★ RELATÓRIO TÉCNICO ★

PADRÕES X500: ESPECIFICAÇÃO E IMPLEMENTAÇÃO DE UM DIRETÓRIO DISTRIBUÍDO PARA O INTER-X

Luci Pirmez
Marcelo Ciminelli
Patrícia Moura

NCE 25/91
Novembro/91

Universidade Federal do Rio de Janeiro
Núcleo de Computação Eletrônica
Caixa Postal 2324
20001 - Rio de Janeiro - RJ
BRASIL

NCE UFRJ
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

Padrões X500: Especificação e Implementação de um Diretório Distribuído para o INTER-X

Luci Pirmez
Marcelo Ciminelli
Patrícia Moura

20 novembro 1991

Abstract

This report presents a proposal from the NCE for the specification and implementation of the Directory System based on the X500 series, for the project Inter-X. It presents a vision of the Directory, the utilization of the underlying services and a detailed description of the functioning of the Directory and its table states. From the point of view of the implementation, its structure of data and interface are described. Finally, the environment and the structure of implementation are presented.

Resumo

O objetivo deste relatório é apresentar a especificação e uma proposta de implementação de um Sistema de Diretório baseada na série X500, para o projeto Inter-X. É apresentada uma visão geral do Diretório, a utilização dos serviços subjacentes, bem como uma descrição mais detalhada do funcionamento do Diretório e de suas tabelas de estados. Do ponto de vista da implementação, são descritas suas estruturas de dados e interfaces e por fim, é apresentado o ambiente e a estrutura de implementação.

1 Introdução

O propósito de um Sistema de Diretório (DS) é fornecer Serviços de Diretório para seus usuários de modo a permitir o acesso às informações sobre objetos armazenados em diferentes sistemas abertos. A série X500 define um DS de acordo com os princípios do modelo OSI e, dessa forma, este sistema pode ser construído sobre qualquer rede física.

A especificação de um Sistema Diretório e futuramente sua implementação fazem parte de um projeto denominado INTER-X. Este projeto tem como objetivo principal gerar um Sistema de Manipulação de Mensagem (MHS) avançado e distribuído acoplado a um Sistema de Diretório padronizado que facilite a interação entre usuários na identificação e endereçamento dos usuários, além do gerenciamento de outras funções do sistema, como roteamento de mensagem e armazenamento de lista de distribuição.

O objetivo desse artigo é apresentar uma proposta de especificação do DS para o projeto INTER-X e está organizado da seguinte forma:

1. Introdução;
2. Uma Visão Geral do DS;
3. Utilização dos Serviços Subjacentes;
4. O funcionamento do DS e suas Tabelas de Estados;
5. Organização da Base de Informação do Diretório;
6. Arquitetura do Sistema Diretório;
7. Estruturas de Dados;
8. Interfaces;
9. Ambiente e Estrutura de Implementação;
10. Considerações Finais;

2 Uma Visão Geral do Diretório

Os aspectos abordados neste tópico são descritos mais detalhadamente em [7].

Diretório é um depósito de informações sobre objetos e fornece Serviços de Diretório para seus usuários de modo a permitir o acesso às informações armazenadas. A utilização dos Serviços de Diretório requer que os usuários e os componentes funcionais do Diretório cooperem entre si. Tal cooperação também é necessária entre aplicações em diferentes sistemas abertos, e para tal necessitam utilizar protocolos padronizados.

O usuário do Diretório é a entidade (aplicação) ou pessoa que acessa o Diretório com o propósito de obter Serviços de Diretório. Cada usuário é representado no acesso a um

Diretório por um Agente de Usuário de Diretório (DUA). O Diretório é composto de um ou mais processos denominados de Agente de Sistema de Diretório (DSA). Cada DSA fornece zero ou mais pontos de acesso. O papel do DSA é prover aos seus DUA's e/ou outros DSA's o acesso à Base de Informação do Diretório (DIB). Um DSA pode usar informações armazenadas em sua base de dados local ou interagir com outros DSA's para executar os pedidos.

As interações entre um DUA e um DSA localizados em sistemas abertos diferentes são suportadas pelo Protocolo de Acesso ao Diretório (DAP). De forma similar, as interações entre DSA's pertencentes a sistemas abertos diferentes são suportadas pelo Protocolo do Sistema Diretório (DSP). Tanto o DAP como o DSP são processos residentes na camada de aplicação.

O DAP e o DSP são protocolos para prover comunicação entre um par de processos de aplicação. No ambiente OSI isto é representado como sendo a comunicação entre um par de Entidades de Aplicação (AE's) utilizando o serviço de apresentação. A função de uma AE é fornecida por um conjunto de Elementos do Serviço de Aplicação (ASE's). A interação entre AEs é descrita em termos da utilização por elas dos serviços fornecidos pelo ASE's. Os dois ASEs comuns a ambos os protocolos do Diretório são: o Elemento de Serviço de Controle de Associação (ACSE) e o Elemento de Serviço de Operações Remotas (ROSE). O ROSE suporta o paradigma pedido/resposta de uma operação abtrata que ocorra nas portas do Modelo Abstrato de Diretório. Os ASEs do Diretório fornecem o mapeamento funcional dos serviços abstratos do Diretório na notação da sintaxe abstrata para os serviços fornecidos pelo ROSE. O ACSE suporta o estabelecimento e término de uma associação de aplicação entre um par de AEs. Associações entre DUA e um DSA podem ser somente estabelecidas pelo DUA. Somente o iniciador de uma associação estabelecida pode liberá-la.

O DAP é utilizado para efetuar os serviços de diretório. É composto por três ASE's específicos em adição ao ROSE e o ACSE, que são:

"ReadASE" - Suporta as operações da porta de leitura ou seja, leitura, comparação e abandono;

"SearchASE" - Suporta as operações da porta de busca ou seja, lista e busca;

"ModifyASE" - Suporta as operações da porta de modificação ou seja, adicionar entrada, remover entrada, modificar entrada e modificar RDN.

O DSP é utlizado para prover a funcionalidade de distribuição de operação. É composto por três ASE's específicos em adição ao ROSE e o ACSE, que são:

"ChainedReadASE" - Suporta as operações da porta de leitura encadeada ou seja, leitura, comparação e abandono encadeados.

"ChainedSearchASE" - Suporta as operações da porta de busca encadeada ou seja, lista e busca encadeadas;

"ChainedModifyASE" - Suporta as operações da porta de modificação encadeada ou seja, adicionar entrada, remover entrada, modificar entrada e modificar RDN [7] encadeadas.

O ASE's do Diretório que compòem o DAP compartilham uma única sintaxe abstrata, id-as-directory-AcessAS. Aqueles que compòem o DSP também compartilham uma única sintaxe abstrata id-as-directorySystemAS. As unidades de dados do protocolo de aplicação APDU do Diretório são definidas pela sintaxe abstrata dos ASE's do Diretório do ROSE e do ACSE.

Como sabemos, um contexto de aplicação (AC) especifica como a associação deve ser estabelecida, quais os elementos de serviços de aplicação (ASE's) estão envolvidos na comunicação e como a associação é terminada. Um ASE pode ser simétrico ou assimétrico dependendo se o usuário é consumidor ou fornecedor de serviços, e cabe ao contexto de aplicação especificar se as partes envolvidas na comunicação são consumidoras ou fornecedoras. O contexto de Aplicação do Acesso ao Diretório (directoryAccessAC) permite o DUA acessar as operações dos seguintes ASE's: ReadASE, searchASE, modifyASE, ACSE e ROSE. O contexto de Aplicação do Sistema de Diretório (directorySystemAC) permite o DSA acessar as operações dos seguintes ASE's: ChainedReadASE, ChainedSearchASE, ChainedModifyASE, ACSE e ROSE.

3 Utilização dos Serviços do CASE e das Camadas Inferiores

O nível de aplicação fornece um conjunto de elementos de serviços que são comuns a várias aplicações (CASE) e este subconjunto se subdivide em ROSE, ACSE e RTSE. As operações abstratas são mapeadas sobre estes elementos de serviços e em seguida passadas para a camada de apresentação. Neste tópico é apresentado como os protocolos DAP e DSP utilizam os serviços subjacentes.

- **Utilização dos serviços ROSE**

  O ROSE é definido na recomendação X219. Os ASEs do Diretório são os únicos usuários dos serviços RO-INVOKE, RO-RESULT, RO-ERROR, RO-REJECT-U e RO-REJECT-P do ROSE.

  Operações remotas do DAP e do DSP são operações de classe 2 (assíncrona). Observe que como o DUA é um consumidor do DAP, ele pode escolher operar em modo síncrono (operações de classe 1).

  DAP utiliza associações cuja classe é 1. Isto significa que o DSA não pode iniciar as operações para o DUA. Já o DSP utiliza associações cuja classe é 3. Isto significa que o DSA respondedor pode invocar as operações para o DSA inicializador e vice-versa.

