

SERVIÇOS DE APRESENTAÇÃO NA REDE-RIO

LUIZ FERNANDO RUST DA COSTA CARMO

Engenheiro eletrônico - UFRJ
M.Sc. em Engenharia de Sistemas e Computação - COPPE/UFRJ
Analista Consultor do Grupo de Redes do NCE/UFRJ

Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro
Caixa Postal 2324 - CEP 20.001
Telefone: 290 - 3212, ramal 203

SUMÁRIO

O artigo retrata os aspectos de maior relevância encontrados no desenvolvimento de uma implementação dos serviços de apresentação para o projeto REDE-RIO (interligação dos centros de pesquisa do Rio de Janeiro através de uma rede de computadores). Inicialmente são descritos os serviços da REDE-RIO e o seu estágio atual de desenvolvimento. A seguir é realizado um embasamento teórico sobre os serviços de apresentação e descrito as principais características do sistema ISODE (ISO Development Environment). Por último são analisados o estágio atual da implementação e o potencial de uma nova proposta incorporando algumas técnicas sugeridas pelo sistema ISODE.

ABSTRACT

The paper presents the most important features found in the development of an implementation of presentation services for the REDE_RIO project (a computer network between the research centers of Rio de Janeiro). Initially, the REDE-RIO services and the current state of the project are presented. Following, a theoretical introduction for OSI presentation services and the ISO Development Environment (ISODE) are described. Finally, an analysis of the actual state of the implementation is made, together with a discussion of a new proposal which adds some ideas presented in ISODE.

I. Projeto REDE-RIO

Este projeto consiste da implantação dos principais serviços de aplicação existentes (Correio Eletrônico, JTM, FTAM e Terminal Virtual), além das camadas básicas de transporte, sessão e apresentação. Para as camadas inferiores adotou-se a utilização da padronização X.25 do CCITT, objetivando a utilização dos serviços da Rede Nacional de Pacotes - RENPAC, oferecidos pela EMBRATEL. O principal objetivo deste projeto é a implantação do embrião de uma rede nacional de computadores capaz de interligar as universidades e

centros de pesquisas e, principalmente, fomentar a disseminação do Modelo OSI/ISO internamente ao país. A implementação deste projeto é conduzida por quatro entidades executoras, sendo uma delas o Núcleo de Computação Eletrônica da UFRJ. A coordenação e administração do projeto cabe ao LARC.

É importante ressaltar que este trabalho sobre os serviços de apresentação é um esforço isolado preliminar do NCE-UFRJ, não ambicionando a definição de uma proposta definitiva dentro do projeto REDE-RIO.

II. Serviços de Apresentação

II.1 Introdução

Os serviços de apresentação estão relacionados à natureza dos parâmetros dos dados do usuário. Nos serviços de sessão estes parâmetros são uma sequência de octetos que traduzem uma representação específica de algum valor de dado. Para os serviços de apresentação os parâmetros são um ou mais valores de algum conjunto complexo e arbitrário de estruturas de dados (denominado sintaxe abstrata).

ex.1 Vamos supor que um usuário defina três tipos de dados (utilizando uma notação de sintaxe abstrata) denominados "Registro pessoal", "Relatório da firma" e "Catálogo de biblioteca". Um possível valor do parâmetro do dado do usuário da primitiva de transferência de dados de apresentação P-DATA poderia ser uma instância de cada um destes tipos. O nível de apresentação por sua vez determina como transportar esta informação através de uma sequência de octetos e passa estes octetos resultantes para o nível de sessão através da primitiva S-DATA. No local de recebimento o processo é inverso, convertendo-se a sequência de octetos em valores dos três tipos de dados.

Em sistemas de informação distribuídos, aplicações localizadas remotamente possuem um determinado requisito para a transferência de informação (semântica), mas elas não estão envolvidas nos detalhes da representação das suas informações (sintaxe de transferência) ou no processo de codificação durante a comunicação. O nível de apresentação proporciona as funções que definem, através de regras de codificação, a sintaxe de transferência para a informação trocada durante a comunicação.

ex.2 Uma implementação real representa "Registro_pessoal", "Relatório da

firma" e "Catálogos da biblioteca" (ou fragmentos deles) de alguma forma incorporada na linguagem de programação utilizada para escrever o protocolo de aplicação. (Tipicamente isto envolverá apontadores e uma árvore de estruturas gerais emaranhadas, e não uma sequência linear de valores.) O emprego da primitiva P_DATA envolve a passagem de um apontador desta estrutura de dados para o nível de apresentação, para que este possa codificar os valores e utilizar a primitiva S_DATA. (Geralmente o nível de apresentação também irá requerer um apontador para a estrutura de dados ou arquivo que contém a definição dos tipos.)

