [30,25]. Sua utilização foi descrita em III.3.3. As primitivas do ACSE são:

- A-ASSOCIATE

- A-RELEASE
- A-ABORT
- A-P-ABORT

E as primitivas do ROSE são:

- RO-INVOKE
- RO-RESULT
- RO-ERROR
- RO-REJECT-U
- RO-REJECT-P

Estas primitivas são definidas em III.3.5.

Interface UA/ Módulo Interface A maneira segundo a qual os serviços de MH são apresentados ao usuário não faz parte do escopo da série de Recomendações X400. No caso do MHS-NCE, existe um módulo de auxílio ao usuário, chamado módulo interface, o qual é ligado ao módulo UAE. Por intermédio deste módulo, o usuário pode selecionar os recursos desejados de forma clara e amigável.

Apesar da não padronização da interface homem-máquina no MHS, entende-se que a interface entre o UA e estes serviços deve ser bem definida tendo em vista a variedade de tipos de interfaces homem-máquina que podem ser construídas sobre o UA.

A informação entre estes módulos é trocada sob a forma de primitivas de serviço que foram definidas localmente. Existem dez primitivas entre os módulos UAE e Interface. Estas primitivas são utilizadas para que o usuário possa solicitar os serviços de MH e para que o UA possa passar ao usuário alguma informação sobre os serviços solicitados. A seguir estas primitivas são descritas:

- Primitivas Invocadas pelo módulo Interface:

Os parâmetros das primitivas abaixo são descritos nas Recomendações X.411 e X.407 do CCITT.

PSUB : Submissão de Mensagens

Parâmetros : IM-UAPDU
NDN-Supress
Deferrid-delivery-time
Delivery-notice
Conversion-Prohibited
Disclose-Recipients
Alternate-recipient-allowed

PPRB : Submissão de Sondagens

Parâmetros : UA-content-id
Recipient-O/R-names
Alternate-Recipient-allowed
Content-lenght

PCAN : Cancelamento de Entrega Pré-datada de Mensagens

Parâmetro : UA-content-id

PSUMM : Solicita sumários sobre entradas do MS

Parâmetros : Selector
Summary-requests

PLIST : Solicita certos atributos de entradas do MS

Parâmetros : Selector
Entry Information Selection

PFETCH : Solicita quaisquer atributos de uma entrada do MS

Parâmetros : Item
Entry Information Selection

PDELETE : Deleta entrada(s) no MS

Parâmetros : Items

PREGISTER_MS : Altera configuração do MS.
Parâmetros : Auto-action-registrations
Auto-action-deregistrations
List-attributes-defaults
Fetch-attributes-defaults
Change-credentials

PSTATE_VAR : Altera os conteúdos das variáveis de estado.

PUNBIND : Solicita o término da interação
Esta primitiva não contém parâmetros

- Primitiva Transmitida pelo UA ao Módulo Interface:

Informa_usr : Transmite mensagem de "status" ao usuário
Parâmetro : Mensagem
Tipo : Cadeia de caracteres

IV.4.2 Arquitetura e Estruturas de Dados do Message Store (MS)

IV.4.2.1 Arquitetura Interna

A arquitetura interna do MS divide suas funções em três módulos principais: o módulo **MS-P7**, o módulo **MS-Base**, e o módulo **MS-MTA**. Esta arquitetura é ilustrada na Figura IV.2.

O módulo **MS-MTA** é responsável por gerenciar a interação entre MS e MTA, estabelecendo a comunicação com o MTA e mapeando solicitações de serviços do usuário em operações que serão passadas ao MTA. Este módulo também recebe mensagens e notificações do MTA e ativa o módulo **MS-Base** que gerará atributos e entradas no MS para posterior acesso pelo usuário.

O módulo **MS-Base** é responsável pela execução de operações, acesso e manutenção da base de informação que armazena mensagens e notificações entregues pelo MTA ao MS.

Este módulo pode ser ativado pelo módulo **MS-MTA** que o entrega mensagens e notificações para geração de atributos e entradas que serão posteriormente acessadas pelo usuário; e pode ser ativado pelo módulo **MS-P7** que solicita a execução de operações na base de informação a pedido do usuário remoto.

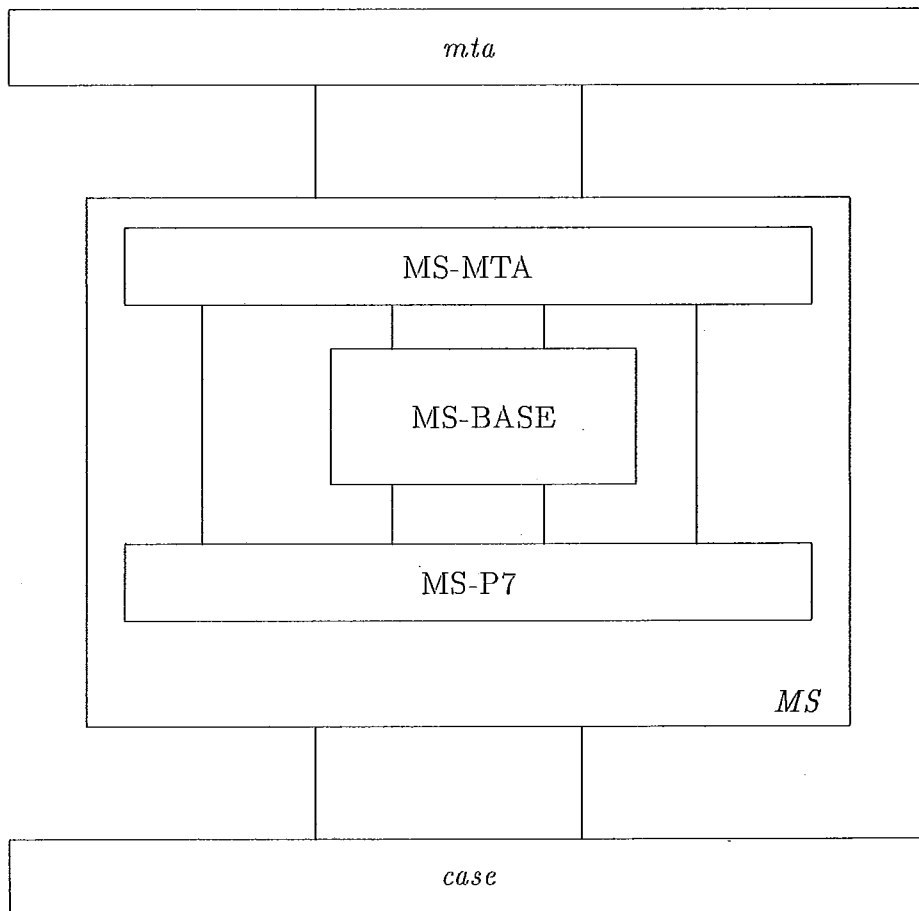


Figura IV.2: Arquitetura de Implementação do MS

A estrutura proposta para esta base de informação e a geração de seus atributos são descritas posteriormente.

O módulo **MS-P7** processa a máquina de transição de estados do Protocolo de Acesso ao MS, o Protocolo P7, baseado na ocorrência de eventos provocados pelo UA remoto. No Protocolo P7, este módulo opera parte do provedor de serviços de MS. Este módulo utiliza os serviços dos Elementos de Serviço Comuns da Camada de Aplicação (CASE), a saber: o ACSE, que estabelece e controla a associação entre as entidades e o ROSE, que realiza as operações remotas.

Dependendo da operação invocada pelo UA remoto, este módulo ativa o módulo MS_Base para o acesso à base de informação e execução da operação solicitada pelo usuário; e/ou ativa o módulo MS-MTA que transfere a operação ao MTA provendo assim serviços de submissão indireta. Este módulo pode também ativar o módulo MTA se ele necessita executar alguma ação automática decorrente da entrega de mensagens.

IV.4.2.2 Estruturas de Dados

Quanto às estruturas de dados, o modelo de implementação propõe que o MS opere sobre áreas de dados em memória e em disco, tabelas de informação, filas de distribuição de eventos e listas.

A área de dados em memória é utilizada para as mensagens recebidas da entidade UA remota ou da entidade MTA.

A área de dados em disco corresponde a base de informação do MS, a qual armazena mensagens e notificações entregues ao MS pelo MTA.

As filas de distribuição de eventos são do tipo FIFO (First-in, First-out) e contém apontadores de eventos provocados pelo MTA ou pelo UA remoto esperando para serem tratados pela máquina de transição de estados.

O MS mantém duas estruturas de dados para armazenamento de informações relativas às operações sendo realizadas. Estas estruturas são listas encadeadas que são criadas e manipuladas dinamicamente em memória conforme as operações são solicitadas e realizadas.

As tabelas armazenam dados relativos a mensagens, as operações remotas, ao MS, ao UA e ao MTA; seus campos podem ser alterados pelo MS durante uma sessão de diálogo com o UA e/ou MTA de acordo com as operações invocadas por ambas as partes, ou podem também ser alteradas pela administração do sistema.

As tabelas são armazenadas em disco e manipuladas em memória, para onde são copiadas em tempo de inicialização. A base de informação, as tabelas e listas são detalhadas a seguir.

(i) Base de Informação

A Base de Informação armazena mensagens e notificações entregues ao MS pelo MTA para posterior acesso pelo usuário remoto. A base constitui-se de entradas que por sua vez são compostas de atributos que são gerados a partir dos elementos que compõem a mensagem ou notificação entregue. As entradas são acessadas pelo seu número de seqüência.

Para cada mensagem ou notificação é criada pelo menos uma entrada na base de informação. Eventualmente, podem ser criadas mais de uma entrada que serão entradas "filho".

Apesar das entradas em uma base de informação serem independentes umas das outras, o modelo de informação do MS permite que as entradas estejam relacionadas entre si.

Uma entrada pode ser "filha" (entrada-filho) de uma outra, chamada "entrada-

pai” em uma organização do tipo árvore. Uma entrada que não é uma entrada-filho é dita entrada-principal.

Este relacionamento é registrado através de dois atributos especiais:

- (a) número-sequência-pai: Este atributo de valor único dá o número de sequência da entrada-pai de uma entrada-filho. Está ausente em uma entrada-principal.
- (b) número-sequência-filho: Este atributo de valor múltiplo dá os números de sequência de todas as entradas-filho de uma entrada-pai. Está ausente em uma entrada que não seja uma entrada-pai.

As operações-abstratas do dado MS agem por default somente em entrada-principais. Algumas delas podem ser direcionadas para agirem em todas as entradas, principais e filhos. Em particular, o argumento de uma operação Delete pode ser somente números de seqüência de entradas-principais. Neste caso a entrada-principal, todas as suas entrada-filho e as entrada-filho das entrada-filho etc, também serão deletadas.

Por exemplo, as partes do corpo de uma mensagem-IP podem ser representadas por entradas-filho individuais. O atributo ”conteúdo” da entrada-principal contém o conteúdo completo, e os dados que representam as partes do corpo da mensagem estão logicamente presentes em mais que uma entrada.

A base de informação de mensagem armazenada age como um depósito para informação obtida das operação-abstrata de Entrega de Mensagens e Notificações; contendo entradas para mensagens e notificações entregues pelo MTS. Uma entrada no base de informação de mensagem armazenada é criada quando uma mensagem ou uma notificação chega ao MS.