- **Utilização dos Serviços ACSE**

  O ACSE é definido na recomendação X210. As operações BIND e UNBIND do Diretório são os únicos usuários dos serviços A-ASSOCIATE e A-RELEASE do ACSE no modo normal. O processo de aplicação é o usuário dos serviços A-ABORT e A-P-ABORT do ACSE.

- **Utilização dos Serviços de Apresentação**

  O serviço de apresentação é definido na recomendação X216. No modo normal, um contexto de apresentação diferente 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 de apresentação.

  O ROSE é o único usuário do serviço P-DATA de apresentação.

- **Utilização dos Serviços dos Níveis Inferiores**

  O serviço de sessão é definido na recomendação X215. As unidades funcionais Kernel e Duplex do nível de sessão são utilizadas pelo nível de apresentação.

  O serviço de transporte é definido na recomendação X214. A escolha da classe de transporte a ser utilizada pela sessão depende dos requisitos de multiplexação e recuperação de erros. Suporte para a classe 0 (sem multiplexação) é mandatória. Serviços de dados urgentes não são utilizados. Suporte para outras classes é opcional. A classe com multiplexação pode ser utilizada para multiplexar o DAP ou DSP e outros protocolos sobre a mesma conexão de rede. Uma classe com recuperação de erros deve ser escolhida sobre uma conexão de rede com uma razão de erro residual inaceitável.

  O serviço de rede é definido na recomendação X213. O endereçamento de rede é aquele definido na recomendação X121, recomendação E163/E164, ou recomendação X200.

4 O Funcionamento do DS e suas Estruturas de Dados

O procedimento básico do funcionamento do DS consiste em um ciclo de interrogações. Este ciclo analisa a comunicação com o módulo que implementa o CASE, analisa a comunicação com o usuário, e ativa quando necessário os procedimentos de transferência das operações para o módulo CASE ou para o usuário.

O sistema pode ser dividido em dois módulos, que são:

- O módulo DUA, que contém as funções ativadas pelo usuário e que correspondem aos serviços de diretório.

  A principal função de um Agente de Usuário de Diretório é acessar o Diretório a fim de obter serviços de diretório de interesse de um usuário. Além disso, o agente de

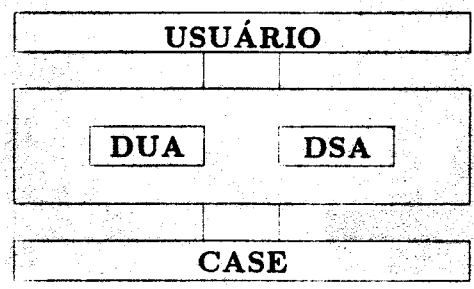

Figure 1: Modelo Funcional do DS.

usuário deve fornecer facilidades locais ao seu usuário de forma a melhor oferecer os serviços de diretório. Existem pedidos de serviços que permitem consultar o diretório (e.g. ler, pesquisar), que permitem modificar o Diretório (e.g. remover entradas, acrescentar entradas), e aqueles serviços que podem ser qualificados (e.g. controles como prioridade, tempo).

As operações de um DUA são descritas em termos das suas interações com o usuário e com o ACSE.

- O módulo DSA, cuja principal função é prover aos seus DUA's e/ou outros DSA's o acesso a DIB. Um DSA pode usar informações armazenadas em sua base de dados local ou interagir com outros DSA's para executar os pedidos.

  As operações de um DSA são descritas em termos das suas interações com o DUA e com o CASE.

4.1 Tabelas de Estados

Neste tópico é descrito o protócolo DAP e DSP em termos de tabelas de estados. Estas tabelas mostram o estado atual, os eventos que ocorrem nos protocolos DAP e DSP , as ações a serem tomadas e o estado resultante. A tabela de estados do DUA controla as mudanças de estados de uma associação entre o DUA com o DSA e a tabela de estados do DSA controla as mudanças de estados de uma associação entre um par de DSAs.

A seguir, é apresentada a notação usada na tabela de estados, descrevendo os estados e os eventos.

ESTADOS:

ESTADO	NOME	DESCRIÇÃO
INATIVO-DUA (DSA)	Estado Inativo	Não há conexão estabelecida entre o DUA e o DSA (entre dois DSA's)
W-ASS-CNF-DUA (DSA)	Estado de Confirmação de associação	O DUA (DSA) espera receber do DSA remoto uma confirmação do seu pedido de associação
ATIVO-DUA (DSA)	Estado Ativo	O DUA (DSA) está apto a submeter operações e receber respostas das operações do DSA remoto
W-RSP-DUA (DSA)	Estado de Espera das respostas de operações submetidas	O DUA (DSA) espera as respostas das operações submetidas para terminar a associação
W-REL-CNF-DUA (DSA)	Estado de Confirmação de término de Associação	O DUA (DSA) em questão espera receber do DSA remoto uma confirmação do seu pedido de término de associação

EVENTOS:

Os possíveis eventos do protocolo são gerados por primitivas de serviços ora acionadas pelo usuário através de pedidos de operações, ora acionadas pelo ACSE ou ROSE.

EVENTO	DESCRIÇÃO
BIND	Estabelecimento de acesso com o diretório
UNBIND	Término de acesso com o diretório
A-ASS-REQ (IND)	Pedido (Indicação) de estabelecimento de associação entre o DUA e o DSA
A-ASS-RSP (CNF)	Resposta (Confirmação) de estabelecimento de associação entre o DUA e o DSA
A-REL-REQ (IND)	Pedido (Indicação) de término de associação entre o DUA e o DSA

EVENTO	DESCRIÇÃO
A-REL-RSP (CNF)	Resposta (Confirmação) de término de associação entre o DUA e o DSA
RO-INV-REQ (IND)	Pedido (Indicação) para que a operação seja executada remotamente
RO-RES-REQ (IND)	Aviso (Indicação) de resposta de uma operação remota
A-ABORT-REQ (IND)	Pedido (Indicação) de liberação anormal de associação
RO-ERR-REQ (IND)	Aviso (Indicação) de erro na execução de uma operação remota
RO-REJU-REQ (IND)	Aviso (Indicação) de rejeição de um RO-INV-IND
RO-REJP-IND	Erro detetado pelo ROSE (resposta)
A-P-ABORT-IND	Liberação anormal solicitada pelo ACSE

TABELA DE ESTADOS DO DUA (PROVEDOR)

	INATIVO -DUA	*W-ASS-CNF -DUA*	*ATIVO -DUA*	*W-RSP -DUA*	*W-REL-CNF -DUA*
USR-BIND	1				
USR-UNBIND	9		3		
RO-REJP-IND			8		
RO-RES-IND			5	4	
RO-ERR-IND			12	13	
OPER		11	7		
A-ASS-CNF		2			
RO-REJU-IND			6	4	
A-REL-CNF					10

PROCEDIMENTO DO DUA PROVEDOR PARA O DAP

1. Inicializa DUA
   Envia a primitiva A-ASS-REQ para ACSE
   ESTADO = W-ASS-CNF-DUA

2. Se resultado da primitiva A-ASS-CNF é ACCEPTED
   ESTADO = ATIVO-DUA
   Envia USR-BIND de resultado para o usuário do DUA
   Enquanto tiver operações na lista de eventos (LEVENTOSDUA)
   Faça:
   retira operação da lista de eventos
   Coloca operação na lista de submissaào (LINVDUA)
   encapsula operação na primitiva RO-INV-REQ
   pendentes = pendentes + 1
   Senão
   ESTADO = INATIVO-DUA
   Envia USR-BIND de erro para usuário do DUA
   Retira as operações da lista de eventos (LEVENTOSDUA)

3. Se existem operações enviadas para o DSA sem resposta (pendentes > 0)
   ESTADO = W-RSP-DUA
   Senão
   envia A-REL-REQ
   ESTADO = W-REL-CNF-DUA

4. Pendentes = Pendentes - 1
   Retira mensagem da lista de submissão (LINVDUA)
   Se existe erro nos parâmetros da mensagem recebida
   envia mensagem de erro para usuário
   Senão /* mensagem não tem erro */
   envia mensagem de submissão OK para o usuário
   Igual ao procedimento 3

5. Pendentes = Pendentes - 1
   Retira msg da lista de submissão (LINVDDUA)
   Se existe erro nos parâmetros da mensagem recebida
   Igual procedimento 3
   Senão /* mensagem não tem erro */
   envia mensagem de submissão OK para usuário

6. Pendentes = Pendentes - 1
   Retira mensagem da lista de submissão (LINVDUA)
   Envia aviso de RO-REJU para usuário

7. Coloca operação na lista de submissão (LINVDUA)
   Encapsula a operação na primitiva RO-INV-REQ
   Pendentes = Pendentes + 1
   Envia operação para DSA

8. Envia aviso de RO-REJP para usuário
   Pendentes = Pendentes - 1
   Retira operação da lista de submissão (LINVDUA)
   Se razão setada
      ERRP = ERRP + 1
      Se ERRP > ERRMAX
         Envia a primitiva A-REL-REQ para ACSE
         ESTADO = W-REL-CNF-DUA
   Senão /* razão não setada */
      ESTADO = INATIVO-DUA