O nível de apresentação também proporciona um serviço comum às aplicações para a negociação e acordo da sintaxe de transferência que será utilizada durante a comunicação.

Os padrões aplicáveis ao nível de apresentação são:

- ISO DP8822 - Definição dos serviços de conexão orientada a apresentação;
- ISO DIS8823 - Especificação do protocolo de conexão orientada a apresentação;
- Recomendação CCITT X.409;
- ISO DIS8824 - Especificação da sintaxe abstrata notação um - ASN.1;
- ISO DIS8825 - Regras de codificação para ASN.1.

Deve ser enfatizado que em todas as definições de serviço da ISO está claramente destacado que as interações das primitivas de serviço simplesmente fazem parte de uma máquina abstrata. As implementações que reagem e produzem alterações no meio físico de uma forma equivalente estão de acordo com os padrões. Não existem requisitos, explícitos ou implícitos, de que uma implementação OSI deva possuir interfaces visíveis correspondentes aos serviços entre níveis nos sistemas reais.

ex.3 Uma implementação pode ter uma interface de nível de apresentação de acordo com o exemplo 2. Outra implementação, também totalmente de acordo com os padrões, poderia ter chamadas a subrotinas para a codificação de valores dos elementos particulares do tipo de dado, construindo uma string de octetos através de uma sequência de chamadas. A string de octetos resultante poderia então ser passada diretamente do módulo de aplicação para a camada de sessão, sem nenhuma interface identificável correspondente ao emprego da primitiva P-DATA.

II.2 Conceitos da camada de apresentação

A camada de apresentação objetiva a negociação e o estabelecimento da sintaxe de transferência, possibilitando a transferência de tipos de dados (estruturados) através de uma representação em valores codificados. Um tipo de dado pode ter mais do que uma possível sintaxe de transferência, proporcionando um grau maior ou menor de compressão e segurança.

A) Definição de Sintaxe Abstrata

Um conjunto de definições de tipos de dados, utilizando uma notação bem definida, constitui o que se chama de **sintaxe abstrata da informação** que será constituída por valores de quaisquer destes tipos de dados. Portanto, uma notação bem definida poderia ser, em teoria, qualquer uma das existentes nas linguagens de programação desde que possua uma notação para dados estruturados suficientemente rica (ex. Pascal, ADA, etc.) ou uma variante da "Bachus-Naur Form" (BNF).

B) Regras de codificação

Para servir aos propósitos do nível de apresentação, uma notação para a definição da sintaxe abstrata necessita estar acompanhada de um ou mais conjuntos de regras de codificação. As regras irão determinar algoritmicamente, para qualquer conjunto de estruturas de dados definido através da notação da sintaxe abstrata, qual a sintaxe de transferência a ser usada (representação). As notações existentes nas linguagens de programação para estruturas de dados são ineficientes devido a falta de regras de codificação.

C) ASN.1 (Abstract Syntax Notation One)

O CCITT e a ISO definiram uma notação para sintaxe abstrata denominada ASN.1, e um correspondente conjunto de regras básicas para a codificação (Basic Encoding Rules for ASN.1 - BER). A possibilidade do uso de outras notações e regras de codificação é explicitamente reconhecido pelo grupo de trabalho da ISO.

D) Nome das sintaxes

Para a negociação da sintaxe de transferência é necessário o estabelecimento de nomes específicos de forma a:

- Identificar o conjunto de estruturas de dados envolvido (nome de sintaxe abstrata);
- Identificar todas as representações definidas de valores para estas

estruturas de dados (nome da sintaxe de transferência).

II.3 Contexto de Apresentação

O nível de apresentação negocia a sintaxe de transferência para ser utilizada nos valores dos dados associados a um nome de sintaxe abstrata. Esta associação entre um conjunto identificado de valores de dados e sua representação constitui o que é denominado contexto de apresentação. Estes contextos têm um tempo de vida que é limitado a uma única conexão de apresentação, precisando de redefinição quando da interrupção e do restabelecimento da conexão.