Para extrair informação do conteúdo de uma mensagem, O MS deve conhecer a sintaxe e a semântica do conteúdo, conforme sinalizado pelo parâmetro ”tipo do conteúdo”. Em geral, uma instância de MS conhece zero ou mais tipos de conteúdo. Quando um MS recebe uma mensagem cujo tipo de conteúdo ele não conhece suficientemente, ele não estará apto a gerar os atributos específicos do tipo de conteúdo (content-type-specif) na entrada da mensagem.

Uma mensagem ou notificação entregue pode resultar em uma entrada-principal e um ou mais níveis de entrada-filho. Por exemplo, quando uma notificação de não-entrega contém um conteúdo (returned-content), a entrada da notificação (delivered-report entry) é a entrada-principal e o conteúdo é a sua entrada-filho (returned-content entry).

As regras pelas quais um conteúdo de mensagem pode ser dividida entre várias entradas são específicas de cada tipo de conteúdo. Um atributo específico, chamado atributo de sinopse pode ser usado para mostrar como a entrada-principal e as

entradas-filho correspondentes estão relacionadas.

Este atributo é definido na Recomendação que define o tipo de conteúdo propriamente dito mas é construído pelo MS.

Por exemplo, no caso de mensagem-IP, mensagem-IP aninhadas dentro de uma mensagem-IP são representadas como entradas-filho. O atributo "ipm-synopses" é um exemplo de um atributo de sinopse específico.

Uma propriedade importante de uma entrada na base de informação de mensagem armazenada é o seu estado (entry-status). Ele é criado e mantido pelo MS e pode assumir os seguintes valores:

- a) Nova(New): A mensagem não foi "listada" pelo UA e nem foi processada automaticamente pelo MS;
- b) Listada(Listed): A informação sobre a mensagem foi retornada ao UA em uma outra "List" ou "Fetch", mas a mensagem não foi ainda completamente processada;
- c) Processada(Processed): A mensagem foi "completamente recuperada" pelo UA ou o MS analisou alguma auto-ação (auto-action) sobre ela. A definição exata de "completamente recuperada" é específica do tipo de conteúdo da mensagem e é definida na Recomendação correspondente.

O estado da entrada de uma notificação assume o valor "processado" quando o envelope de relatório (delivered-report-envelope) é recuperado pelo usuário.

Para uma proposta inicial, uma entrada contém os seguintes atributos:

Num_Seq Número de seqüência da entrada. Este atributo está presente em qualquer tipo de entrada, e é gerado pelo MS no momento de criação da entrada. O MS atribui a este atributo um valor único em ordem ascendente.

Data_Criação Data de criação da entrada. Este atributo está presente em qualquer tipo de entrada, e é gerado pelo MS no momento de criação da entrada. O MS atribui a este atributo a data de criação da entrada.

Estado_Entrada Indica o estado da entrada. Pode assumir três valores: nova, listada e processada. Este atributo está presente em qualquer tipo de entrada, e é gerado pelo MS no momento de criação da entrada.

Tipo_Entrada Indica o tipo de entrada. É criado pelo MS a partir da informação recebida do MTA. Assume o valor "mensagem-entregue" se a entrada é criada

a partir de uma mensagem, ou o valor "notificação-entregue" se a entrada é criada a partir de uma notificação. Se a notificação retorna um conteúdo de mensagem, é criada uma entrada filho para armazenar este conteúdo. Para esta entrada filho, o tipo de entrada assume o valor "conteúdo-retornado".

Num_Seq_Filho Atributo gerado pelo MS. Indica os números de seqüência das entradas filho desta entrada. Atributo pode estar presente em qualquer tipo de entrada.

Conteúdo Atributo obtido a partir do parâmetro conteúdo (content) de uma mensagem ou do parâmetro conteúdo-retornado (returned-content) de uma notificação. Atributo presente somente em entradas do tipo "mensagem-entregue" ou "conteúdo-retornado".

Tam_Conteúdo Atributo gerado pelo MS para indicar o tamanho do conteúdo da mensagem entregue ou retornada, em octetos. Atributo presente somente em entradas do tipo "mensagem-entregue" ou "conteúdo-retornado".

Tipo_Conteúdo Atributo obtido a partir do parâmetro correspondente do envelope da mensagem entregue ou retornada, em octetos. Atributo presente somente em entradas do tipo "mensagem-entregue" ou "conteúdo-retornado".

Indicadores_Entrega Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Se este parâmetro não está presente no envelope, o atributo é criado com o valor default do parâmetro. Atributo presente somente em entradas do tipo "mensagem-entregue".

Dest_Original Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo pode estar presente somente em entradas do tipo "mensagem-entregue".

Envelope_Msg Atributo obtido a partir do envelope de entrega de mensagens. Atributo presente somente em entradas do tipo "mensagem-entregue".

Id_Entrg_Msg Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo presente somente em entradas do tipo "mensagem-entregue".

Data_Entrg_Msg Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo presente somente em entradas do tipo "mensagem-entregue".

Data_Subm_Msg Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo presente somente em entradas do tipo "mensagem-entregue".

Originador Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo presente somente em entradas do tipo "mensagem-entregue".

Outros_Dest Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo presente somente em entradas do tipo "mensagem-entregue".

Num_Seq_Pai Atributo criado pelo MS para indicar os números de seqüência da entrada pai da entrada em questão, caso haja. Atributo pode estar presente somente em qualquer tipo de entrada.

Prioridade Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo presente somente em entradas do tipo "mensagem-entregue".

Id_Conteúdo Atributo obtido a partir do parâmetro correspondente do envelope de entrega de mensagens. Atributo pode estar presente somente em entradas do tipo "mensagem-entregue" ou do tipo "notificação-entregue".

Retorna_Cont Atributo criado pelo MS para indicar se a notificação recebida retorna um conteúdo de mensagem não entregue. Atributo presente somente em entradas do tipo "mensagem-entregue". Conseqüentemente, este atributo indica se esta entrada possui uma entrada filho que armazena um conteúdo retornado. Atributo presente somente em entradas do tipo "notificação-entregue".

Info_Por_Dest Atributo obtido a partir do parâmetro correspondente do envelope de entrega de notificações. Atributo presente somente em entradas do tipo "notificação-entregue".

Envelope_Not Atributo obtido a partir do envelope de entrega de notificações. Atributo presente somente em entradas do tipo "notificação-entregue".

Id_Msg_Subm Atributo obtido a partir do parâmetro correspondente do envelope de entrega de notificações. Atributo presente somente em entradas do tipo "notificação-entregue".

Para cada entrada são também criados mais dois parâmetros que auxiliam o acesso e manutenção da base de informação:

Deleted Marca a entrada ara deleção.

Presents Indica que a atributos estão presentes na entrada em questão.

(ii) Tabelas

TAB_MS (Tabela Relativa ao MS) Esta tabela contém todas as informações pertinentes ao MS ao qual o UA está associado. As informações desta tabela reflete a configuração do MS e podem ser alteradas pela administração do sistema ou por operações remotas disparadas pelo UA. Esta tabela contém somente um registro cujos campos são descritos abaixo.

Credenciais	Registro_MS	Auto_Act_Avail	Att_Type_Avail	Contents_Type	Senha	Alert
-------------	-------------	----------------	----------------	---------------	-------	-------

Onde,

- Credenciais: Armazena credenciais do MS que são validadas pelo UA durante o estabelecimento de uma associação.
- Registro_MS: Armazena informações mantidas no MS que são utilizadas na execução de certas operações como mostram seus sub-campos abaixo descritos. Estas informações são configuradas pelo usuário através da operação "Register_ms" do Message Store.

Auto_Act_Reg	Fetch_Att	List_Att
--------------	-----------	----------

- Auto_Act_Reg: Indica quais as ações automáticas que estão correntemente ativas no MS.
 - Fetch_Att: Indica os atributos que são utilizados em uma operação "fetch" no caso em que o usuário não fornece estes parâmetros.
 - List_Att: Indica os atributos que são utilizados em uma operação "list" no caso em que o usuário não fornece estes parâmetros.
- Auto_Act_Avail: Indica os tipos de ações automáticas disponíveis no MS remoto.

- Att_Type_Avail: Indica os tipos de atributos disponíveis nas entradas do MS remoto.
- Contents_Type: Indica os tipos de conteúdo de mensagens que podem ser manipulados pelo MS.
- Senha: Senha de acesso ao MTA. Utilizada para compatibilização com o MTA 1984.
- Alert : Indica se existe operação "alert" pendente esperando associação com UA remoto para ser realizada.

TAB_UA (Tabela Relativa ao UA) Esta tabela armazena todas as informações pertinentes ao UA tais como, credenciais de acesso ao MS, nome O/R do usuário, e registro do UA. Esta tabela contém somente um registro cujos campos são descritos abaixo.

Registro_UA	Credenciais	Restrições	Nome_Livre
-------------	-------------	------------	------------

Onde,

- Registro_UA: Armazena as informações de registro do UA que também são mantidas no MTA. Este campo contém seis sub-campos:

Tipo_Cod	Tam_Max	Prior_Min	Flg_Msg	Flg_Prpb	Nome_OR
----------	---------	-----------	---------	----------	---------

- Tipo_Cod : Indica os tipos de informação codificada que o MTA pode transmitir ao UA.
 - Tam_Max : Indica o comprimento máximo do conteúdo que o MTA pode transmitir ao UA.
 - Prior_Min: Indica a prioridade mínima da mensagem que a MTA pode transmitir a UA.
 - Flg_Msg : Indica se a MTA pode ou não entregar mensagens ao UA.
 - Flg_Not : Indica se a MTA pode ou não entregar notificações ao UA.
 - Nome_OR: Indica o nome O/R do usuário (UA).
- Credenciais: Indica as credenciais de acesso do usuário (UA) ao MS.

- Restrições: Armazena as restrições correntes sobre a operação de "Fetch" do Message Store. Este campo se subdivide em sub-campos:

Cont.Types	EIT	Cont.Lenght
------------	-----	-------------

- Cont.Types: Indica os tipos de conteúdos que podem ser solicitados em um operação de "fetch".
 - EIT: Indica os tipos de informação codificada que podem ser solicitados em um operação de "fetch".
 - Cont.Lenght: Indica o comprimento máximo de conteúdo que pode ser solicitado em um operação de "fetch".
- Nome_Livre: Indica um nome amigável para identificação do usuário.

TAB_MTA (Tabela Relativa ao MTA) Esta tabela contém todas as informações pertinentes ao MS ao qual o UA está associado. As informações desta tabela refletem a configuração do MS e podem ser alteradas pela administração do sistema ou por operações remotas disparadas pelo UA. Esta tabela contém somente um registro cujos campos são descritos abaixo.

Nome_MTA	Senha_MTA	Subm_Ctrl
----------	-----------	-----------

Onde,

- Nome_MTA: Identificação do MTA ao qual o MS está associado. Utilizado para compatibilização com MTA 1984.
- Senha_MTA: Senha de acesso do MTA. Utilizado para compatibilização com MTA 1984.
- Subm_Ctrl: Contém os valores correntes de controle de submissão de mensagens/sondagens. Estes valores podem sofrer alterações durante um diálogo entre UA e MS e se subdivide em cinco sub-campos:

Tipo_Cod	Tam_Max	Prior_Min	Flg_Msg	Flg_Pr	Nome_OR
----------	---------	-----------	---------	--------	---------

- Tipo_Cod : Indica os tipos de codificação permitidos nas mensagens/sondagens que o UA submete.