9. Envia USR-UNBIND de resultado para o usuário do DUA

10. Envia USR-UNBIND para usuário do DUA
    ESTADO = INATIVO-DUA

11. Coloca operação da lista de eventos (LEVENTOSDUA)

12. Pendentes = Pendentes - 1
    Retira mensagem da lista de submissão (LINVDUA)
    Se existe erro nos parâmetros da mensagem recebida
       Igual procedimento 3
    Senão /* mensagem não tem erro */
       Se erro indicar que operação foi abandonada
          retirar operação abandonada da lista
          Pendentes = Pendentes - 1
    Envia mensagem para usuário

13. Pendentes = Pendentes - 1
    Retira mensagem da lista de submissão (LINVDUA)
    Se existe erro nos parâmetros da mensagem recebida
        Envia mensagem de erro de parametro para usuário
    Senão
        Se erro indicar que operação foi abandonada
            Retirar operação abandonada da lista de submissão
            Pendentes = Pendentes - 1
            Enviar mensagem para usuário
    Igual procedimento 3

TABELA DE ESTADOS DO DSA

1 - TABELA GERAL DO DSA

	INATIVO
A-ASS-IND	1

Procedimento da Entidade DSA

1. Se A-ASS-IND veio de um DUA
    vá para Tabela 2
    Senão
    vá para Tabela 3

2 - TABELA RELACIONADA COM O DAP

	INATIVO-DUA	*ATIVO-DUA*
A-ASS-IND	1	
A-REL-IND		2
RO-INV-IND		4
RO-REJP-IND		3

Procedimento relacionado com o DAP

1. Checa credenciais do DUA
    Se sucesso
        Envia a primitiva A-ASS-RSP (+)
        ESTADO = ATIVO-DUA
    Senão
        Envia a primitiva A-ASS-RSP (-)

2. Envia a primitiva A-REL-RSP
   ESTADO = INATIVO-DUA

3. Se razão setada
   não faz nada
   Senão
   Estado = INATIVO-DUA

4. Se existe erro nos parâmetros da mensagem recebida
   Envia a primitiva RO-REJU-REQ
   Senão
   Ativa Operation Dispacher (NomeResolution + Evaluation)
   Case (retorno):
   Se retorno é erro:
   Encapsula em uma RO-ERR-REQ
   Envia ao originador
   Se retorno é resultado
   Ativa o procedimento Result-Merging
   Se retorno é referência externa
   Coloca operação em LRSP
   Para cada DSA referenciado
   resp-pendentes = resp-pendentes + 1
   Coloca operaccão na lista de submissão (LINVDSA)
   Pendentes = Pendentes + 1
   Encapsula operação em RO-INV-REQ
   Envia operação para outro DSA

3 - TABELA RELACIONADA COM O DSP

	INATIVO-DSA	*ATIVO-DSA*	*W-REL-CNF-DSA*	*W-ASS-CNF-DSA*	*W-RSP-DSA*
A-ASS-IND	1				
A-ASS-CNF				5	
A-REL-IND		2			
A-REL-CNF			6		
RO-INV-IND		4		8	
RO-REJP-IND		3			
RO-RES-IND		9			7
RO-ERR-IND		10			11

Procedimento relacionado com o DSP

1. Checa credenciais do DSA
   Se sucesso
      Envia a primitiva A-ASS-RSP (+
      Estado = ATIVO-DSA
   Senão
      Envia a primitiva A-ASS-RSP (-)

2. Se existem operações enviadas para outro DSA sem resposta
      ESTADO = W-RSP-DSA
   Senão
      Envia a primitiva A-REL-RSP
      ESTADO = INATIVO-DSA

3. Retira a mensagem da lista de submissão (LINVDSA)
   Se razão setada
      Se este DSA é o iniciador da conex ao
         ERRP = ERRP + 1
         Se ERRP > ERRPMAX
            Envia a primitiva A-REL-REQ
            ESTADO = W-REL-CNF-DSA
         Senão
            Se não existem operações enviadas para outro DSA sem resposta
               Envia a primitiva A-REL-REQ
               ESTADO = W-REL-CNF-DSA
      Senão
         Não faz nada
   Senão
      ESTADO INATIVO-DSA

4. Se existe erro nos parâmetros da mensagem recebida
      Envia a primitiva RO-REJU-REQ
   Senão
      Ativa Operation Dispacher (NomeResolution + Evaluation)
      Case (retorno):
            Se retorno é erro:
               Encapsula em uma primitiva RO-ERR-REQ
               Envia ao originador
            Se retorno é resultado
               Ativa o procedimento (Result-Merging)

Se retorno é referência externa
Coloca operação em LRSP
Para cada DSA referenciado
resp-pendentes = resp-pendentes + 1
Coloca operação na lista de submissão (LINVDSA)
Pendentes = Pendentes + 1
Encapsula operação em RO-INV-REQ
Envia operação para DSA referenciado

5. Se resultado da primitiva A-ASS-CNF é ACCEPT
ESTADO = ATIVO-DSA
Enquanto tiver operações na lista de eventos (LEVENTOSDSA)
Faça:
Retira operação da lista de eventos
Igual procedimento 4
Senão
ESTADO = INATIVO-DSA
Enquanto existirem operações na lista de eventos (LEVENTOSDSA)
Faça:
Retira operações da lista de eventos
Enviar primitiva de RO-ERR-REQ para associação anterior

6. ESTADO = INATIVO-DSA

7. Pendentes = Pendentes - 1
Retira mensagem da lista de submissão (LINVDSA)
Se existe erro nos parâmetros da mensagem recebida
Se este DSA é o iniciador da conexão
Se existem operações pendentes
ESTADO = W-RSP-DSA
Senão
Envia a primitiva A-REL-REQ
ESTADO = W-REL-CNF-DSA
Senão
Descarta
Senão
Ativa o procedimento (Result-Merging)
Igual ao procedimento 2

8. Coloca operação na lista de eventos (LEVENTOSDSA)

9. Pendentes = Pendentes - 1
   Retira mensagem da lista de submissão (LINVDSA)
   Se existe erro nos parâmetros da mensagem recebida
      Se este DSA é o iniciador da conexão
         Se existem operações pendentes
            ESTADO = W-RSP-DSA
         Senão
            Envia a primitiva A-REL-REQ
            ESTADO = W-REL-CNF-DSA
      Senão
         Descarta
   Senão
      Ativa o procedimento (Result-Merging)
   Se não existirem operações enviadas para outro DSA sem resposta
      Se este DSA é iniciador da conexão
         Envia a primitiva A-REL-REQ
         ESTADO = W-REL-CNF-DSA

10. Pendentes = Pendentes - 1
   Retira mensagem da lista de submissão (LINVDSA)
   Se existe erro nos parâmetros da mensagem recebida
      Se este DSA é o iniciador da conexão
         Se existirem operações pendentes
            ESTADO = W-RSP-DSA
   Senão
      Se erro indicar que operação foi abandonada
         Retirar operação abandonada da lista LINVDSA
         Pendentes = Pendentes - 1
         Ativa procedimento (Result-Merging)
   Se não existirem operações enviadas para outro DSA sem resposta
      Se este DSA é o iniciador da conexão
         Envia a primitiva A-REL-REQ
         ESTADO = W-REL-CNF-DSA

11. Pendentes = Pendentes - 1
   Retira mensagem da lista de submissão (LINVDSA)
   Se existe erro nos parâmetros da mensagem recebida
      Se este DSA é o iniciador da conexão
         Se existirem operações pendentes
            ESTADO = W-RSP-DSA
   Senão
      Se erro indicar que operação foi abandonada

Retirar operação abandonada da lista LINVDSA
Pendentes = Pendentes - 1
Ativa procedimento (Result-Merging)
Igual procedimento 2

Procedimento do Result-Merging

Verifica a qual operação os resultados/erros se referem
Se os resultados/erros forem locais
Se a operação é simples
Encapsula o resultado /erro em uma RO-RESULT-REQ / RO-ERR-REQ
Envia para o originador e termina
Senão
Case (lista)
Se a lista contém apenas resultados
Encapsula em uma primitiva RO-RESULT-REQ
Envia para o originador e termina
Se a lista contém apena erros
Encapsula em uma primitiva RO-ERR-REQ
Envia para o originador e termina
Se a lista contém resultados e erros
Monta um resultado incompleto
Encapsula em uma primitiva RO-RESULT-REQ incompleta
Envia para o originador e termina
Senão
Se a operação é simples
Retira a operação da lista de resposta (LRSP)
Envia RO-RESULT-REQ / RO-ERR-REQ para o originador e termina
Senão
Resp-pendentes = Resp-pendentes - 1
Inclui resultados/erros na lista de respostas da operação
Se não existirem respostas pendentes
Retira a lista de respostas de LRSP
Case (lista)
Se lista contém apenas resultados
Encapsula em RO-RES-REQ
Envia ao originador da operação e termina
Se lista contém apenas erros
Encapsula em RO-ERR-REQ
Envia ao originador da operação e termina
Se lista contém resultados e erros
Monta um resultado incompleto

Encapsula em uma RO-RES-REQ
Envia para o originador e termina

5 Organização da Base de Informação do Diretório

Neste item é apresentado um modelo para a organização da Base de Informação do Diretório (DIB). Procurou-se definir um esquema de armazenamento e estrutura de acessos simples, que permitissem uma maior facilidade no retorno das informações. O tempo de resposta foi considerado como item principal e procurou-se sempre minimizá-lo, principalmente com o emprego de "buffers". Sob este ponto de vista, outros ítens, como o espaço de armazenamento tiveram menor importância.