Em qualquer momento, vários contextos de apresentação podem ser definidos através da negociação da sintaxe de transferência. Isto forma o que é denominado "Conjunto Definido de Contextos" (DCS).

ex.4 Inicialmente é estabelecido que a definição do tipo de dado "Registro_pessoal" faz parte do mesmo conjunto de definições de tipos de dado (sintaxe abstrata) denominado "DADOS_APOIO". Supondo agora que foi definida uma primeira e simples sintaxe de transferência para a sintaxe abstrata "DADOS_APOIO" denominada "CODIFICACAO_DADOS_APOIO(BASICA)", e uma segunda sintaxe de transferência mais complexa denominada "CODIFICACAO_DADOS_APOIO(COMPRIMIDA)" podemos ter a seguinte situação: O nível de apresentação define que durante uma conexão realizada através de uma linha com alta banda-passante, seja estabelecido um contexto de apresentação para transferência dos valores de "DADOS_APOIO" através da "CODIFICACAO_DADOS_APOIO(BASICA)". Essa linha entretanto pode falhar durante a transferência, obrigando que o restabelecimento da conexão se dê através do uso de uma linha com baixa banda-passante. Neste caso, o contexto de apresentação para a continuação da transferência dos valores "DADOS_APOIO" poderia ser restabelecido utilizando a "CODIFICACAO_DADOS_APOIO(COMPRIMIDA)".

O iniciador de uma conexão de apresentação pode (unilateralmente e opcionalmente) especificar um contexto default através da definição de um nome de sintaxe abstrata e um único nome de sintaxe de transferência. Se este for inaceitável para o receptor, a conexão de apresentação não é estabelecida.

Finalmente, o nível de apresentação reconhece a possibilidade de

identificação do uso dos serviços de sessão "por acordo prévio" Este tipo de acordo envolveria a definição de um contexto default e a omissão de todos os parâmetros da primitiva P-CONNECT, dado que estes valores já seriam conhecidos ("pelo acordo prévio") por ambos os lados. Neste caso o P-CONNECT é nulo. Isto é feito visando a compatibilização com protocolos existentes com a série X.400, que faz uso direto dos serviços de sessão.

II.4 Primitivas de serviço de apresentação

Todas as primitivas de serviço de apresentação correspondem às primitivas de sessão de mesmo nome, com exceção das primitivas P-DEFINE-CONTEXT e P-DELETE-CONTEXT que são as únicas introduzidas pela camada de apresentação. A definição destas primitivas pode ser encontrada na referência [4].

III Ambiente ISODE

O ISODE (Ambiente de Desenvolvimento ISO) foi desenvolvido como uma ferramenta de pesquisa e representa um esforço de promoção para a utilização do modelo OSI da ISO, particularmente nas comunidades de pesquisa "DARPA/NSF Internet" e "RARE". O desenvolvimento das primeiras versões do ISODE foi conduzido pelo Centro de Tecnologia e Pesquisa NORTHROP. Atualmente o grupo Wollongong continua a dar suporte no trabalho ISODE.

Este software foi escrito em "C" para ambientes compatíveis ao "UNIX", e suporta diferentes tipos de serviços de rede. A sua classificação é definida como "non-proprietary".

O objetivo da inclusão desta descrição da solução ISODE para os serviços de apresentação, é que ela representa um excelente exemplo de materialização dos modelos abstratos da ISO. Desta forma a contribuição deste item não se restringe apenas a apresentação de soluções particulares, possuindo também um caráter didático.

O ambiente de desenvolvimento ISO - ISODE é composto das seguintes facilidades:

- Biblioteca LIBACSAP - implementa o serviço de controle de associação OSI;
- Biblioteca LIBROSAP - implementa diferentes estilos de serviços de operações remotas OSI;
- Biblioteca LIBRTSAP - implementa o serviço de transferência segura OSI;
- Biblioteca LIBPSAP - implementa a sintaxe abstrata OSI e os mecanismos de

transferência;

- Biblioteca LIBPSAP2 - implementa os serviços de apresentação OSI;
- Biblioteca LIBSSAP - implementa os serviços de sessão OSI;
- Biblioteca LIBTSAP - implementa um ponto de acesso de serviço de transporte OSI.