- Tam_Max : Indica o comprimento máximo do conteúdo que o UA pode submeter.
- Prior_Min: Indica a prioridade mínima permitida para mensagens/sondagens que o UA submete.
- Flg_Msg : Indica se o UA pode ou não submeter mensagens.
- Flg_Prb : Indica se o UA pode ou não submeter sondagens.

(iii) Listas

LIST_INV (Lista Referente à Operações Remotas) Esta lista é utilizada para armazenar informação sobre quaisquer operações solicitadas pelo MS ao UA remoto e/ou ao MTA. É criado um ítem da lista para cada operação. Cada ítem possui os seguintes campos:

Id	Ação_MS	Wait_Result
----	---------	-------------

Onde,

- Id: Contém o identificador da operação submetida para transferência pelo ROSE. É idêntico ao parâmetro "invoke-id" da primitiva RO-INVOKE do ROSE.
- Ação_MS: Indica se a operação é fruto de ação explícita do UA remoto ou é fruto de uma ação automática do MS.
- Wait_Result: Indica se a operação em questão está aguardando um resultado.

LIST_WAIT Esta lista é utilizada para armazenar operações, resultados ou erros que aguardam que a comunicação com MTA esteja estabelecida. Cada ítem possui os seguintes campos:

Id	Ação_MS	Wait_Result	Invoke	W_Msg
----	---------	-------------	--------	-------

Onde,

- Id: Contem o identificador da operação submetida para transferência pelo ROSE. É idêntico ao parâmetro "invoke-id" da primitiva RO-INVOKE do ROSE.

- Ação_MS: Indica se a informação armazenada no ítem é fruto de ação explícita do UA remoto ou é fruto de uma ação automática do MS.
- Invoke: Indica se a informação armazenada no ítem é uma operação.
- W_Msg : Armazena a operação, resultado ou erro propriamente dito.

IV.4.2.3 As Interfaces do MS

Uma Entidade Message Store deve se comunicar-se ora com o UA remoto para prestar-lhe os serviços de acesso à base de informação ou serviços de submissão indireta ao MTA, ora com o MTA, através do qual as mensagens são submetidas e recebidas. A comunicação com o UA remoto utiliza os elementos de serviço comuns da camada de aplicação (CASE), os quais efetuam estabelecimento de conexão e transferência de dados. As duas interfaces através das quais o MS se comunica são: a interface com o módulo MTA e a interface com o módulo CASE.

Interface MS/CASE Assim como descrito no caso do UA, a interface entre MS e CASE engloba todas as primitivas de serviço que são oferecidas pelo ACSE e ROSE. Sua utilização foi descrita em III.3.3. As primitivas do ACSE são:

- A-ASSOCIATE
- A-RELEASE
- A-ABORT
- A-P-ABORT

E as primitivas do ROSE são:

- RO-INVOKE
- RO-RESULT
- RO-ERROR
- RO-REJECT-U
- RO-REJECT-P

Estas primitivas são definidas em III.3.5.

Interface MS/MTA Quanto à distribuição e localização física, o MS e o MTA podem estar situados no mesmo ambiente (co-residentes) ou situados em diferentes ambientes. Neste caso, as entidades se comunicam de acordo com o protocolo P3, que foi especificado na seção III.3.4. O protocolo P3 utiliza os elementos de serviço comuns da camada de aplicação para transportar as operações remotas definidas para o serviço do MTS.

Para esta proposta de implementação foi adotada a configuração em que os MSs e MTA são co-residentes e, neste caso não é necessário um protocolo de comunicação entre as entidades. Entretanto, deve ser mantida a preocupação de se construir uma interface bem definida entre os módulos. Na medida do possível, esta interface deve seguir a filosofia do protocolo que seria utilizado se as entidades estivessem em sistemas abertos distintos. Isto permite uma fácil adaptação no caso de alteração de parâmetros de projeto.

Esta proposta de migração parte de uma implementação de MHS em conformidade com os padrões X.400 1984, portanto nesta primeira fase de migração o MTA utilizado será aquele previamente construído de acordo os antigos padrões. A antiga configuração era de UAs co-residentes com o MTA, portanto já existe uma interface MTA/UA local, porém projetada segundo o serviço do MTS definido na recomendação X.411 1984. O Protocolo P3, e, portanto, o serviço do MTS sofreu alterações como descrito na seção IV.2.9.3.

Na recomendação X.411 1984, a interface MTA/UA era descrita através de primitivas de serviço [58]; nos novos padrões isto é feito através de operações remotas. A Tabela IV.1 mostra a relação entre as primitivas de serviço definidas para a interface entre o MTA e seu usuário nos padrões X.400 1984 e as operações remotas definidas nos novos padrões para este propósito. A descrição das primitivas e seus parâmetros se encontra na recomendação X.411 1984 [7].

Nesta fase da migração, a interface entre MS/MTA deve utilizar a definição destas operações observando-se as restrições que as tornam compatíveis com um MTA de acordo com a recomendação X.411 1984.

As mensagens trocadas na interface MS/MTA tem o formato abaixo:

id-ms	id-operação	operação
-------	-------------	----------

Na primeira fase da migração deve-se manter a preocupação de minimizar o impacto sobre o software do MTA existente. Neste caso específico, o MTA utilizado foi alterado basicamente em dois sentidos:

- para inicializar a comunicação com o usuário, caso este seja um MS.
- para gerar e processar um identificador para cada operação.

Os demais identificadores utilizados são:

– content-identifier (antigo ua-content-id (X.411 1984))

Primitiva de Serviço (X.411 1984)	Operação Remota (X.411 1988)
(UAL) LOGON (MTL) LOGON	MTS-BIND
LOGOFF	MTS-UNBIND
REGISTER	REGISTER
(UAL) CONTROL	DELIVERY-CONTROL
(MTL) CONTROL	SUBMISSION-CONTROL
(UAL) CHANGE-PASSWORD (MTL) CHANGE-PASSWORD	CHANGE-CREDENTIALS
SUBMIT	MESSAGE-SUBMISSION
DELIVER	MESSAGE-DELIVERY
PROBE	PROBE-SUBMISSION
NOTIFY	REPORT-DELIVERY
CANCEL	CANCEL

Tabela IV.1: Relação entre as Primitivas de Serviço (X.411 84) e as Operações Remotas (X.411 88)

- message-submission-identifier (antigo submit-event-id (X.411 1984))
- probe-submission-identifier (antigo probe-event-id (X.411 1984))

O identificador content-identifier consiste de uma cadeia de caracteres contendo a data local em segundos.

Os identificadores message-submission-identifier e probe-submission-identifier consistem de um inteiro contendo a data local em segundos.

O identificador subject-submission-identifier (antigo deliver-event-id) não é utilizado.

IV.4.3 Ambiente e Estrutura de Implementação

IV.4.3.1 Os Ambientes de Implementação

O ambiente de implementação adotado inicialmente para os UAs remotos é o microcomputador do tipo PC com o sistema operacional DOS, devido a sua disponibilidade e baixo custo. Posteriormente esta implementação será transportada para estações de trabalho com sistema operacional tipo Unix.

O ambiente de implementação adotado para os MSs constitui-se do sistema computacional VAX 8810 com o sistema operacional VMS. Este sistema é multi-usuário e incorpora mecanismos para comunicação e sincronismo entre processos

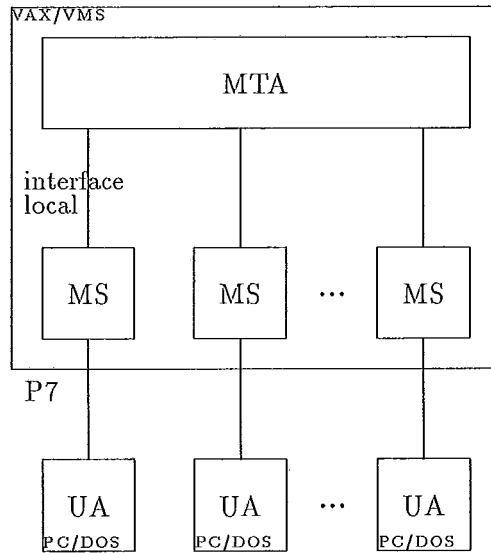


Figura IV.3: Estrutura da Implementação da Primeira Fase da Migração

utilizando recursos de memória virtual. Sua configuração atual inclui recursos de armazenamento em disco e um "spolling" para impressão em dispositivos de saída. A sua operação se dá através de terminais interativos.

A Figura IV.3 ilustra a estrutura da implementação da primeira fase da migração. Esta estrutura é descrita a seguir.

IV.4.3.2 A Estrutura de Implementação do UA no PC/DOS

Uma vez que o sistema PC/DOS não é do tipo multi-tarefas, a estrutura de implementação dos UAs remotos resultou em um só programa executável neste ambiente. O software consiste dos módulos que implementam o UA propriamente dito e a interface homem-máquina do sistema. A comunicação entre os diversos módulos ocorre através da troca de mensagens como parâmetros de funções destes módulos.

O procedimento básico do funcionamento do UA consiste de um ciclo de interações. Este ciclo analisa a comunicação com o módulo que recebe comandos do usuário, analisa o módulo que implementa os Elementos de Serviço Comuns da Camada de Aplicação (ACSE e ROSE), e ativa, quando necessário, os procedimentos da interface Homem-Máquina do UA ou do Protocolo P7, respectivamente.

As entidades UA implementadas em PC/DOS foram desenvolvidas utilizando-se a linguagem de programação C com o compilador Turbo C versão 2.2.

Implementação da Interface Homem-Máquina no Ambiente PC/DOS

A interface homem-máquina é o ponto de entrada do usuário no sistema. É a parte do Agente Usuário que interage diretamente com o usuário, apresentando o conjunto de recursos de manipulação de mensagens disponível, bem como facilidades locais para, por exemplo, edição, envio e seleção de mensagens.

É importante que esta interface seja amigável, que supra as necessidades da comunidade de usuários, e tanto quanto possível, tenha funcionamento intuitivo. De forma a atingir estes objetivos, a interface homem-máquina foi desenvolvida utilizando-se janelas e "menubar"s, de modo que com pouco esforço o usuário "navega" através das opções, lendo e enviando mensagens, notificações [59].

As opções contidas no menu principal da interface permitem que o usuário, dentre outras, componha mensagens através do editor de texto de sua preferência; leia, imprima e apague mensagens armazenadas em arquivos; cancele o envio de mensagens pré-datadas submetidas ao MHS; envie pedidos de sondagem; e formule e envie pedidos de operações específicas do Message Store.

Quanto à seleção de mensagens para leitura, na tela principal aparece uma listagem das mensagens armazenadas, contendo dados sobre cada mensagem, tais como, o originador, assunto, data de envio e recebimento, e uma indicação de que a mensagem foi lida ou não. No caso de mensagens confidenciais, nenhuma destas informações aparece, para que o usuário possa "abrir" a mensagem em momento apropriado.