Tendo em vista que o número de leituras neste tipo de base de dados é muito maior que o número de inserções, remoções e atualizações, procurou-se definir uma estrutura de armazenamento que favorecesse as consultas ainda que em detrimento das inserções.

Quanto às remoções e atualizações, estas são compostas de uma consulta seguida de uma operação de escrita no arquivo. Assim, apesar de serem pouco frequentes, têm tempos de resposta reduzidos, já que estes dependem basicamente dos tempos das consultas.

A base de dados de Diretório (DIB) foi dividida em arquivos da seguinte forma: cada nível da árvore de informações de Diretório (DIT) está armazenado em um arquivo diferente, de nome nívelX.dsk, onde X é o número do nível da entrada na árvore. Na figura 2, a árvore tem 4 níveis (não contando com a raiz). Teríamos então 4 arquivos:

- nível1.dsk, que armazenaria o nível 1 da árvore, representado pelos países;
- nível2.dsk, que armazenaria as organizações UFRJ e UCLA e cidades Rio de Janeiro, São Paulo e Nova Iorque;
- nível3.dsk, que armazenaria os registros das pessoas José, João, Maria, Pedro, John, Peter, Mary e das unidades organizacionais NCE, CT, D1 e D2;
- nível4.dsk, que armazenaria os registros das pessoas Marcelo, Pat, Luci, Linda e Mike.

Pode-se observar que os arquivos agruparam classes de objetos específicas:

- nível1.dsk só possuirá entradas da classe "country";
- nível2.dsk possuirá entradas das classes "organization" , "locality" e "ADMD"
- nível3.dsk terá entradas das classes "residential person", "organization unit", "PRMD"
- nível4.dsk terá nomes O/R e "organizational persons".

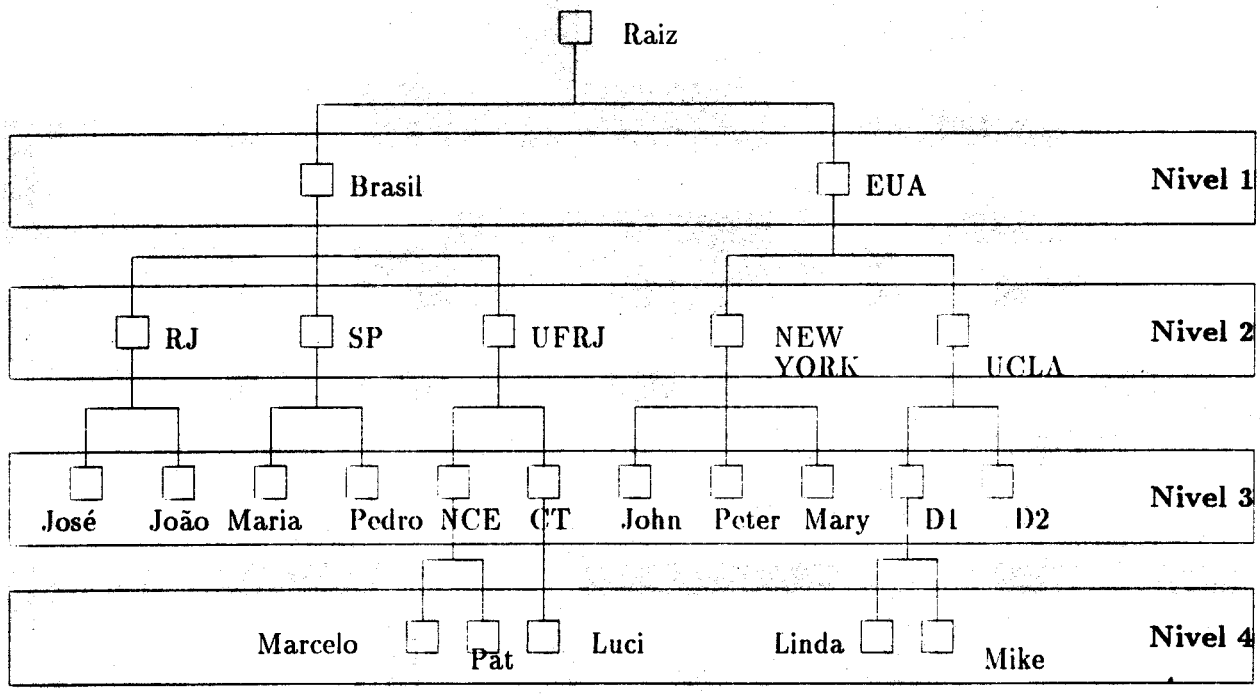

Figure 2: Exemplo de uma DIB.

Com essa estrutura em níveis e divisão em arquivos, consegue-se simplificar a base de dados e satisfazer às características heterogêneas de suas entradas.

Cada entrada de Diretório representa um nó na árvore de informações do Diretório (DIT). Isto significa que as entradas estão organizadas segundo uma hierarquia de acordo com a classe a que pertencem. Como já foi dito, organizou-se a base de informação seguindo a estrutura da DIT.

Para fazer a conexão entre um nó pai da DIT e seus nós filhos, colocou-se em cada entrada um ponteiro para a posição no arquivo de nível imediatamente inferior em que o registro de seu filho mais à esquerda se encontra. Os registros dos outros filhos estarão organizados sequencialmente ao primeiro, de forma que ponteiros entre nós irmãos são desnecessários. O último filho (o mais à direita na DIT) conterá uma marca indicando que ele é o último.

5.1 Operação

Para maior rapidez nas operações, são mantidos em memória "buffers" - um para cada arquivo. Esses "buffers" contêm trechos dos arquivos e são atualizados quando necessário. Assim, praticamente todo o processamento necessário por uma operação é feito em memória, diminuindo o tempo de resposta.

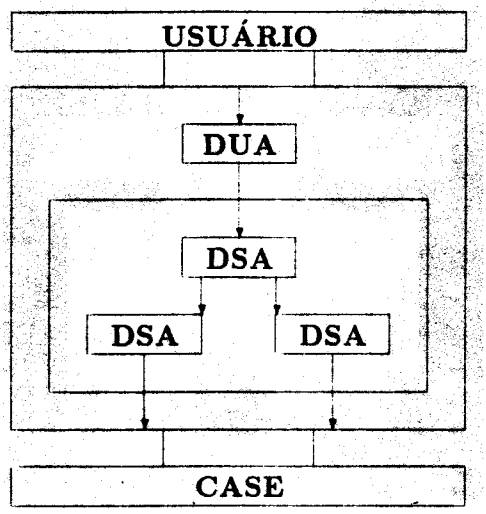

Figure 3: Funcionamento do DS.

O esquema de armazenamento, com nós irmãos colocados sequencialmente nos arquivos, reforça ainda mais a ação dos buffers, já que a probabilidade de um nó procurado estar no buffer carregado do disco a partir do ponteiro indicado pelo seu nó pai é muito grande. Neste caso, o nó filho desejado, com grande probabilidade, será encontrado em um único acesso ao disco. Se o nó procurado já estiver no buffer atualmente residente na memória, então não será necessário carregá-lo do disco, o que melhora ainda mais a performance deste esquema.

No entanto, para inserir um novo registro num arquivo, torna-se necessário rearrumá-lo completamente. Um novo arquivo deve ser criado, e os registros iniciais devem ser copiados para ele. Depois então o novo registro é inserido e os registros finais copiados. Como já foi dito, a performance das inserções foram prejudicadas pois partiu-se do princípio de que a entrada é inserida uma vez mas recuperada várias vezes.

6 Arquitetura do Sistema de Diretório

O procedimento básico do funcionamento do Sistema de Diretório (DS) consiste de um ciclo de interrogações. Este ciclo analisa a comunicação com o módulo que implementa os Elementos de Serviços comuns da camada de Aplicação (CASE), analisa a comunicação com o usuário, e ativa quando necessário os procedimentos de transferência das operações para o módulo CASE ou para o usuário. Cada análise realizada se refere a uma possível recepção de uma operação ou um resultado de uma operação proveniente do módulo CASE ou do usuário. O funcionamento do DS é ilustrado na figura 3.

O funcionamento do Sistema de Diretório pode ser mais facilmente compreendido se o mesmo for descrito através de seus dois componentes DUA e DSA's.