Além destas facilidades, ISODE possui outras bibliotecas e programas que traduzem:

- Um exemplo de desenvolvimento de uma aplicação, composto por uma entidade básica FTAM - OSI;
- Um sistema de desenvolvimento de aplicações através de operações remotas;
- Diretório QUIPU.

Neste trabalho, nos concentraremos apenas nas facilidades envolvidas com os serviços de apresentação do ISODE.

III.1 Serviços de apresentação

Atualmente a implementação consiste do subconjunto Kernel da especificação ISO. Isto significa que a biblioteca LIBPSAP2 suporta quaisquer requisitos de sessão que o usuário deseje empregar, negocia o contexto de apresentação durante o estabelecimento da conexão e utiliza a notação de transferência abstrata para a transmissão de estruturas de dados de uma forma independente de máquina.

Todas as primitivas de serviço de apresentação são implementadas através de rotinas que retornam um valor inteiro. Este valor pode significar sucesso ou insucesso. Um programa quando quiser se conectar a outro usando os serviços de apresentação, deve chamar a rotina PConnRequest, que corresponde a ação P-CONNECT.REQUEST.

```
int PConnRequest (calling, called, ctxlist, defctxname,  
                 requirements, squirements, isn, settings,  
                 ref, data, ndata, qos, pc, pi)
```

A descrição destes parâmetros podem ser encontradas na página 27 do volume 2 da referência [9]. Quando a chamada à PConnRequest tiver sucesso (correspondendo ao evento P-CONNECT.CONFIRMATION) serão retornadas informações sobre a conexão no parametro pc (ponteiro para uma estrutura denominada PSAPconnect). Internamente a esta estrutura existe um elemento

denominado `pc-result` que determina o estado da conexão. Se este estiver setado para `PC_ACCEPT`, significa que o estabelecimento da conexão está completo. Caso contrário, a conexão foi rejeitada. Se a chamada à rotina falhar, significa que a conexão também não foi estabelecida, e a estrutura denominada `PSAPabort` (que faz parte da estrutura `PSAPindication`) será atualizada. Uma vez que a conexão foi estabelecida, um descritor de apresentação é utilizado como referência desta conexão. Usualmente este descritor será o primeiro parâmetro passado para as demais rotinas da biblioteca `LIBPSAP2`. O último parâmetro é um ponteiro para a estrutura `PSAPindication`. Se uma chamada a qualquer uma destas rotinas falhar, este será atualizado. Se uma consulta a estrutura `PSAPabort` (está contida na estrutura `PSAPindication`) indicar que houve um erro fatal, significa que a conexão foi fechada e um evento `P-P-ABORT.INDICATION` ocorreu.

III.2 Serviços de codificação das estruturas de dados

O sistema `ISODE` implementa as abstrações sintáticas de apresentação através do intercâmbio de estrutura de dados independentes de máquinas. Para isso são manipulados dois tipos de objetos: "stream" de apresentação e elemento de apresentação.

Um elemento de apresentação é um objeto utilizado para a representação de uma particular estrutura de dados de uma forma independente de máquina. Uma estrutura "Pelement" contém vários elementos, os mais interessantes são descritos a seguir:

- `pe_context` - Contexto de apresentação a que este elemento pertence;
- `pe-class` - Classe do elemento de apresentação (aplicação, universal, contexto específico, etc.), esta classificação é prevista pela `ASN.1` - referência [5];
- `pe_form` - Forma utilizada por este elemento (primitiva ou construída);
- `pe_id` - Código específico da classe de identificação do tipo deste elemento de apresentação;
- `pe_next` - Apontador para o próximo elemento de apresentação em sequência.

A partir da inicialização e leitura dos elementos, `ISODE` fornece um conjunto de rotinas que pode ser utilizado na tradução necessária entre a representação independente de máquina do elemento e os objetos específicos de máquina como inteiros, string, etc. A utilização destas rotinas proporcionam uma interface simples e uniforme entre protocolos.

Um "stream" de apresentação é um objeto utilizado para a leitura e escrita de elementos de apresentação. ISODE fornece um conjunto de rotinas que fazem a manipulação de uma "Pstream" (criação, inicialização, leitura, escrita, deleção ...).