A interface também procura facilitar a utilização de serviços opcionais para envio de mensagens. Como, por exemplo, em campos do cabeçalho da mensagem que devem ser preenchidos com referências a outras mensagens. Neste caso, é apresentada uma listagem das mensagens armazenadas, de modo que basta selecionar a mensagem a qual se deseja referenciar pressionando uma tecla.

Foi implementado um diretório de "apelidos" (aliases), de modo a facilitar o endereçamento do ponto de vista do usuário. Desta forma, o usuário não necessita usar a forma padrão de endereçamento do MHS (nomes O/R) para envio de mensagens. A interface permite que o usuário acrescente, atualize, procure ou apague destinatários cadastrados no diretório.

Uma das principais preocupações no desenvolvimento desta interface foi minimizar o impacto sobre o usuário em relação à interface desenvolvidas para os UAs co-residentes ao MTA no ambiente VAX/VMS, mesmo não dispondo das facilidades oferecidas pelo ambiente VMS.

IV.4.3.3 A Estrutura de Implementação do MS no VAX/VMS

A estrutura de implementação do MHS no VAX/VMS é composta de vários processos que se comunicam através de troca de mensagens. Cada entidade (MTA ou MS) é implementado como um processo independente.

As entidades do MHS implementadas no VAX/VMS foram desenvolvidas utilizando-se a linguagem de programação C com o compilador VAX C.

O procedimento básico do funcionamento do MS consiste de um ciclo de interrogações. Este ciclo analisa a comunicação com o módulo que implementa os Elementos de Serviço Comuns da Camada de Aplicação (ACSE e ROSE), analisa a comunicação com o MTA, e ativa, quando necessário, os procedimentos do Protocolo P7 ou da interface entre o MS e MTA, respectivamente. Cada análise realizada se refere a uma possível recepção de eventos provenientes do ACSE/ROSE ou do MTA.

Quanto á comunicação entre as entidades do MHS no VAX/VMS, existem dois tipos de mensagens: a mensagem de dados e a mensagem de sinalização.

As mensagens de sinalização são trocadas através de caixas postais ("Mailboxes"), isto é, filas FIFO do próprio Sistema Operacional. Um processo pode ter caixas postais de entrada (fila de mensagens recebidas) e caixas postais de saída (filas de mensagens enviadas). A cada caixa postal de saída está associada a identificação do processo receptor das mensagens transmitidas àquela caixa postal de saída.

Por questões de eficiência, preferiu-se realizar a transferência de dados através de áreas globais, isto é, áreas de memória comuns a todos os processos. A transferência de dados entre os processo é realizada através das caixas postais e das áreas globais, sendo a caixa postal utilizada apenas para sincronização, ou seja, somente o apontador para a área alocada para a mensagem é realmente passado pela caixa postal. Isto se deve ao fato de que utilizar o procedimento de comunicação de processos via caixas postais para mensagens de dados é um processo lento no ambiente em questão.

A comunicação entre entidades pode ser dividida em dois casos:

- a) Comunicação de entidades pertencentes a um único domínio de gerenciamento;
- b) Comunicação de entidades pertencentes a domínios de gerenciamento distintos.

O primeiro caso é mais simples e se resume basicamente na troca de apontadores. A mensagem sendo trocada está armazenada na área global e apenas

seu deslocamento dentro dessa área é colocado numa caixa postal de entrada do processo destino, ou seja, uma simples troca de apontadores.

O segundo caso é mais complexo pois envolve a transmissão/recepção de unidades de dados entre os domínios de gerenciamento distintos. Para tal, são implementados os protocolos padrões do MHS e são utilizados recursos de rede.

O MHS utiliza as camadas de sessão e transporte desenvolvidas pelo NCE/UFRJ [64,65], e para os níveis inferiores, o padrão X.25 oferecido pela RENPAC (Rede Nacional de Pacotes).

O procedimento básico da implementação do MS no VAX/VMS é mostrado abaixo, e os recursos disponíveis neste ambiente de implementação para a comunicação entre as entidades co-residentes do MHS são descritos no parágrafo seguinte.

```
main()
{
    ...

    ...

    mem_init    ();    /* inicializacao da global section */
    ini_mbx_usr ();    /* inicializacao das mbx do usr */
    ini_mbx_mta ();    /* inicializacao das mbx do mta */

    /* liga attentions dos mbx */
    set_mbx_attn(chan_frmta, from_mta, 0);
    set_mbx_attn(chan_frusr, from_usr, 0);

    /* Processa a fila */
    fim_ms = FALSE;
    while (!fim_ms)
    {
        qp = getqueue ();

        if (qp->q_from == msg_from_mta)
            trata_mta (&qp->q.msg);
        else
            trata_usr (&qp->q.umsg);
    }
}
```



```
/* maquina de estados do MS */
fim_ms = ms_provider (&ev);

free (qp);
}

fim ("");

...

...
}
```

Comunicação entre as Entidades do MHS Co-residentes no VAX/VMS

A comunicação entre as entidades do MHS co-residentes no VAX/VMS é realizada através dos recursos disponíveis neste ambiente de implementação. Os recursos descritos a seguir são definidos no manual de referência dos serviços do sistema VAX/VMS ("VAX/VMS System Services Reference Manual") [66] e no manual de programação em "C" para o VAX ("Programming in VAX C") [67].

(i) Estabelecimento de uma Caixa Postal (Mailbox):

Esta função é utilizada por todas as entidades do MHS co-residentes no VAX para que se estabeleça o mecanismo para sua comunicação.

As caixas postais são dispositivos virtuais disponíveis no VMS que funcionam como filas FIFO e fornecem mecanismos de sincronização para troca de dados entre processos. Uma vez criada, pode-se requisitar uma interrupção quando uma mensagem for escrita na caixa postal.

O estabelecimento de uma caixa postal no VAX/VMS é realizado através de uma chamada ao sistema. O serviço definido pelo sistema é:

SYS\$CREMBX

Nesta implementação as caixas postais são utilizadas somente para sincronização entre as entidades. Portanto, a caixa postal é utilizada para passar apenas a referência da área de memória compartilhada alocada para os dados que se deseja transferir. O processo receptor, então, sofre uma interrupção que normalmente desvia sua execução para uma rotina que tratará os dados recebidos.

(ii) Leitura de Dados de uma Caixa Postal com Espera Bloqueante:

Esta função é utilizada por todas as entidades do MHS co-residentes no VAX para a efetivação da operação de leitura bloqueante de dados de uma caixa-postal enviados por outro processo. O serviço que realiza esta operação é:

SYS\$QIOW

(iii) Envio de Dados à uma Caixa Postal:

Esta função é utilizada por todas as entidades do MHS co-residentes no VAX para a efetivação da operação de envio de dados de à uma caixa-postal. Esta operação não implica o bloqueio do processo emissor. O serviço que realiza esta operação é:

SYS\$QIO

(iv) Liberação de uma Caixa Postal:

Esta função é utilizada por todas as entidades do MHS co-residentes no VAX ao final de sua execução. O serviço que realiza esta operação é:

SYS\$DEASSIGN

(v) Criação e Mapeamento de uma Área Global (Global Section):

Esta função é utilizada para o estabelecimento de registros globais e áreas de memória comuns aos processos do MHS. O serviço que realiza esta operação é:

SYS\$CRMPSC

No MHS, as áreas de memória compartilhadas são manipuladas através de funções desenvolvidas especificamente para a alocação e desalocação dinâmicas. Desta forma, um processo pode alocar a área necessária para os dados, copiá-los para esta área global, e enviar a referência (deslocamento com relação ao início da área) para outro processo através de uma caixa postal.

Este mecanismo de comunicação entre processos provou ser eficiente, combinando o sincronismo da caixa postal com a rapidez de acesso da memória compartilhada.

(vi) Criação de Sub-processo:

Esta função é utilizada para a criação das entidades que compõem o MHS no VAX. O serviço que realiza esta operação é:

SYS\$CREPRC

IV.4.4 Comentários e Propostas

IV.4.4.1 Desenvolvimento do UA Remoto

A implementação desta versão inicial dos UAs remotos suporta todos os elementos de serviço do Protocolo P2, exceto os de indicação de corpo incompleto e indicação de idioma. A inclusão destes serviços impediria a interoperabilidade com UAs projetados segundo a recomendação X.420 1984.

Uma restrição imposta é que, por não haver suporte à conversão no MTS utilizado, se oferece os tipos de parte de corpo IA5Text e de mensagem redirecionada, entretanto esta é a configuração mais comum nos sistemas correntemente em operação.

O principal problema encontrado no desenvolvimento do UA no ambiente PC/DOS foi a restrição de tamanho de memória disponível. Geralmente, a implementação de entidades MHS requerem uma quantidade razoável de memória devido a grande quantidade de dados manipulados e a grande quantidade de operações oferecidas sobre estes dados.

Devido a restrições do próprio compilador, o número de atributos que podem ser selecionados para a operação Summarize do MS foi reduzida para cinco.

Nesta versão não foi implementado o mecanismo de segurança proposto pelos padrões. Este mecanismo poderá ser implantado tendo-se a configuração resultante da segunda fase da migração proposta na seção IV.3, já que estará disponível o Sistema de Diretório para armazenamento de informação e chaves necessárias a estes mecanismos.

Inicialmente, para comunicação com o MS é utilizado implementações dos serviços do ACSE e ROSE e um protocolo correspondente à camada de enlace do Modelo OSI. Não se considerou necessário a utilização das camadas intermediárias para a configuração inicial. Entretanto, estas serão necessárias caso se deseje ligar estes UAs ao MSs através de redes públicas de dados. Deve se levar em conta a restrição de tamanho de memória imposta pelo ambiente PC/DOS. Futuramente podem ser desenvolvidos estudos de forma a obter uma configuração OSI completa e eficiente neste ambiente.

IV.4.4.2 Desenvolvimento do MS

Com relação ao Message Store, a implementação desenvolvida suporta todas as operações propostas pela Recomendação X.413 [16], ou seja, as operações Summarize, List, Fetch, Delete, Register-MS, e também ações automáticas como Auto-

Alert e Auto-Forward descritas na seção IV.4.2.1.

A implementação contém restrições apenas no que tange a forma de selecionar entradas. No MS, as entradas podem ser selecionadas de duas formas principais: utilizando diretamente o(s) número(s) de seqüência, ou seletores que suportam expressões relacionando atributos que podem estar presentes nas entradas a serem selecionadas.

O seletor tem o seguinte formato:

```
Selector ::= SET {
    child-entries [0] BOOLEAN DEFAULT FALSE,
    range         [1] Range      OPTIONAL  - default is unbounded -
    filter        [2] Filter     OPTIONAL  - default is all entries within
                                           the specified range -,
    limit         [3] INTEGER (1 .. ub-messages) OPTIONAL,
    override      [4] OverrideRestrictions OPTIONAL  - default is that any
                                           - fetch-restrictions
                                           - in force do apply - }
```

Um parâmetro selector é usado para selecionar entradas de uma base de informação. A seleção opera em três estágios. Primeiramente, o conjunto total de entradas na base de informação pode ser restrita ao conjunto contíguo particular especificando-se a extensão (range). Em segundo lugar, entradas de dentro deste conjunto podem ser selecionadas especificando-se um filtro (filter) o qual a entrada selecionada deve satisfazer. Em terceiro lugar, um limite (limit) pode ser imposto sobre o número de entradas selecionadas; neste caso, as entradas selecionadas são aquelas com os menores números de seqüência. O parâmetro Child-entries indica se as entradas filho devem ser consideradas para seleção. O parâmetro Override permite sobrepor restrições anteriormente especificadas.