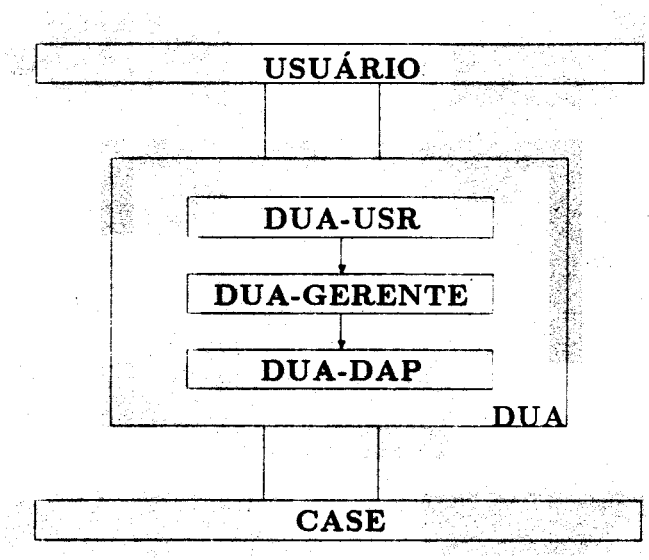

Figure 4: Arquitetura Interna do DUA.

6.1 Arquitetura Interna do Agente Usuário do Diretório (DUA)

A arquitetura interna do DUA divide suas funções em três módulos principais: o módulo DUA-GERENTE, o módulo DUA-DAP e o módulo DUA-USR. Esta arquitetura é ilustrada na figura 4. O módulo DUA-USR é a parte do DUA que interage diretamente com o usuário do Sistema de Diretório, apresentando seus recursos de inclusão, busca e alteração dos dados de diretório e fornecendo facilidades para que estes recursos possam ser acessados de maneira simples e eficiente, além de prover serviços locais de navegação pela base de dados distribuída e nome amigável para acessar uma base de dados local.

Os recursos do DUA-USR são oferecidos através de comandos selecionados pelo usuário através de menus de barra. Estes comandos podem ativar dois tipos de funções: funções executadas no próprio módulo ou funções executadas no módulo DUA-GERENTE. As funções próprias do módulo são aquelas de âmbito local e correspondem ás facilidades estritamente locais, não sendo objeto de padronização. As funções executadas no módulo DUA-GERENTE são aquelas que requerem interação com o DSA remoto. Estas funções correspondem ás operações de Serviço de Diretório.

O módulo DUA-DAP processa a máquina lógica que opera a transição de estados do Protocolo de Acesso ao DSA (DAP), baseada na ocorrência de eventos provocada pelo módulo DUA-GERENTE. O protocolo DAP suporta as interações entre um DUA e um DSA localizados em sistemas abertos distintos. Por outro lado, este módulo torna disponível para o módulo DUA-GERENTE qualquer resposta à operações previamente submetidas. Este módulo utiliza os serviços dos Elementos de Serviços Comuns da Camada de Aplicação (CASE), a saber: o ACSE, que estabelece e controla a associação entre as entidades de aplicação e o ROSE, que realiza as operações remotas.

O módulo DUA-GERENTE é responsável por mapear solicitações de operações do módulo DUA-USR em primitivas de serviço para o CASE e ativar a máquina de estados do DUA isto é, ativar o módulo DUA-DAP. Por outro lado, este módulo torna disponível para o módulo DUA-USR qualquer resposta à operações previamente submetidas. Este módulo também realiza operações de gerenciamento que visam controlar as interações entre DUA e DSA. Este módulo também é responsável em inicializar as estruturas de dados utilizadas no DUA.

6.2 Arquitetura Interna do Agente Sistema do Diretório (DSA)

A arquitetura interna do DSA divide suas funções em quatro módulos principais: o módulo DSA-GERENTE, o módulo MAQEST-DUA, o módulo MAQEST-DSA e o módulo OPERATION DISPACHER.

O módulo DSA-GERENTE é responsável por verificar a origem das solicitações ou respostas de operações remotas e ativar a máquina de estados do Protocolo de Acesso ao Diretório (DAP) ou a máquina de estado do Protocolo do Sistema de Diretório (DSP). Este módulo também realiza operações de gerenciamento que visam controlar as interações entre DUA e DSA e entre DSA's. Ele também é responsável em inicializar as estruturas de dados utilizadas no DSA.

O módulo MAQEST-DUA processa a máquina lógica que opera a transição de estados do Protocolo de Acesso ao Diretório (DAP), baseado na ocorrência de eventos provocada por um DUA adjacente e remoto. Este módulo utiliza os serviços dos Elementos de Serviços Comuns da Camada de Aplicação (CASE) e é responsável por ativar o módulo OPERATION DISPACHER para possibilitar a execução das operações submetidas pelo DUA adjacente.

O módulo MAQEST-DSA processa a máquina lógica que opera a transição de estados do Protocolo do Sistema de Diretório (DSP), baseado na ocorrência de eventos provocada por um DSA remoto e adjacente. Este módulo utiliza os serviços CASE e é responsável por ativar o módulo OPERATION DISPACHER para solicitar a execução das operações submetidas por um DSA remoto ou processar as respostas de operações previamente submetidas por este DSA.

O módulo OPERATION DISPACHER é responsável em validar a operação recebida em um DSA verificando a ocorrência de loop ou erros de autenticação. Se não ocorreu problema são acionados sequencialmente 3 submódulos que são: NAME RESOLUTION, AVALIAÇÃO e MERGE RESULT. O submódulo "NAME RESOLUTION" verifica se a operação submetida referencia uma entrada pertencente a base de dados local. Caso a operação não possa ser processada neste DSA, ela será posteriormente transmitida para um outro DSA ou posteriormente enviada de volta ao originador da operação. O submódulo "AVALIAÇÃO" é disparado para processar a operação em questão. O submódulo "MERGE RESULT" junta todos os resultados internos e externos a este DSA formando um único resultado e retorna-o para o DUA ou DSA chamador. Este módulo utiliza os serviços do CASE.

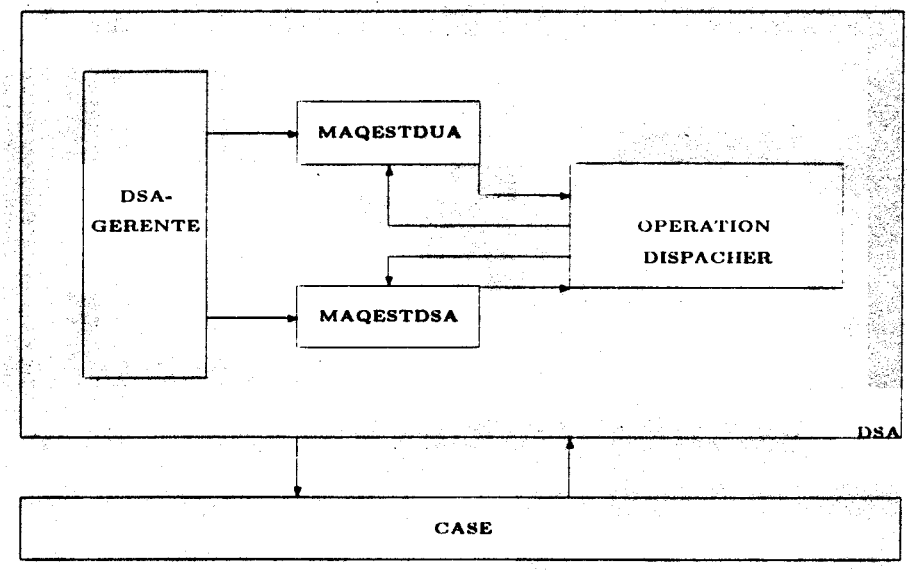

Figure 5: Diagrama de Blocos do DSA.

7 Estruturas de Dados

7.1 Estrutura de Dados do DUA

Quanto às estruturas de dados, o modelo de implementação propõe que o DUA opere sobre áreas de dados em memória e em discos, tabelas de informações, listas e filas de distribuição de eventos.

A área de dados é composta das áreas das mensagens recebidas do DSA remoto.

A cada DUA 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 DSA remoto não foi completada.

O DUA mantém estruturas de dados para armazenamento de informações relativas às operações sendo realizadas. Esta estrutura são listas encadeadas que são criadas e manipuladas dinamicamente em memória conforme as operações são solicitadas e realizadas.

As tabelas armazenam informações cujos campos podem ser alterados pelo DUA durante uma sessão de diálogo entre DUA e um DSA remoto de acordo com as operações invocadas.

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.

7.1.1 Tabelas

- **TAB-DSAR (tabela relativa à associação com DSA remoto)**

  A TAB-DSAR é uma tabela que contém informações relativas à associação desse DUA com cada DSA remoto e adjacente tais como, o nome do DSA remoto, o estado em que se encontra a máquina de estados do protocolo DAP e o número de operações transmitidas mas não respondidas. Esta tabela contém um registro para cada DSA adjacente a esse DUA cujos campos podem ser alterados pelo DUA e são descritos abaixo:

DSA	*ESTADO-DSA*	*ERRP*	*ERR*	*PENDENTES*

  onde,

  - **DSA** : Indica o nome do DSA remoto
  - **ESTADO-DSA** : Indica o estado em que se encontra a máquina de estados do DUA
  - **ERR** : Indica o número de erros ocasionados por um RO-REJECT-U
  - **ERRP** : Indica o número de erros ocasionados por um RO-REJECT-P
  - **PENDENTES** : Indica o número de operações transmitidas mas não respondidas

- **TAB-IDENT-DUA (Tabela Relativa ao DUA)**

  Essa tabela armazena todas as informações pertinente ao DUA tais como, credenciais de acesso ao DUA, nome do DUA e a versão. O nome do DUA dado aqui é o nome que os DSA's adjacentes devem registrar em suas tabelas de DUA's, para permitir o estabelecimento da conexão entre os DSA's e esse DUA. Esta tabela contém somente um registro cujos campos são descritos abaixo.