III.3 Sistema de desenvolvimento de aplicações ISODE

A metodologia empregada para a construção de aplicações distribuídas é baseada em dois aspectos chaves: tipos de dados abstratos e operações remotas.

Tipos de dados abstratos:

Um tipo de dado abstrato é um conceito para a descrição de uma estrutura de dados cujo acesso é "bem-definido". Isso desencadeia as seguintes implicações:

- Embora a estrutura de dados deva ter uma representação concreta em um determinado sistema local (ex. uma estrutura em "C"), seu correspondente tipo abstrato de dado é definido de uma forma independente de implementação denominada sintaxe abstrata.
- Um conjunto de regras, definido por uma "notação abstrata de transferência", é usado na transmissão da sintaxe abstrata. Isto permite que a estrutura de dados correspondente ao tipo de dado abstrato seja transmitida sem ambiguidades pela rede. Existem dois tipos de mapeamentos envolvidos:
 - Primeiro, a estrutura de dados é mapeada na sintaxe abstrata. Este mapeamento é dependente da implementação;
 - Segundo, a sintaxe abstrata é mapeada para a sintaxe concreta. É importante que não se confunda o conceito "representação concreta" descrito anteriormente (que está relacionado a linguagens de programação) com "sintaxe concreta" (que está relacionado às propriedades de transmissão).

Operações Remotas:

A utilização de operações remotas é uma técnica bastante conhecida na construção de aplicações distribuídas. Atualmente a tecnologia OSI fornece uma poderosa notação para as interações externas de sistemas. De acordo com o modelo OSI, as características externas que necessitam ser visíveis e homogêneas são completamente especificadas, permitindo a compatibilização de interconexões e evitando restrições desnecessárias.

Uma operação, na sua forma mais primitiva, é uma simples interação do

tipo requisição/resposta. Quando uma operação é invocada, é retornado um resultado para o caso de sucesso, ou um erro para o caso de insucesso, ou ainda uma rejeição para o não desenvolvimento da operação.

Um usuário no desenvolvimento de sua aplicação, cria o que foi denominado módulo de operações remotas, definindo as operações junto com a sintaxe abstrata das estruturas de dados trocadas pelo serviço (através da ASN.1). O programa `rosy` lê a descrição do módulo e produz os correspondentes fragmentos em "C", ou seja, as rotinas que o usuário deve utilizar na requisição da operação pelo executor. A sintaxe do arquivo de entrada pode ser encontrada na referência [9].

O programa `rosy` gera também o módulo de sintaxe abstrata, este nada mais é que o módulo de operações remotas exceto as definições das operações. O que resta é apenas a definição das estruturas de dados através da ASN.1.

O programa `posy` (compilador para especificações abstratas) lê uma descrição do módulo de sintaxe abstrata e gera as correspondentes estruturas em "C" que serão manipuladas pelo usuário. Este compilador gera também um módulo de sintaxe abstrata estendido. Este nada mais é que o módulo de sintaxe abstrata incrementado de algumas diretivas de uso do programa `pepy`.

O programa `pepy` (compilador de elementos de apresentação) lê uma descrição de um módulo e produz um programa "C" que reconhece os objetos descritos no módulo, ou gera estes objetos. A sintaxe de entrada é exatamente correspondente a ASN.1.

Resumindo, têm-se que o `posy` gera uma representação concreta das estruturas de dados especificadas através da ASN.1, e o `pepy` gera os programas que irão fazer a conversão destas estruturas em elementos de apresentação e vice-versa (sintaxe concreta).

IV. Serviços de Apresentação na REDE-RIO

IV.1 Primeira aproximação

Inicialmente não foi prevista a utilização dos serviços de apresentação na REDE-RIO. Adotou-se uma sintaxe de transferência comum, tornando sem efeito os serviços de negociação. Esta sintaxe padrão nada mais era que um transporte das estruturas de dados especificadas através da ASN.1 para uma estrutura definida pela linguagem "C", através da interpretação e cruzamento dos tipos de dados. De acordo com este procedimento não é garantido a interoperabilidade entre uma aplicação da REDE-RIO com uma

aplicação OSI do mesmo tipo e perfil funcional, pois existe uma diferença nos procedimentos para o estabelecimento de uma associação de aplicação (com exceção do MHS).