Um parâmetro "filter" aplica um teste a uma entrada em particular. O filtro contém expressões sobre a presença ou o valor de determinados atributos da entrada, e é satisfeito se e somente se o resultado é TRUE.

```
Filter ::= CHOICE {
    item [0] FilterItem,
    and  [1] SET (SIZE( 1 .. ub-nested-filters)) OF Filter,
    or   [2] SET (SIZE( 1 .. ub-nested-filters)) OF Filter,
    not  [3] Filter }
```

Um filter é um "filter-item", ou uma expressão envolvendo filtros que são compostos usando-se operadores lógicos "and", "or" ou "not".

Onde o filter é:

- a) um "item"; o resultado é verdadeiro se e somente se o filter-item correspondente é verdadeiro;
- b) um "and"; o resultado é verdadeiro a menos que algum filtro no SET seja falso. Se não existem filters no SET, o and resulta verdadeiro.
- c) um "or"; o resultado é falso a menos que algum filtro no SET seja verdadeiro. Se não existem filters no SET, o or resulta falso.
- d) um "not"; o resultado é verdadeiro se e somente se o filter é falso.

Um filter-item é uma avaliação da presença ou valor(s) de um atributo de um tipo particular na entrada sob teste. Cada tal avaliação resulta verdadeiro ou falso. O formato ASN.1 do FilterItem pode ser encontrado em [16].

A restrição na implementação do seletor é que somente a opção "item" do filtro (filter) foi suportada.

Quanto ao número de entradas suportadas pelo MS, esta quantidade depende do espaço em disco disponível a cada entidade no momento da instalação. Isto também depende do número de MSs que poderão ser criados no sistema já que cada MS está associado a somente um UA. Posteriormente, seria interessante que este fator fosse configurável pela administração do sistema.

O fato de um MS não poder ser compartilhado se constitui em um problema a nível de eficiência, já que para cada MS é criado um processo no sistema operacional, e as entradas referentes a uma mesma mensagem entregue a vários usuários não pode ser compartilhada. Isto causa um "overhead" considerável levando-se em consideração que um sistema do porte do MHS é projetado para atender a um grande número de usuários.

Um MS eficiente a nível de aproveitamento de espaço poderia ser construído explorando-se o esquema de referência a entradas via seus números de seqüência. Entradas podem ser logicamente duplicadas referenciando-as em outras entradas e mantendo contadores para estas referências de forma a controlar as múltiplas referências. Deste modo, um único MS poderia implementar múltiplas caixas postais lógicas, e compartilhar uma única cópia de mensagem entre todos os destinatários locais desta mensagem.

Em [56] é projetado um MS multi-usuário, que, apesar de não obedecer aos padrões, fornece grande funcionalidade, provendo até mesmo aplicações do tipo conferência distribuída através do MS. Soluções deste tipo devem ser levadas em consideração pelos órgãos de padronização.

Atualmente, as entradas são armazenadas no arquivo segundo uma estrutura fixa, acrescida de um prefixo que contém informações para facilitar o acesso. Pretende-se, portanto, alterar a forma de armazenamento, dividindo o arquivo em dois: um que constitui um "diretório" de entradas armazenadas, e outro com as entradas propriamente ditas. Tal alteração permite o tratamento de mensagens através de estruturas variáveis, eliminando o desperdício com campos opcionais, o que é frequente na sintaxe de representação de dados no padrão X400.

O desenvolvimento da primeira fase da migração resultou em cerca de 500 Kbytes de código executável no PC/DOS, compreendendo o UA e a interface homem-máquina, e 150 Kbytes de código executável no VAX/VMS, compreendendo o MS.

IV.5 Aspectos Relativos ao Desenvolvimento da Fase 2

Conforme visto na seção IV.3, a segunda fase da migração consiste basicamente da evolução do Agente de Transferência de Mensagens (MTA) com a implantação de um Sistema de Diretório X.500 (DS). Deve-se notar que o estudo e desenvolvimento do DS não faz parte do escopo deste trabalho, porém isto vem sendo realizado paralelamente.

O principal uso esperado do Diretório é a obtenção de endereços de usuários dado seu nome de diretório, conforme visto na seção II.3. Entretanto, o DS possui grande potencial de utilização em outros aspectos do MHS tais como maximização de conectividade, roteamento, expansão de listas de distribuição.

IV.5.1 Resolução de Endereços

Espera-se que eventualmente UAs contenham subsistemas DUAs interativos para permitir que o usuário acesse o DS a fim de identificar um destinatário potencial ao enviar mensagens. Além disso, espera-se que o UA ofereça ao usuário um diretório pessoal onde os endereços O/R de usuários mais frequentes possam ser associados a chaves mais amigáveis (nome, iniciais). O UA poderia realizar o mapeamento

entre endereços e chaves para o usuário na submissão/recepção de mensagens. Este tipo de uso do Diretório está fora de escopo de padronização.

Entretanto, os novos padrões definem um método de uso do Diretório. na submissão, uma mensagem pode conter Nomes O/R que contêm somente Nomes de Diretório. O MTA inicial usará o Diretório para obter os Endereços O/R correspondentes. A mensagem pode conter um indicação do método de entrega preferido do originador, de forma que quando somente o Nome de Diretório é dado, o MTA pode tentar obter a informação de endereçamento apropriada até encontrar o método de entrega que o destinatário aceita (ou o que ele especificou no DS).

Além do MTA inicial, o DS é utilizado somente se a entrega é considerada impossível para o Endereço O/R especificado e se o Nome O/R contém também um Nome de Diretório. Então, se o DS fornece um Endereço O/R diferente daquele inválido, este será utilizado para a entrega da mensagem.

IV.5.2 Maximização de Conectividade e Roteamento

Um importante aspecto de sistemas distribuídos é a sistematização de operações freqüentes. A conexão entre qualquer par de MTAs é realizada segundo acordos bilaterais. Se cada acordo requer intervenção humana, então provavelmente não haverão muitas conexões para cada MTA. Conforme o sistema cresce, isto pode levar à uma conectividade esparsa, e conseqüentemente, a uma estrutura relativamente cara para transferência de mensagens. Seria interessante automatizar o MTA tal que uma simples alteração que somasse conectividade a um MTA pudesse ser propagada aos demais MTAs que necessitassem desta informação. Isto permitiria ao MTS utilizar a conectividade potencial OSI. Esta automação pode ser alcançada através do uso do DS, e seria altamente desejável, principalmente dentro de Domínios de Gerenciamento que contenham muitos MTAs.

O DS pode armazenar informação que os MTAs recuperariam quando necessário. Os endereços O/R contidos no DS podem apontar para nomes de MTAs através dos quais estes endereços podem ser alcançados. Estes nomes podem ser acompanhados de informações adicionais sobre as características dos MTAs tais como capacidade de conversão, de expansão de listas de distribuição; estas informações são úteis na escolha de um MTA ótimo quando existem várias opções. Finalmente, os nomes de MTAs podem estar associados à informação sobre a conexão entre os MTAs.

IV.5.3 Listas de Distribuição

Quanto à expansão de Listas de Distribuição, de um modo geral a expansão pode ocorrer pelo UA originador, por um UA especial agindo como um "agente de grupo", ou pode ser expandida no MTS. Os novos padrões definem suporte à expansão no MTS e estudos mostram que este modo é o mais vantajoso [54].

Para expansão de listas de distribuição no MTS as seguintes informações podem ser obtidas do DS:

- Lista de destinatários;
- Controle de Acesso à Lista para:
 - alteração autorizada (atualização)
 - expansão autorizada (leitura)

É possível restringir o acesso à lista a alguns usuários, tanto para entrada na lista quanto para envio de mensagens à lista (e.g. aplicações comerciais).

- Seleção de originadores autorizados: Três políticas típicas podem existir:
 - Listas abertas, para as quais qualquer usuário pode enviar mensagens;
 - Listas fechadas, para as quais somente os membros da lista podem enviar mensagens;
 - Listas restritas, para as quais somente um conjunto explícito de usuários autorizados pode enviar mensagens.

A necessidade de listas fechadas ou restritas surge da questão do gerenciamento e contabilização de custos. Em alguns casos, a política de uma lista pode estabelecer que os membros ou o proprietário (responsável) da lista pagarão pela distribuição de mensagens; pode haver a possibilidade de determinar se um determinado originador que executa ações dispendiosas tem autorização para tal. Além disso, os membros da lista podem desejar ser protegidos contra mensagens espúrias.

No caso de listas aninhadas, deve ser implementada um tipo de autorização de acordo, e.g., "qualquer usuário autorizado a enviar mensagens à lista A é também autorizado a enviar mensagens à lista B se B é um dos membros de A.

IV.5.4 A Questão da Interoperabilidade

Apesar da proposta deste trabalho especificar os passos para a evolução de um MHS, deve-se ter em mente a preocupação de manter a capacidade de comunicação

com sistemas baseados em X.400 1984, já que o novo modelo não se encontra completamente disseminado. Os novos padrões colocam extensões ao Protocolo P1 para que os novos serviços de transferência de mensagens possam ser suportados.

Na verdade, o Protocolo P1 1988 perde qualidade de serviço quando se comunica com MTAs 1984. No caso em que a comunicação entre usuários em sistemas X.400 1988 necessita ser roteada através de sistemas 1984, as novos padrões indicam uma degradação do Protocolo P1 1988 para que este possa operar em modo 1984. Com isto o protocolo não estaria apto a prover os novos serviços de transferência de mensagens.

Esta questão é bastante controversa. Se se assume que a degradação não é aceitável, então uma solução do tipo "gateway" poderia reduzir alguns dos problemas de interoperabilidade. Se este último tipo de solução não é considerada aceitável, então uma solução alternativa deve ser encontrada juntamente com seu impacto sobre a qualidade de serviços tanto para usuários de sistemas X.400 1984 como para usuários de sistemas X.400 1988.

As regras a serem seguidas quando um MTA 1988 se comunica com um MTA 1984 se dividem em três áreas funcionais:

- Regras para o estabelecimento de associação;
- Regras para a transferência de MTS-APDU para um MTA 1984;
- Regras para a recepção de MTS-APDU de um MTA 1984.

A seguir são analisadas as regras para interoperabilidade entre os sistemas.

IV.5.4.1 Estabelecimento de Associação

As restrições que um MTA 1988 deve obedecer para estabelecer e terminar associações com MTAs 1984 se referem somente ao uso de elementos relacionados a contexto de segurança (security-context), já que sistemas X.400 1984 não são capazes de gerar nem processar tais elementos.