DUA	*CREDENCIAIS*	*VERSÃO*

  onde,

  - **DUA** : Indica o nome do DUA
  - **CREDENCIAIS** : Indica as credenciais de acesso do usuário ao DUA
  - **VERSÃO** : Indica a versão

7.1.2 Listas

- **LINVDUA (Lista relativa às operações submetidas a um DSA remoto e adjacente)**

  Esta lista é utilizada para armazenar informações sobre quaisquer operações solicitadas pelo DUA a um DSA remoto tais como, o identificador da operação submetida, o tamanho da operação e a operação propriamente dita. Existe uma lista desta para cada DSA remoto e adjacente a este DUA.

  São necessários dois registros para a inserção de uma operação na lista relativa às operações remotas submetidas pelo DUA a um DSA adjacente. O primeiro conteria informações sobre a última operação da lista e o segundo conteria informação sobre a operação a ser inserida. Este último registro é também utilizado para a leitura de uma operação da lista.

  Os registros apresentam o seguinte formato:

NLST	*ID*	*TAM*	*OPER*	*WAITRESULT*

  Onde,

  - **NLST** : Aponta para o próximo da lista
  - **ID** : Contém o identificador da operação submetida para Transferência pelo ROSE. É idêntica ao parâmetro "RO-INVOKE-ID" da primitiva RO-INVOKE do ROSE
  - **TAM** : Tamanho da operação
  - **WAITRESULT** : Indica se a operação em questão está aguardando resultado
  - **OPER** : Contém o formato das operações submetidas

- **LEVENTOSDUA (Lista de Distribuição de Eventos)**

  Esta lista é utilizada para armazenar informações sobre quaisquer operações solicitadas por este DUA a um DSA remoto mas ainda não transmitidas, pois a associação entre o DUA e o DSA não foi estabelecida. Existe uma lista desta para cada DSA remoto e adjacente ao DUA em questão e contém as mesmas informações da lista anterior exceto o campo WAITRESULT.

- **LPTRINVDUA (Lista de cabeça das listas LINVDUA e LEVENTOSDUA)**

  Esta é uma lista utilizada como cabeça das listas relativas às operações remotas submetidas para um DSA (LINVDUA) e das listas de distribuição de eventos (LEVENTOSDUA). Cada item da lista de cabeças de listas contém o nome do DSA remoto e adjacente ao DUA em questão e apontadores para o início e fim da lista LINVDUA e da lista LEVENTOSDUA.

  Cada item da lista de cabeça de lista apresenta o seguinte formato:

DSA		*INV*		*EVENTO*	
NAME	AP	IPTR	FPTR	IPTR	FPTR

onde,

- **AP** : Endereço de apresentação do DSA
- **IPTR** : Indica o início da lista
- **FPTR** : Indica o fim da lista
- **INV** : Relacionado com a lista LEVENTOSDUA

Toda vez que a lista de cabeça de listas for atualizada, sua cópia em disco também o será.

7.2 Estruturas de Dados do DSA

Quanto às estruturas de dados, o modelo de implementação propõe que o DSA 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 de um DSA ou DUA remoto.

A cada DSA 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. Não é necessário uma fila deste tipo para a associação com o DUA adjacente pois é o DUA e somente ele o responsável em estabelecer a associação com qualquer DSA conectado diretamente a ele, isto é, é o DUA que é o responsável em submeter operações para um DSA. Sendo assim, estas filas são utilizadas enquanto a associação com o DSA remoto não foi completada.

O DSA mantém 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. Existem tabelas que armazenam dados relativos a um DUA conectado diretamente a esse DSA; seus campos podem ser alterados pelo DSA durante uma sessão de diálogo entre este DSA e o DUA de acordo com as operações invocadas pelo DUA.

Existem também tabelas que armazenam dados relativos a um DSA remoto adjacente a esse DSA; seus campos podem ser alterados pelo DSA durante uma sessão de diálogo entre este DSA e o DSA remoto 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.

7.2.1 Tabelas

- **TAB-DSAR (Tabela relativa à associação com um DSA remoto)**

  Esta tabela contém informações relativas à associação desse DSA com seus DSA's remotos e adjacentes tais como, o nome do DSA remoto, o estado em que se encontra a máquina de estados do DSP, o número de operações transmitidas mas não respondidas, e se este DSA foi o responsável pela inicialização da associação. Ela contém um registro para cada um dos seus DSA's adjacentes cujos campos podem ser alterados pelo DSA local e são descritos abaixo:

DSA	*ESTADO-DSA*	*RX*	*ERRP*	*ERR*	*FIM-RES*	*PENDENTES*

  - **DSA** : Indica o nome do DSA remoto
  - **ESTADO-DSA** : Indica o estado em que se encontra a máquina de estados do protocolo DSP
  - **RX** : Informa se este DSA local foi o responsável pela inicialização da associação
  - **ERRP** : Indica o número de erros ocasionados por um RO-REJ-P
  - **ERR** : Indica o número de erros ocasionados por um RO-REJ-U
  - **FIM-RES** : Informa se está pronto para transmitir o resultado para o originador
  - **PENDENTES** : Indica o número de operações transmitidas mas não respondidas

- **TAB-DUAR (Tabela relativa à associação com um DUA remoto)**

  Esta tabela contém informações relativas à associação desse DSA com seus DUA's adjacentes. Ela contém um registro para cada um dos seus DUA's adjacentes cujos campos podem ser alterados por este DSA local e são descritos abaixo:

DUA	*ESTADO-DUA*	*ERRP*

  Onde,

  DUA : Indica o nome do DUA

  ESTADO-DUA : Indica o estado em que se encontra a máquina de estados do protocolo DAP

- **INFODSA (Tabela relativa ao DSA)**

  Esta tabela armazena informações pertinentes ao DSA tais como, nome do DSA, endereço de apresentação do DSA, tipo de interação permitida entre DSA's e informações relativas aos contextos existentes neste DSA.

MYDSA		*DSPCHAIN*	*INFOCONTEXTO*
DSA	AP		

INFOCONTEXTO												
NCONT	REFINT		REFSUB			REFSUBN		REFSUP		REFCROSS		
	RDN	PTR	RDN	DSA	AP	DSA	AP	DSA	AP	CONT	DSA	AP

Onde,

- **AP** : Endereço de apresentação do DSA
- **DSPCHAIN** : Tipo de interação permitida entre DSA'S default (referral)
- **MYDSA** : Informação relativa ao meu DSA
- **NCONT** : Nome do Contexto
- **REFINT** : Referência interna
- **REFSUB** : Referência subordinada para outros contextos
- **REFSUBN** : Referência subordinada não específica para outros contextos
- **REFSUP** : Referência superior
- **REFCROSS** : Referência cruzada

- **TAB-DUA (Tabela de DUA's adjacentes)**

  Esta tabela contém informações de todos os DUA's adjacentes a este DSA em questão, como as credenciais que contém o nome e a "password" do DUA.

- **TAB-DSA (Tabela de DSA'S adjacentes)**

  Esta tabela contém informações de todos os DSA's adjacentes a este DSA em questão, como o nome do DSA e seu endereço de apresentação.

DSA	*AP*

- **ENTRY (Tabela relativa a uma entrada)**

  Esta tabela contém todas as informações de uma entrada na Base de Informações do Diretório (DIB), como o nome da entrada, a classe do objeto, se a entrada é folha, se é ALIAS (nome alternativo de um objeto), etc. Esta tabela contém somente um único registro cujos campos são descritos abaixo:

RDN	*CLASS*	*COMPLETE*	*DATA*	*PASS*	*ALIAS*	*ALLCHILD*	...

...	*DSAMASTER*	*DSASLAVE*	*VER*	*STATUS*	*CHILD*

Onde,

- **RDN** : Nome da entrada
- **CLASS** : Classe do objeto
- **LEAF** : A entrada é folha
- **COMPLETE** : Estão todos os atributos na entrada
- **DATA** : É cópia ou não
- **PASS** : Password da entrada
- **ALIAS** : É alias
- **ALLCHILD** : Todos filhos presentes
- **DSAMASTER** : DSA que contém o original
- **VERS** : Número da versão
- **STATUS** : Estado da Entrada (ok, vazia, removida)
- **CHILD** : PTR para o filho mais a esquerda

7.2.2 Listas

- **LINVDSA (Lista relativa às operações remotas submetidas para um DSA)**

  Esta lista é utilizada para armazenar informações sobre quaisquer operações solicitadas por este DSA a um DSA remoto tais, como o identificador da operação submetida, o tamanho da operação e a operação propriamente dita.