IV.2 Especificação funcional

Inicialmente é importante distinguir dois subconjuntos de serviços de apresentação. O primeiro é composto por todos os serviços envolvidos na negociação e estabelecimento de uma conexão de apresentação, excetuando-se as transformações nas estruturas de dados. Este primeiro grupo foi denominado de "Serviços comuns de apresentação". Consequentemente, o segundo grupo resume-se às conversões de sintaxes abstratas e codificação/decodificação dessas sintaxes.

A especificação dos serviços comuns de apresentação compreende a definição das unidades funcionais suportadas. A camada de apresentação define duas categorias distintas de unidades funcionais.

A primeira categoria é composta pelas unidades funcionais de sessão. A determinação destas unidades para o projeto REDE-RIO é delimitada pela própria classificação funcional adotada para uma entidade de sessão. Atualmente está disponível o subconjunto de serviços representativos da classe 0, incluindo as unidades funcionais "kernel" e "full duplex". Consequentemente, nesta primeira categoria de unidades funcionais os serviços de apresentação comportará apenas estas duas unidades.

A segunda categoria é denominada de unidades funcionais de apresentação, sendo compreendida por:

- Unidade funcional "kernel";
- Unidade funcional de gerenciamento de contexto;
- Unidade funcional de restauração de contexto.

A unidade "kernel" deve sempre estar disponível e suportará a transferência de informações através das unidades funcionais de sessão selecionadas. As outras duas são opcionais, sendo que a utilização da terceira somente se dará quando da utilização da segunda. A unidade de gerenciamento de contexto, quando selecionada, permite que o conjunto de contextos definido seja alterado durante uma única conexão. Como inicialmente os serviços de apresentação só permitem um único contexto (vide próximo item), a utilização desta unidade fica comprometida e portanto não está incluída na especificação. Como a seleção da unidade de restauração está condicionada a de gerenciamento, esta também não foi incluída na

especificação.

A definição da especificação funcional dos serviços de codificação/decodificação se resume na determinação de uma estratégia a ser adotada pela implementação.

Existem duas condições básicas que devem influenciar nesta decisão. A primeira é a natureza do projeto REDE-RIO e o objetivo da utilização dos serviços de apresentação. O projeto REDE-RIO tem como público alvo os usuários finais de aplicações OSI, portanto não se caracteriza como um ambiente para o desenvolvimento de novas aplicações. O objetivo da utilização dos serviços de apresentação é apenas garantir a interoperabilidade das aplicações da REDE-RIO com as demais aplicações OSI. Estas duas condições nos levam a delinear um perfil de características para estes serviços:

- Se o objetivo é garantir a interoperabilidade, não é necessário a previsão de diversos contextos de apresentação. Conforme determinação da própria ISO, todas as aplicações OSI devem suportar pelo menos o contexto "ASN.1 + BER". Portanto a utilização deste único contexto adicionada dos serviços comuns garantirá os requisitos de interoperabilidade desejados.
- Como não se almeja a criação de um ambiente de desenvolvimento de aplicações, a solução adotada não necessita possuir características genéricas, possibilitando a adoção de soluções simples e particulares ao contexto REDE-RIO.

Inicialmente foram levantadas três soluções.

A primeira consiste em adotar-se os mecanismos descritos para o ambiente ISODE, utilizando dois programas compiladores com as mesmas características descritas para o POSY e o PEPY.

A segunda compreende o desenvolvimento de apenas um único compilador genérico. Este programa seria utilizado em tempo de execução e possuiria as seguintes características: a sua entrada seria uma estrutura de dados concreta definida em "C" através de uma notação previamente definida que corresponderia ao tipos de dados definidos através da ASN.1. A sua saída consistiria de uma sequência de octetos representativa da codificação dos dados através da BER-ASN.1.

A terceira estratégia é o desenvolvimento de um programa compilador, que seria utilizado como ferramenta de apoio para a criação das estruturas de dados pelo usuário. Este programa teria como entrada uma definição de uma estrutura de dados especificada através da ASN.1 e como saída uma definição de uma estrutura concreta em "C". Esta estrutura quando montada pelo compilador correspondente deveria gerar um código correspondente a aplicação

da BER-ASN.1 a definição de entrada. A atualização de valores ocorreria em tempo de execução diretamente nestas estruturas. A figura I exemplifica este processo.