IV.5.4.2 Regras para Transferência para Sistemas X.400 1984

A Recomendação X.419 define regras para degradação de uma MTS-APDU conforme a Recomendação X.411 1988 para uma conforme X.411 1984. Se todas as regras são aplicadas e determina-se a degradação bem sucedida, então a MTS-APDU assim degradada é transferida, caso contrário, a ação tomada pelo MTA será

Elementos Críticos	Elementos Não-Críticos
Recipient Reassignment Prohibited	Originator Requested Alternate Recipient
DL Expansion Prohibited	Redirection History
Conversion With Loss Prohibited	DL EXpansion History
Latest Delivery Time	Physical Forwarding Address
Requested Delivery Method	Recipient Certificate
Physical Forwarding Prohibited	Proof of Delivery
Physical Forwarding Address Request	Originator and DL Expansion History
Physical Delivery Modes	Reporting DL Name
Registered Mail Type	Originating MTA Certificate
Recipient Number For Advice	Proof of Submission
Originator Return Address	
Physical Delivery Report Request	
Originator Certificate	
Message Origin Authentication Check	
Message Security Label	
Proof of Submission Request	
Proof of Delivery Request	
Probe Origin Authentication Check	
Reporting MTA Certificate	
Report Origin Authentication Check	

Tabela IV.2: Elementos de Extensão Críticos e Não Críticos

idêntica a que seria tomada se houvesse falha na transferência. As regras são agrupadas por tipo de elemento de protocolo.

Extensões As regras para o elemento "extensions" são as seguintes:

"Se algum elemento de extensão por-mensagem está presente, e nenhum campo "extension-field" contém o valor "crítico-para-transferência" ou "crítico-para-entrega", então estes elementos devem ser deletados.

Se algum elemento de extensão por-mensagem está presente, e algum campo "extension-field" contém o valor "crítico-para-transferência" ou "crítico-para-entrega", então a degradação não foi bem sucedida.

Estas regras devem ser aplicadas antes de qualquer outra regra para transferência de MTS-APDU."

A Tabela IV.2 sumariza os campos de extensão que podem ser marcados como "críticos" e os que não podem.

No caso dos elementos não-críticos, se estes elementos podem ser salvos, talvez encapsulados em alguma parte de corpo do Protocolo P2, então quando a mensagem chega ao MTA 1988, esta informação seria recuperada sem qualquer perda de qualidade de serviço. Isto deveria ser feito de forma transparente aos sistemas X.400 1984 intermediários.

Nomes O/R A regra para o elemento Nome O/R (O/R Names) é a seguinte:

”O Nome O/R deve ser degradado deletando-se o componente Nome de Diretório (Directory Name), caso esteja presente, e degradando-se o Endereço O/R (item abaixo).”

Um dos problemas mais significantes dos padrões X.400 1984 é a falta de endereçamento amigável. Isto foi reconhecido e a solução foi estender o Nome O/R para que este contenha opcionalmente um Nome de Diretório, e estender o Endereço O/R (antigo Nome O/R 1984). Com o tempo, o uso de Endereços O/R diminuiria em favor dos Nomes de Diretório. Além disso, o processo de expansão de Listas de Distribuição requerem o uso opcional do Diretório, e espera-se que este uso se torne obrigatório com o tempo. Com este crescente uso do Diretório, esta função de degradação causará um número crescente de casos de não entrega de mensagens.

Analogamente ao caso anterior, esta informação sujeita a degradação e deleção poderia ser encapsulada em alguma parte de corpo para transmissão através de sistemas X.400 1984, e então seria recuperada no sistema X.400 1988. No caso em que a mensagem deve ser entregue a um MTA 1984, este deve determinar se a mensagem pode ou não ser entregue ao UA destino.

Endereços O/R A degradação do Endereço O/R (O/R Address) é realizada como segue:

”Se o Endereço O/R contém qualquer atributo codificado com cadeias Teletex e cadeias printáveis, as cadeias Teletex devem ser deletadas.

Se o Endereço O/R é um endereço-OR-numérico ou um endereço-OR-de-terminal contendo um nome-de-domínio-privado, o Endereço O/R não pode ser degradado.

Se o Endereço O/R é um endereço-OR-telemático:

a) que contém um nome-de-país, um nome-de-domínio-administrativo, um endereço-de-rede, opcionalmente atributos-definidos-no-domínio, e nenhum outro componente, então o Endereço O/R não deve ser alterado;

b) que contém um endereço-de-rede, opcionalmente um identificador-de-terminal, e nenhum outro componente, então o Endereço O/R não deve ser alterado;

c) que contém combinações de atributos diferentes das descritas acima, todos os atributos exceto o endereço-de-rede e identificador-de-terminal, caso estejam presentes devem ser deletados.

Se o Endereço O/R contém qualquer atributo codificado como cadeias Teletex e as correspondentes cadeias printáveis estão ausentes, o Endereço O/R não pode ser degradado.

Se após a aplicação de todas as regras acima o Endereço O/R ainda contiver qualquer atributo-de-extensão, o Endereço O/R não pode ser degradado.”

A maioria dos argumentos acima utilizados contra a degradação do Nome O/R podem ser aplicados ao Endereço O/R. A aplicação da última regra causará a não entrega de qualquer mensagem que possua atributos de endereçamento a sistemas de entrega física (PDS).

Outros Elementos de Protocolo As demais regras de degradação estão relacionadas à deleção de elementos antes da transferência à MTAs 1984, ou à alteração de Nomes O/R. Os elementos de protocolo que se enquadram nesta categoria são: informação-bilateral-por-domínio, informação-de-trace, informação-de-trace-intermediária, nome-de-originador, nome-destinatário-de-relatório, campos-por-destinatário, pertencentes aos envelopes de transferência de mensagens, sondagens e relatórios. Estas regras podem ser encontradas no Anexo B da Recomendação X.419.

Além das regras descritas acima, a nível da notação ASN.1, foram adicionados novos tipos aos tipos básicos definidos na Recomendação X.409 1984. Deve-se observar que sistemas baseados em X.400 1984 podem não estar aptos a manipular estes novos tipos. Portanto, os dados enviados a sistemas X.400 1984 devem ser restringidos aos tipos definidos na Recomendação X.409 1984.

IV.5.4.3 Regras para Recepção de Sistemas X.400 1984

Os novos padrões estabelecem que tão logo os sistemas X.400 1984 utilizem as restrições de tamanho que foram definidas para vários elementos do protocolo P1 (1984), nenhuma atitude especial precisara ser tomada pelos sistemas X.400 1988 na recepção destes elementos de protocolo.

IV.6 Comentários

Este capítulo descreveu e discutiu aspectos importantes dos padrões X.400 que antes eram insuficientemente tratados pela recomendações, apresentando soluções

que podem ser tomadas a nível de implementação. Com isto são levantadas as vantagens que a migração de um MHS-84 para um MHS-88 pode trazer.

Entretanto, a estratégia escolhida para a migração deve ser cuidadosamente estudada para que se isole o efeito de conseqüências inesperadas e para que se minimize o impacto sobre os usuários do sistema. No caso particular do MHS, este planejamento é de grande importância já que se deve levar em consideração os possíveis problemas de interoperabilidade; e para que, em primeira instância, se implante, na medida do possível, novos serviços, protocolos e extensões dos novos padrões sem perder a compatibilidade com sistemas X.400 1984 quanto à transferência de mensagens.

A Recomendação X.419 define a degradação de vários elementos do Protocolo P1 sempre que um sistema X.400 1988 se comunicar com um sistema X.400 1984. Isto significa que sempre que isto ocorrer, a qualidade de serviço dos usuários finais será reduzida àquela dos elementos dos protocolos suportados pelos sistemas X.400 1984.

Estas regras de degradação podem encorajar a proliferação de sistemas X.400 ao invés de incentivar a existência gradual de implementações X.400 1988. Este problema se agrava se considerarmos que a maioria dos ADMDs estarão oferecendo X.400 1984 ainda por algum tempo. Acredita-se que este seja, atualmente, o maior obstáculo à uma maior aceitação dos sistemas X.400 1988.

Com a introdução de um "gateway MHS" que encapsulasse os elementos incompatíveis, implementações X.400 1984 e 1988 poderiam co-existir com pequena perda de qualidade de serviço para os usuário de X.400 1988. Além disso, implementações X.400 1984 estariam aptas a gradualmente suportarem alguns dos serviços X.400 1988 conforme necessário. Como a evolução para os novos padrões é altamente desejável, uma solução do tipo gateway para os problemas de interoperabilidade pode ser uma solução viável.

Capítulo V

Conclusões

V.1 Introdução

Este capítulo apresenta os comentários finais e conclusões obtidas durante o desenvolvimento desta tese, apresentado um direcionamento para a evolução deste trabalho.

Na seção V.2 são apresentados os resultados alcançados neste trabalho.

Na seção V.3 é apresentado um direcionamento para a validação da proposta de implementação.

Na seção V.4 são abordados aspectos acerca dos testes de conformidade propostos pelo CCITT para implementações de MHS 1988, e sobre a avaliação do desempenho desta implementação.

Na seção V.5 são discutidas a evolução do tema deste trabalho e as futuras tendências dos serviços de Manipulação de Mensagens.

V.2 Resultados

O trabalho desenvolvido nesta tese apresenta três resultados principais: a análise dos padrões X.400, a especificação formal do MHS e a estratégia de migração de um MHS-84 para um MHS-88.

A partir de uma análise dos elementos funcionais e de protocolos do MHS pôde-se levantar aspectos problemáticos dos padrões X.400, o impacto destes aspectos no sistema resultante, e com isto, definir soluções para tais questões a nível de implementação. Uma análise deste tipo é relevante, dada a importância atual dos Sistemas Eletrônicos de Mensagens e o esforço internacional de migração para a estrutura de comunicação definida pelo Modelo OSI.

Com os padrões X.400 1988 surgem extensões aos serviços e protocolos de MHS, as quais permitem considerável aumento da capacidade funcional e de distribuição do sistema. Entretanto, surge também a possibilidade de problemas de interoperabilidade com sistemas X.400 baseados nos padrões de 1984 devido a elementos de protocolo que tais sistemas não são capazes de manipular. Este trabalho levanta esta questão, analisando os novos elementos de protocolo, os problemas de interoperabilidade que podem surgir e apresenta um direcionamento para uma possível solução para este problema.

A questão da interoperabilidade não é trivial. Isto requer não somente a implementação correta das características dos protocolos incluídas nas especificações, mas exige também escolhas compatíveis principalmente em relação àqueles aspectos chamados "questões locais".

Portanto, a correta implementação de sistemas OSI exige um investimento substancial no entendimento dos padrões e na avaliação das implicações de diferentes escolhas de projeto. Neste contexto, a especificação dos protocolos são de particular importância, já que representam os padrões que são a base para a implementação e testes de sistemas abertos compatíveis e cooperantes. O uso de especificações formais e precisas aliadas a ferramentas que facilitem a geração automática de código para o sistema não somente acelerariam o processo de implementação como também garantiriam que tal implementação está em conformidade com sua especificação.

A modelagem do MHS descrita neste trabalho mostra o esforço de se descrever formalmente as entidades e protocolos de um sistema distribuído da complexidade e porte do MHS. Esta modelagem partiu de um modelo abstrato que não permite a visualização das fronteiras entre os objetos, não especifica o comportamento das entidades, e não mostra o mapeamento do modelo nos serviços de suporte. Estes, entre outros aspectos importantes dos protocolos e dos sistema, são originariamente descritos em linguagem natural.