  São necessários dois registros para a inserção de uma operação na lista relativa às Operações Remotas de um dado DSA. O primeiro conteria informações sobre a última operação da lista e o segundo conteria informações sobre a operação a ser inserida. Este último registro é também utilizado para a leitura de uma operação da lista.

  Os registros apresentam o seguinte formato:

NLST	*ID*	*TAM*	*OPER*	*WAITRES*

Onde,

- **NLST** : Aponta para o próximo da lista
- **ID** : Contém o identificador da operação submetida para Transferência pelo ROSE. É idêntica ao parâmetro "RO-INV-ID" da primitiva RO-INV do ROSE
- **TAM** : Tamanho da operação
- **WAITRES** : Indica se a operação em questão está aguardando resultado
- **OPER** : Contém o formato das operações submetidas

- **LRSP (Lista de respostas pendentes)**

  Esta lista é utilizada para armazenar informações sobre as respostas de uma operação de um determinado DSA. Existe uma lista desta para cada DSA remoto e adjacente ao DSA em questão e contém as mesmas informações da lista anterior exceto o campus WAITRES.

- **LEVENTOSDSA (Lista de distribuição de eventos)**

  Esta lista é utilizada para armazenar informações sobre quaisquer operações solicitadas por este DSA a um DSA remoto mas ainda não transmitidas, pois a associação entre os dois DSA's não foi estabelecida. Existe uma lista desta para cada DSA remoto e adjacente ao DSA em questão e contém as mesmas informações da lista anterior exceto o campo WAITRES.

- **LPTRINVDSA (Lista de cabeça das listas LINVDSA, LEVENTOSDSA e LRSP)**

  Esta é uma lista utilizada como cabeça das listas relativas ás operações remotas submetidas para um DSA (LINVDSA), das listas de distribuição de eventos (LEVENTOSDSA) e das lista de respostas pendentes (LRSP). Cada item da lista de cabeças das listas contém o nome do DSA remoto e adjacente ao DSA em questão e apontadores para o início e fim da lista LINVDSA, da lista LEVENTOSDSA e da lista LRSP. A seguir é apresentado o formato da Lista de Cabeça de Lista:

DSA		*INV*		*EVENTO*		*RESPOSTA*	
NAME	AP	IPTR	FPTR	IPTR	FPTR	IPTR	FPTR

Onde,

- **NAME** : Nome do DSA
- **IPTR** : Indica o início da lista
- **FPTR** : Indica o fim da lista
- **INV** : Relacionada com a lista LINVDSA
- **EVENTO** : Relacionada com a lista LEVENTOSDSA
- **RESPOSTA** : Relacionada com a lista LRSP

Toda vez que a lista de cabeça de listas for atualizada, sua cópia em disco também o será. As listas são armazenadas e manipuladas em disco.

8 Interfaces

8.1 Interface do DUA

Uma Entidade de Agente de Diretório deve comunicar-se ora com o usuário para prestar-lhe os serviços de diretório, ora com os Elementos de Serviço Comuns da Camada de Aplicação (CASE), os quais efetuam estabelecimento de conexão e transferência de dados ao DSA remoto. As duas interfaces através das quais o DUA se comunica são: a interface com o usuário, módulo DUA-USR, e a interface com o módulo CASE.

8.1.1 Interface DUA/Módulo DUA-USR

A maneira segundo a qual as operações do Sistema de Diretório são apresentadas ao usuário não faz parte do escopo da série de recomendações X500. No caso do Diretório-NCE, existe um módulo de auxílio ao usuário, chamado módulo DUA-USR, o qual é ligado ao módulo DUA. As operações abstratas do DUA são ativadas pelo módulo DUA-USR através da chamada de funções (primitivas) que preenchem os parâmetros corretamente, com informações obtidas do usuário, e montam as estruturas das operações que serão enviadas pelo diretório. As operações abstratas do Diretório estão padronizadas na recomendação X511.

Quem determina quando uma função deve ser ativada é o próprio usuário através da escolha de uma das opções do menu do Diretório.

A informação entre estes módulos é trocada sob a forma de primitivas de serviços que foram definidas localmente. Existem dez primitivas entre os módulos DUA e a DUA-USR. Estas primitivas são utilizadas para que o usuário possa solicitar as operações de Diretório e para que o DUA possa passar ao usuário alguma informação sobre as operações solicitadas.

Primitivas (funções) Invocadas pelo módulo DUA-USR

ARGBIND - Uma chamada de uma operação abstrata é sempre precedida pela operação BIND, caso não haja um estabelecimento de acesso com o Diretório. Isto é, a interface chama uma função para preencher os parâmetros do argumento da operação BIND. Após este preenchimento, a operação é enviada ao DUA.

Se algum erro ocorrer durante a execução desta operação, o DUA notificará a interface que informará ao usuário através de uma mensagem na tela. Isto também é válido para as demais operações abstratas que podem ser ativadas peloa módulo DUA-USR.

Parâmetros : Versão do serviço solicitado;
Credenciais do usuário.

ARGREAD - Leitura de uma entrada da DIB

Parâmetros: Nome distinto da entrada objeto;
Seleção das informações da entrada requerida;
Argumentos comuns às operações abstratas.

ARGCOMPARE - Comparação de um atributo fornecido com um atributo de uma entrada;

Parâmetros: Nome distinto da entrada objeto;
Argumentos propostos (tipo e valor) para comparação;
Argumentos comuns às operações abstratas.

ARGABANDON - Abandonar uma determinada operação;
Parâmetros: InvokeId da operação.

ARGLIST - Retorna uma lista de subordinados de uma determinada entrada da DIT;

Parâmetros: Nome distinto da entrada objeto;
Argumentos comuns às operações abstratas.

ARGSEARCH - Retorna uma lista de entradas da DIT que satisfazem algum filtro;

Parâmetros: Nome distinto da entrada objeto;
Argumento de subconjunto;
Filtro;
Indicação para procura ou não de entradas ALIAS;
Seleção das informações da entrada requerida;
Argumentos comuns às operações abstratas.

ARGADDENTRY - Adicionar uma nova entrada na DIT;

Parâmetros: Nome distinto da entrada objeto;
Atributos da nova entrada;
Argumentos comuns às operações abstratas.

ARGREMOVEENTRY - Remover uma entrada na DIT;

Parâmetros: Nome distinto da entrada objeto;
Argumentos comuns às operações abstratas.

ARGMODIFYENTRY - Modificar uma entrada na DIT;

Parâmetros: Nome distinto da entrada objeto;
Modificações da entrada;
Argumentos comuns às operações abstratas.

ARGMODIFYRDN - Modificar o RDN de uma determinada entrada;

Parâmetros: Nome distinto da entrada objeto;
Novo nome relativo distinto (RDN) da entrada;
Indicação para deleção ou não do nome relativo (RN) antigo;
Argumentos comuns às operações abstratas.

Os pedidos por execução de operações são mantidos em uma fila no DUA. Uma operação é retirada desta fila assim que ela é completada. Quando esta fila está vazia, o DUA envia uma operação UNBIND para indicar o fim de acesso ao diretório e a interface é notificada. A operação UNBIND não possui parâmetros.

8.1.2 Interface DUA/CASE

A interface entre o DUA e o CASE engloba todas as primitivas de serviços que são oferecidas pelo ACSE e ROSE.

O serviço do ACSE é definido por quatro primitivas:

1. **A-ASSOCIATE** - Estabelece uma Associação de Aplicação.
2. **A-RELEASE** - Causa o término de uma associação de aplicação sem perda de informação.
3. **A-ABORT** - Causa o término anormal de uma associação de aplicação com a possibilidade de perda de informação em trânsito.
4. **A-P-ABORT** - Indica o término anormal de uma associação de aplicação pelo ACSE com a possibilidade de perda de informação em trânsito.

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

1. **RO-INVOKE** - Solicita a execução de uma operação.
2. **RO-RESULT** - Retorna o resultado de uma operação bem sucedida.
3. **RO-ERROR** - Retorna erros da execução de uma operação.
4. **RO-REJECT-U** - Retorna falha em uma das primitivas acima detectada no sistema remoto.
5. **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 síncrona, ou seja, o iniciador de uma operação necessita receber uma resposta antes de solicitar outra operação. Operações de 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 condição de erro.

Deve sempre existir uma associação de aplicação quando o ROSE é iniciado, e o ROSE continua sua execução até que a associação seja 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 controle da associação de aplicação é realizado pelo ACSE.

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

8.2 A Interface do DSA

Uma Entidade de Agente do Sistema de Diretório deve comunicar-se ora com o DUA para prestar-lhe os serviços de acesso à base de informação, ora com um DSA remoto e adjacente, através do qual as operações previamente submetidas são transmitidas para prover o acesso à base de informação remota. Tanto a comunicação com o DUA quanto a comunicação com o DSA remoto utilizam os Elementos de Serviço Comum da Camada de Aplicação (CASE), os quais efetuam estabelecimento de conexão e transferência de dados. Os dois tipos de comunicação são realizados através da interface com o CASE. A interface entre o DSA e CASE engloba todas as primitivas de serviços que são oferecidas pelo ACSE e ROSE e são descritas no item 4.1.2.

9 Ambiente e Estrutura de Implementação

O ambiente de implementação adotado para os DUA's será a estação de trabalho SUN com sistema operacional UNIX. Já o ambiente de implementação adotado para os DSA's constitui-se do sistema computacional VAX 8810 com sistema operacional VMS. Ambos os sistemas são multi-usuários e incorporam mecanismos para comunicação e sincronização entre processos.

9.1 A Estrutura de Implementação do DUA na SUN/UNIX

A estrutura de implementação do DUA na SUN/UNIX será composta por três processos que se comunicam através de troca de mensagens.

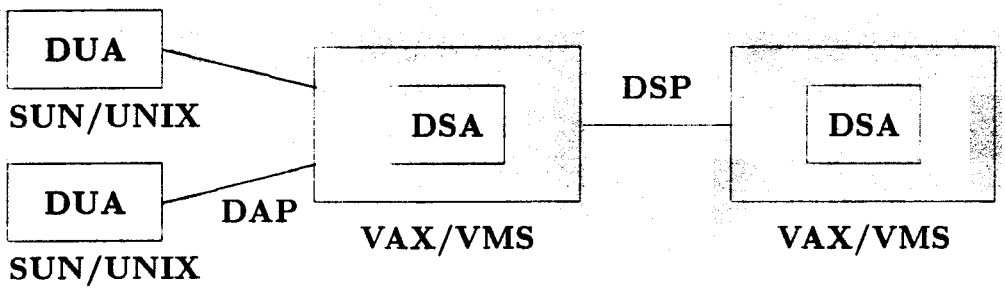

Figure 6: Ambiente de Implementação.

A entidade DUA que contém os módulos DUA-GERENTE e DUA-DAP será implementada como um processo independente do módulo que implementará a interface do DUA com o usuário (DUA-USR). O módulo DUA-USR será dividido em duas partes: a interface do DUA com o usuário propriamente dito e a interface que atua entre o Sistema de Diretório e o correio X400. Estas partes estão organizadas em dois processos distintos. As entidades DUA's implementadas na SUN/UNIX serão desenvolvidas utilizando-se a linguagem de programação C.

O procedimento básico do DUA consiste de um ciclo de interrogações. Este ciclo analisa a comunicação com o módulo que implementa os Elementos de Serviços Comuns da Camada de Aplicação (ACSE e ROSE), analisa a comunicação com o usuário, e ativa, quando necessário, os procedimentos do protocolo DAP ou os procedimentos da interface homem-máquina do DUA, respectivamente.

A comunicação entre os processos na SUN será obtida através da troca de mensagens que são divididas em duas partes: a primitiva propriamente dita e os dados de usuário. A primeira é enviada através de filas de mensagens mantidas em memória compartilhada. Nela estão contidos o deslocamento e o tamanho dos dados de usuário que também são mantidos na memória compartilhada. As filas de mensagens simulam em funcionalidade as caixas postais disponíveis no VMS da DEC onde muitos módulos do projeto foram originalmente desenvolvidos. A divisão da mensagem em duas partes também decorre daí pois o VMS limitava o tamanho máximo de uma mensagem trocada via mailbox acrescido do fato de a comunicação via caixas postais para mensagem de dados ser um processo lento na DEC.

9.1.1 Implementação da Interface DUA-USR no Ambiente SUN

O objetivo da interface é prover ao usuário leigo uma ferramenta para recuperação de informações de propósito geral, sem que sejam necessários conhecimentos detalhados sobre conceitos de diretório.

A interface deverá ser amigável e ter funcionamento razoavelmente intuitivo. Para isso, ela deverá ser desenvolvida utilizando-se janelas e menus de barra, além de ter suporte para uso de "mouse". Assim, o usuário poderá facilmente caminhar pelas opções oferecidas e preencher os campos necessários para a execução das operações requeridas.

O funcionamento de toda a interface deverá ser simples e totalmente ditado pelo usuário. Quando uma operação for selecionada posicionando-se a barra na opção correspondente e pressionando-se o primeiro botão do "mouse", a função que a executará será acionada; novas janelas sobrepostas à janela principal serão apresentadas para que o usuário preencha os parâmetros referentes à operação.

O usuário não precisará preencher todos os parâmetros de uma operação, desde que estes não sejam obrigatórios. Além disso, muitos dos parâmetros já possuem valores "defaults", ficando a critério do usuário modificá-los. Os parâmetros cujo domínio de valores é limitado, por exemplo SIM ou NÃO, serão alterados apenas pressionando-se o botão esquerdo do "mouse" sobre a opção correspondente.

Ao contrário de outras interfaces de sistemas para recuperação de informações, esta oferecerá acesso a uma grande variedade de tipos diferentes de informação. Por isso, a interface DUA de uso geral deverá suprir o usuário com o necessário conhecimento sobre os tipos de informações que ele pode encontrar no Diretório. Esta facilidade será oferecida através da opção de "navegação" pela Árvore de Informações de Diretório (DIT).

Na implementação de janelas, menus de barras e, para controlar todo fluxo de informações entre o usuário e o sistema, será utilizado o pacote de rotinas OpenWindows. Este pacote oferece poderosos recursos para a construção de interfaces baseadas em X-Windows.

Para administrar os arquivos de dados do Diretório e, de um modo geral, todo o sistema de armazenamento de informações em memória secundária, tais como "log" e nome amigável, serão usadas as rotinas da biblioteca stdio da linguagem C.

9.2 A Estrutura de Implementação do DSA no VAX/VMS

A estrutura de implementação do DSA no VAX/VMS é composta por dois processos que se comunicam através de troca de mensagens, são eles: o processo DSA que contém os módulos DSA-GERENTE, MAQEST-DUA e MAQEST-DSA, e o processo OPERATION DISPACHER. A linguagem utilizada para a implementação dos processos será o C com o compilador VAX C.

O procedimento básico do funcionamento do DSA consiste em analisar a comunicação com o módulo que implementa os Elementos de Serviço Comuns da Camada de Aplicação (ACSE, ROSE), e ativa, quando necessário, os procedimentos do protocolo DAP ou do protocolo DSP. Cada análise realizada se refere a uma possível recepção de eventos provenientes do ACSE/ROSE.

Quanto à comunicação entre os processos 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 daquela caixa postal de saída.

Por questões de eficência, 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 processos 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.

10 Considerações Finais

Os estudos, a especificação e a implementação descritas neste relatório foram fundamentados em padrões internacionais baseados na arquitetura OSI. Parte do projeto Inter-X que está sendo implementado na UFRJ seguiu a orientação dos padrões X500 do CCITT para Sistema de Diretório , levando-se em consideração estudos que estão sendo desenvolvidos por grupos de trabalho da própria CCITT.

Este projeto permitirá a formação de recursos humanos em Sistema de Diretório colaborando assim com um dos objetivos do projeto responsável pela formação da Rede Nacional de Pesquisa (RNP).

A proposta de implementação do Sistema de Diretório pode ser utilizada em diversos sistemas operacionais, visto que as soluções a serem adotadas não são particulares aos sistemas operacionais utilizados em nossa instalação, VMS e UNIX.

Apesar disso, existem problemas que devido à sua natureza, só admitem soluções dependentes do sistema operacional, como é o caso da comunicação entre processos. Nestes casos, as soluções serão postas em um módulo separado para facilitar modificações futuras caso o sistema operacional seja trocado.

Quanto ao estágio atual da implementação do Sistema de Diretório podemos dizer que os módulos dependentes dos S.O.'s e os módulos que gerenciam as estruturas de dados estão prontos. Pretendemos ter em maio de 1992 uma versão simplificada do Sistema de Diretório.

11 Referências

[1] Benford S., "Navigation and Knowledge Management within a Distributed Directory System", Proceeding of the IFIP WG 6.5 Working Conference on Message Handling System and Distributed Applications, Costa Mesa, California, U.S.A., 1988, pp 115-134

[2] Huitema C., "Using a standard Directory as a Name Server within the Thorn Project", Proceeding of the IFIP WG 6.5 Working Conference on Message Handling System and Distributed Applications, Costa Mesa, California, U.S.A., 1988, pp 135- 144

[3] Kille S., "The QUIPU Directory Service", Proceeding of the IFIP WG 6.5 Working Conference on Message Handling System and Distributed Applications, Costa Mesa, California, U.S.A., 1988, pp 145-158

[4] CCITT Recommendations X500 Series: The Directory System, 1988, Blue Book, vol VIII Fascicle VIII.8.

[5] CCITT X217 Recommendation:Association Control Service Definition for CCITT Applications.

[6] CCITT X219 Recommendation:Remote Operations:Model, Notation and Service Definition.

[7] Pirmez I., Mendes S., O Diretório X500: Uma solução para o problema de resolução de endereços no MHS", IX Simpósio Brasileiro de Redes de Computadores,Florianópolis, SC, Maio de 1991.