A importância desta modelagem está na definição precisa do sistema em suas entidades e protocolos, e da utilização dos serviços que servem de suporte ao MHS. A especificação Estelle força a definição das interfaces entre módulos, através da especificação de canais e pontos de interação; além de forçar a especificação precisa dos predicados que devem ser satisfeitos de forma a ativar transições.

Algumas características importantes desta modelagem são a descrição do comportamento das entidades, o que é facilitado pelo fato de Estelle ser uma linguagem algorítmica; a utilização dos elementos de configuração dinâmica de Estelle, permitindo a visualização da operação distribuída dos MTAs, e as possíveis

configurações dos Elementos de Serviço Comuns da Camada de Aplicação; e a definição dos módulos de acesso, que conferiram maior modularidade e legibilidade à modelagem.

A principal dificuldade apresentada no desenvolvimento da modelagem do MHS em Estelle se refere à especificação dos dados dos protocolos, já que as Recomendações X.400 utilizam extensivamente a notação ASN.1, e traduzir esta declaração de dados para Pascal na especificação Estelle seria uma solução ineficiente. Para superar este problema, este trabalho propõe uma extensão ASN.1 para a linguagem Estelle. Com esta extensão, a linguagem Estelle poderia suportar a declaração e manipulação de dados em ASN.1, tornando as especificações ainda mais eficientes e completas, além de permitir a geração automática de código completa para os protocolos da Camada de Aplicação.

Outro resultado importante deste trabalho foi a definição de uma estratégia de migração de um sistema X.400 1984 para um sistema X.400 1988.

O processo de migração proposto parte de uma implementação de MHS 1984 onde as entidades de acesso (UA) e de transferência (MTA) são co-residentes, e esta é a configuração predominante em sistemas X.400 1984 em operação. Portanto, esta é uma estratégia que pode ser aplicada a várias implementações.

Com esta migração obtém-se uma configuração distribuída onde as entidades de acesso (UA) e de transferência (MTA) se encontram em sistemas abertos distintos e as entidades de acesso dispõem de entidades de armazenamento (MS) remotas. Com isto, as entidades de acesso podem ser implementadas em equipamentos de pequeno porte, os quais apresentam grande disponibilidade e baixo custo.

Segundo esta estratégia, em uma primeira instância, enquanto os padrões X.400 1988 não estão disseminados e implantados, a entidade de transferência (MTA) se comunicaria com outras segundo o Protocolo P1 1984, não obstante, poderia oferecer localmente alguns dos novos serviços advindo com os padrões X.400 1988. Isto permite a utilização local da funcionalidade estendida dos padrões X.400 1988 e da capacidade de interoperabilidade dos sistemas X.400 1984 implantados.

Finalmente, tem-se também como resultado deste trabalho o desenvolvimento da primeira fase do processo de migração proposto, com UAs implementados no ambiente PC/DOS remotos a MS implementados no ambiente VAX/VMS, além da discussão e direcionamento para o desenvolvimento da segunda fase da migração.

V.3 Validação da Proposta de Implementação

A validação da implementação proposta neste trabalho poderia ser encaminhada em duas partes distintas. A primeira parte compreende a elaboração de ferramentas para suporte ao processo de validação. A segunda parte engloba uma avaliação experimental do comportamento dos procedimentos realizada de forma exaustiva. Nesta segunda fase é observada a relação entre os procedimentos propostos para a implementação e os procedimentos implementados efetivamente.

O primeiro elemento de suporte para o processo de implementação consiste na implantação de uma nova tarefa no sistema cuja principal atividade é armazenar todas as mensagens trocadas entre duas ou mais entidades funcionais do MHS permitindo assim a obtenção de um "arquivo de rastreamento" que contém todas as unidades de dados transmitidas ou recebidas para/de um outro sistema aberto ou entidade. Esse arquivo de rastreamento é dito "horizontal".

O segundo elemento de suporte também consiste na implementação de uma nova tarefa no sistema cujo principal atividade é armazenar todos os eventos ocorridos no âmbito de uma submissão/recepção de mensagem permitindo a obtenção de um arquivo de rastreamento que contém todas as informações do percurso da mensagem dentro de um único sistema aberto ou entidade. Esse arquivo de rastreamento é dito "vertical".

O terceiro elemento consiste de um simulador da camada de sessão, que permite a instalação e interligação de duas instâncias da implementação do MHS em um único ambiente de implementação.

De posse das ferramentas descritas acima, é iniciado o processo de validação experimental da implementação, de uma forma gradual, permitindo que a complexidade embutida nesta fase seja diluída em etapas sequenciadas. Antes de atingir este estágio de validação, cada um dos módulos que compõem o modelo funcional da implementação devem ter sido testados separadamente, usando os elementos de suporte apresentados acima. Este processo de validação pode ser dividido em quatro etapas:

Em uma primeira etapa é analisado o comportamento de um domínio de gerenciamento simples formado por um único MTA e um pequeno número de UAs e MSs para que possam ser verificados os aspectos abaixo sem se levar em consideração os aspectos de transferência de mensagens:

- As operações de submissão e entrega, verificando a interface UA/CASE, MS/CASE e MS/MTA;

- A geração de entradas no MS;
- A realização de ações automáticas no UA e MS;
- Validação das operações de acesso ao MS, e conseqüentemente, o protocolo P7.

Em uma segunda etapa realiza-se a análise de dois domínios de gerenciamento simples, ou seja, um sistema obtido pela duplicação do sistema utilizado na primeira etapa descrita acima.

A análise é feita através de mecanismos de simulação. Isto permite que o monitoramento se concentre em um único ambiente de implementação. Desta forma, são criadas duas instâncias de implementação do MHS, interligados por um simulador da camada de sessão. Por meio dos elementos supervisores das suas instâncias, pode ser analisado o comportamento do sistema.

Neste ambiente é possível a simulação de falhas ocasionais, permitindo a observação do comportamento da implementação nestas condições.

Em uma terceira etapa transpõem-se a configuração estabelecida na etapa anterior para dois ambientes distintos, os quais são interligados por serviços de rede. A observação do comportamento do sistema localizado remotamente também é realizada através dos elementos supervisores utilizados na segunda fase.

Em uma última etapa, o número de domínios de gerenciamento é expandido para vários ambientes distintos, permitindo a avaliação do comportamento da implementação quanto aos aspectos relacionados ao tratamento simultâneo de diversas mensagens, as quais são transferidas através de unidades de dados de protocolo. E, então esta implementação de MHS é conectada a outro sistema X.400 já em operação, com isto se valida de forma completa os protocolos P2 e P1.

V.4 Avaliação da Implementação e Testes de Conformidade

Existem dois enfoques distintos na avaliação da implementação. O primeiro é extrair a eficiência da implementação e o segundo é avaliar a influência exercida por esta no ambiente de implementação [53]. Inicialmente é necessário estabelecer as condições pelas quais se dará a medida de desempenho da implementação.

Esta medida não deve incluir os tempos de quaisquer outros serviços que não sejam provenientes de uma entidade do MHS. Isto exclui as operações de transferência de mensagens realizadas através de rede.

É necessário estabelecer um conjunto de atividades padrão para a realização de comparações dos resultados, sobre diferentes situações.

Para o ambiente VAX/VMS, que é multi-usuário, há a necessidade de se estabelecer mecanismos para a ponderação da medida, mediante a carga de utilização instantânea do sistema. Uma forma prática para contornar esta situação é estabelecer sempre um número médio de medidas em instantes distintos, visando a obtenção de uma carga média do sistema.

Para avaliar a eficiência da implementação pode-se adicionar aos "arquivo de rastreamento" definidos na seção anterior, a informação do instante da ocorrência da cada troca de mensagens. Este mecanismo, conhecido como "timestamp", ocasiona uma identificação da seqüência dos tempos envolvidos nas várias fase de tratamento de uma mensagem. A partir desta informação, pode-se estabelecer os pontos críticos da implementação.

Durante o proceso de avaliação da eficiência da implementação deve-se aumentar gradativamente o número de mensagens tratadas, tanto no UA como no MTA, de entradas geradas no MS, e variar os tipos de serviços solicitados por estas mensagens, além de aumentar o acesso do UA ao MS. Isto permite avaliar a degradação desempenho da implementação em função do aumento de serviço efetuado, além de verificar os tempos de submissão e entrega de mensagens.

No caso do VAX/VMS, quanto ao aspecto da influência da implementação no ambiente, um ponto crítico a ser observado é o número de processos existentes no sistema (um para cada MS). O fato de um MS não poder ser compartilhado se constitui em um problema a nível de eficiência, já que para cada MS é criado um processo no sistema operacional, e as entradas referentes a uma mesma mensagem entregue a vários usuários não pode ser compartilhada. Isto causa um "overhead" considerável levando-se em conta que um sistema do porte do MHS é projetado para atender a um grande número de usuários. Para avaliar esta situação, deve ser verificada a carga média do sistema antes da implantação efetiva dos MSs.

Quanto aos testes de conformidade, podem ser utilizado os métodos, notações e critérios definidos na Recomendação X.403 [12].

V.5 Continuidade do Trabalho e Futuras Tendências

A importância deste trabalho reside não somente nos resultados discutidos na seção V.2, mas também em seu potencial de continuidade.

Inicialmente, este trabalho pode ser continuado no sentido de se realizar gradualmente a segunda fase do processo de migração proposto sem se perder de vista a questão da interoperabilidade com sistemas X.400 1984. Podem ser estudados os aspectos relativos à implementação de uma solução do tipo "gateway" descrita em IV.5.4.

No momento em que a implementação atingir um grau de maturidade razoável, deve-se estimular e facilitar a sua transposição para outros ambientes de implementação.

Além disso, esta implementação pode servir de base para inúmeros desenvolvimentos e estudos tais como:

- Desenvolvimento de aplicações baseadas no Diretório X.500 para gerenciamento de funções do MHS, como roteamento, maximização de conectividade, resolução de endereços e funções de segurança.
- Transferência de Arquivos utilizando o MHS. A idéia é prover uma base de transferência "store-and-forward" para transferência de arquivos, e que é diferente da transferência de arquivos através da aplicação FTAM (File Transfer and Manipulation), a qual é baseada em conexões diretas entre as máquinas. A intenção é prover transferência de arquivos compatível com FTAM e ODA (Office Document Architecture), provendo atributos similares a estas aplicações nos arquivos transferidos. Isto é pode ser conseguido no MHS utilizando um novo tipo de conteúdo para este propósito. Porém, a criação de um novo tipo de parte de corpo no Protocolo P2 seria também útil, já que combinaria a transferência de arquivos com a comunicação interpessoal.
- Integração de serviços facsímile ao MHS. Tal serviço é útil principalmente quando o originador deseja enviar um fax para vários destinatários em outros países, ou quando um usuário do MHS deseja enviar uma mensagem para destinatários que possuem somente equipamentos facsímile.

Seriam necessárias algumas adaptações aos padrões X.400 para a manipulação de facsímile quanto a notificações de entrega e não-entrega. Como não existem mecanismos de extensão para notificações, estas soluções podem ser estudadas, e até mesmo enviadas aos órgãos de padronização. A idéia de um mecanismo de extensão é que quando existir tal mecanismo, sistemas que não reconhecem certos elementos de protocolo não os rejeitem, caso os recebam.

- Interconexão global de serviços públicos de caixas postais e acesso a serviços telemáticos, como Telex, e a Sistemas de Entrega Física, através da Unidades

de Acesso.

- Desenvolvimento de suporte no MHS para comunicação de grupo.
- Utilização do Protocolo P7 sobre "links" móveis. Um sistema de mensagens "store-and-forward" pode ser um importante componente em ambientes de comunicação interpessoal baseados em links móveis ("mobile office environment"). Neste contexto, dois problemas surgem: capacidade de comunicação limitada e falta de segurança no meio de comunicação. Estes problemas são inerentes à telefonia celular sobre os quais projetos deste tipo são baseados. A utilização do Message Store definido pela Recomendação X.413 [16] e as extensões dos padrões X.400 para segurança podem superar estes problemas.

As tendências futuras para os Sistemas de Manipulação de Mensagens se concentram na evolução dos padrões X.400. Os vários aspectos inadequadamente ou insuficientemente definidos nos padrões X.400 1984 representaram tarefas não triviais para os primeiros implementadores de MHS. Os novos padrões X.400 tratam alguns destes aspectos, somando precisão à definição dos serviços; enquanto outros aspectos continuam a cargo das equipes de implementação ("questões locais").

Certamente surgirão soluções comuns para estas questões, e espera-se que estas sejam consideradas em futuros trabalhos de padronização. De qualquer forma, a existência de várias implementações X.400 demonstram a viabilidade destes sistemas. Futuras implementações devem ser flexíveis o bastante para se adaptarem aos novos desenvolvimentos nesta área. Mantendo-se uma interface bem definida entre os dispositivos existentes e novas tecnologias pode-se minimizar o impacto de possíveis migrações.

Bibliografia

- [1] ISO. Information Processing Systems - Open Systems Interconnection - Basic Reference Model. 1986. (ISO 7498).
- [2] CCITT Recommendations X.400 Series Message Handling Systems - 1984: System Model and Service Elements.
- [3] CCITT Recommendations X.400 Series Message Handling Systems - 1984: Basic Service Elements and Optional User Facilities.
- [4] CCITT Recommendations X.400 Series Message Handling Systems - 1984: Encoded Information Types Conversion Rules.
- [5] CCITT Recommendations X.400 Series Message Handling Systems - 1984: Presentation Transfer and Notation.
- [6] CCITT Recommendations X.400 Series Message Handling Systems - 1984: Remote Operations and Reliable Transfer Server.
- [7] CCITT Recommendations X.400 Series Message Handling Systems - 1984: Message Transfer Layer.
- [8] CCITT Recommendations X.400 Series Message Handling Systems - 1984: Interpersonal Messaging User Agent Layer.
- [9] CCITT Recommendations X.400 Series Message Handling Systems - 1984: Access Protocol for Teletex Terminals.
- [10] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Message Handling System and Service Overview.
- [11] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Overall Architecture.
- [12] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Conformance Testing.

- [13] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Abstract Service Definition Conventions.
- [14] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Encoded Information Types Conversion Rules.
- [15] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Message Transfer System: Abstract Service Definition and Procedures.
- [16] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Message Store: Abstract-service Definition.
- [17] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Protocol Specifications.
- [18] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Interpersonal Messaging System.
- [19] CCITT Recommendations X.400 Series Message Handling Systems - 1988: Access Protocol for Teletex Terminals.
- [20] ISO 10021-1 - 10021-7, Information Processing Systems - Text Communication - Message Oriented Text Interchange System.
- [21] CCITT Recommendations X.500 Series: The Directory, 1988, Blue Book, vol VIII Fascicle VIII.8.
- [22] CCITT. Data Communications Networks Interfaces. Recommendations X.20 - X.32, vol VIII, fascicle VIII.3 Geneva, 1985.
- [23] CCITT X.217 Recommendation: Association Control Service Definition for CCITT Applications.
- [24] CCITT X.218 Recommendation: Reliable Transfer: Model and Service Definition.
- [25] CCITT X.219 Recommendation: Remote Operations: Model, Notation and Service Definition.
- [26] CCITT X.227 Recommendation: Association Control Protocol Specification for CCITT Applications.
- [27] CCITT X.228 Recommendation: Reliable Transfer: Protocol Specification.

- [28] CCITT X.229 Recommendation: Remote Operations: Protocol Specification.
- [29] ISO/IEC 8649: 1988, Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element.
- [30] ISO/IEC 9072-1: 1989, Information Processing Systems - Text Communication - Remote Operations - Part 1: Model, Notation and Service Definition.
- [31] ISO/IEC 9066-1: 1989, Information Processing Systems - Text Communication - Reliable Transfer - Part 1: Model and Service Definition.
- [32] ISO/IEC 8649: 1988, Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element.
- [33] ISO/IEC 9072-1: 1989, Information Processing Systems - Text Communication - Remote Operations - Part 2: Protocol Specification.
- [34] ISO/IEC 9066-1: 1989, Information Processing Systems - Text Communication - Reliable Transfer - Part 2: Protocol Specification.
- [35] CCITT X.208 Recommendation: Specification Abstract Syntax Notation One (ASN.1).
- [36] CCITT X.209 Recommendation: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).
- [37] ISO/TC97/SC21/WG1-FDT/B, IS9074: Estelle, a Formal Description Technique Based on an Extended State Transition Model"
- [38] CCITT X400-Series Implementor's Guide - version 6.
- [39] LININGTON, P. F., "Why OSI?", *Computer Networks and ISDN Systems*, 17 (1989) n° 4 e 5 (287-290).
- [40] GREENE, N., e TISSOT, R.; "A Distributed User Agent: Concept and Implementation", *Proceedings of IFIP WG 6.5 Working Conference on Message Handling Systems and Distributed Applications*, Munich, April 1987.
- [41] SVOBODOVA, L., "Implementing OSI Systems", *IEEE Journal on Selected Areas in Communications*, vol 7, n° 7, September 1989.
- [42] SCHICKER, P., "Message Handling Systems X.400", *Proceedings of IFIP WG 6.5 Conference on Message Handling Systems and Distributed Applications*, California, 1988.

- [43] BOCHMANN, G. v., "Protocol Specification for OSI", *Computer Networks and ISDN Systems*, 18 (1989/90) (167-184).
- [44] SOUZA, W. L., "Estelle: Uma Técnica para a Descrição Formal de Serviços e Protocolos de Comunicação", *Revista Brasileira de Computação*, vol 5 n° 1 (33-49) jul/set 1989.
- [45] MACHADO, R. C., "Um Sistema de Suporte ASN.1", *Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos*, Setembro, 1989, Florianópolis-SC, Brasil.
- [46] SIJELMASSI, R., GAUDETTE, P., "An Object-Oriented Model for Estelle", *Proceedings of the First International Conference on Formal Description Techniques*, Stirling, Scotland, September, 1988.
- [47] COURTIAT, J. P., DEMBINSKI, P., GROZ. R., JARD C., "Estelle: Un Language ISO pour les Algorithmes Distribués et les Protocoles", *INRIA Rapports de Recherche n° 595*, Décembre 1986.
- [48] PEDROZA, A. C., GOULART, C. C., VALIM, P. R., OLIVEIRA, R. C., "Um Sistema de Auxílio ao Projeto de Protocolos de Comunicação para Redes de Computadores", *Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos*, Setembro, 1989, Florianópolis-SC, Brasil.
- [49] WHITE, J. E., "ASN.1 and ROS: The Impact of X.400 on OSI", *IEEE Journal on Selected Areas in Communications*, vol 7, n° 7, September 1989.
- [50] TAROUCO, L., *Redes de Computadores: Locais e de Longa Distância*, McGraw-Hill, São Paulo, 1986.
- [51] LEBECK, S. K., "Implementing MHS: 1984 versus 1988", *Proceedings of IFIP WG 6.5 Conference on Message Handling Systems and Distributed Applications*, California, 1988.
- [52] COURTIAT, J. P., DIAZ, M., AYADI, M. A. D., "Description Formelle de Protocolos OSI en Estelle", *Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos*, Setembro, 1989, Florianópolis-SC, Brasil.
- [53] CARMO, L. F. R. C., "Ambiente OSI de Transferência e Manipulação de Tarefas para um Sistema VAX/VMS", XII, 244 p., Tese - Universidade Federal do Rio de Janeiro, COPPE, M.Sc., Engenharia de Sistemas, 1988.

- [54] HUITEMA, C., e SARAS, J. A.; "The Use of Distribution Lists in MHS", *Proceedings of IFIP WG 6.5 Working Conference on Message Handling Systems and Distributed Applications*, Munich, April 1987.
- [55] MENDES, S., "Especificação Formal do Sistema de Manipulação de Mensagens (X.400-MHS) utilizando a Linguagem Estelle", Relatório Técnico NCE a ser publicado.
- [56] PAYS, P. A., e YOU, Y. Z., "A General Multi-User Message Store", *Proceedings of IFIP WG 6.5 Working Conference on Message Handling Systems and Distributed Applications*, Costa Mesa, California, USA, October, 1988.
- [57] PIRMEZ, L. e MENDES, S., "Uma Proposta de Implementação para o Agente de Transferência de Mensagens no Contexto do MHS", Relatório Técnico NCE 04/90.
- [58] MENDES, S. and PIRMEZ, L., "Uma Proposta de Implementação para o Agente Usuário no Contexto do MHS", Relatório Técnico NCE 18/90.
- [59] MOURA, P. M., e MENDES, S., "Implementação de uma Interface Homem-Máquina para o MHS no ambiente PC/DOS", Relatório Técnico NCE a ser publicado.
- [60] KEHRES, T., "Message Handling Systems: Interoperability Problems Between 1984 and 1988 X.400", *Proceedings of IFIP WG 6.5 Conference on Message Handling Systems and Distributed Applications*, California, 1988.
- [61] MENDES, S., PIRMEZ, L., "Desenvolvimento de um Sistema de Manipulação de Mensagens", *III Semana de Informática da Bahia (SEMINFO-UFBA)*, Salvador, BA, Abril de 1990.
- [62] MENDES, S., PIRMEZ, L., "Experiences on the Implementation of a MHS-84 and its Migration to a MHS-88", Artigo aceito para publicação e apresentação da 10nd International Conference on Computer Communication (ICCC'90), New Delhi - Índia, Novembro de 1990.
- [63] COCHRANE, D.; "Easing the Migration into the Future of Communications Technology", *Computer Networks and ISDN Systems 14 (1987)* pag. 323-329.
- [64] PIRMEZ, L., "Estudos dos Serviços de Sessão", Relatório Técnico NCE 07/87, Maio de 1987.

- [65] PIRMEZ, L., "Estudos dos Serviços de Transporte" Relatório Técnico NCE 04/89.
- [66] VAX/VMS System Services Reference Manual, Digital Equipment Corporation, nº AA-Z501B-TE, April 1986.
- [67] Programming in VAX C, Digital Equipment Corporation, nº AA-L370B-TE, April 1985.