Analisando estas três soluções descritas anteriormente, verifica-se que todas podem conviver conjuntamente em um mesmo ambiente, sem que haja incompatibilidades. Para isso é necessário uma implementação dos serviços comuns de apresentação (provedor de serviços) que manipule como unidades de dados de entrada, estruturas de dados previamente codificadas. Isto permite que o projetista de uma aplicação da REDE-RIO possa optar pela utilização da estratégia que mais lhe convém. Porém é necessário incluir como parte integrante dos serviços de apresentação uma ferramenta "default". Desta forma optou-se por incluir um programa compilador orientado de acordo com a terceira solução. Sendo este uma ferramenta estática, não implicará em alterações substanciais nos procedimentos de transferência utilizados nas primeiras versões de aplicações. Como descrito anteriormente, as primeiras aproximações codificavam suas estruturas diretamente em "C" de acordo com o mesmo enfoque da saída definida para o compilador.

IV.3 Proposta de implementação

A proposta descrita a seguir destina-se ao ambiente VAX/VMS. O projeto REDE-RIO como um todo adotou um processo de desenvolvimento das diversas camadas e aplicações baseado em programas multi-usuários. Com este enfoque, o programa correspondente a um provedor de serviços deve ser definido como um "módulo" que possui interfaceamento com as camadas vizinhas através dos pontos de acesso.

Os módulos por sua vez são compostos por processos que se intercomunicam através de caixas-postais, também utilizadas na representação dos pontos de acesso.

Para a camada de apresentação, são utilizados dois processos denominados AP_TRANS e AP_REC. Estes são responsáveis pela transmissão e recepção de mensagens, respectivamente. A figura II ilustra a composição da camada de apresentação. Os eventos de apresentação correspondem ao recebimento e envio de PDUs nas caixas-postais.

O processo AP_TRANS implementa a máquina de estado do protocolo de apresentação referente aos procedimentos de transmissão definida na referência [3]. Possui como ponto de entrada a caixa-postal AP_TRANS_CP, recebendo mensagens do processo AP_REC e dos processos das entidades de

aplicação. Por sua vez este processo envia mensagens para a caixa-postal AP_REC_CP, para a camada de sessão e para os processos das aplicações.

O processo AP_REC implementa a máquina de estado do protocolo de apresentação referente aos procedimentos de recepção definida na referência [3]. Possui como ponto de entrada a caixa-postal AP_REC_CP, recebendo mensagens do processo AP_TRANS e da camada de sessão. Por sua vez este processo envia mensagens para a caixa-postal AP_TRANS_CP, para a camada de sessão e para os processos das aplicações.

As mensagens trocadas entre AP_TRANS e AP_REC fazem parte dos procedimentos de gerência dos serviços, distribuídos entre os dois processos. É importante destacar que as caixas postais são utilizadas, na maioria dos casos, apenas como elementos de sincronização. Devido aos requisitos de eficiência do sistema, são utilizadas áreas de dados compartilhadas entre os diversos processos dos vários módulos para a efetiva transferência dos dados.

Como definido na especificação funcional, a implementação dos serviços de apresentação ainda incorpora uma ferramenta estática para facilitar a definição das estruturas de dados. Este programa consiste de procedimentos para o mapeamento de uma definição das estruturas de dados em ASN.1 para a definição de uma estrutura na linguagem "C". O principal requisito é que a estrutura de saída deve possuir a codificação referente a aplicação da BER-ASN.1 na estrutura de entrada, quando montada pelo compilador específico. As atribuições de valores ocorrem em tempo de execução. Como este programa possui as características de um compilador cruzado, denominou-se de tradutor sintático. As figuras 1 e 3 exemplificam o tradutor.

V. Conclusões

Pode-se destacar como ponto forte desta implementação, a flexibilidade decorrente da possibilidade de utilização de diversos mecanismos para a codificação dos dados internamente ao contexto ASN.1 + BER.

A restrição de utilização de um único contexto de apresentação inviabiliza o aproveitamento futuro de novas tecnologias otimizadas. Porém entende-se que o esforço necessário para inserir esta mobilidade neste dado estágio de desenvolvimento não justifica a sua inclusão.

Com respeito ao tradutor, verifica-se através da experiência já adquirida, que certas condições não são completamente satisfeitas. Porém de forma alguma estas restrições inviabilizam a sua utilização. O objetivo deste tradutor é apenas de oferecer ao usuário uma ferramenta de auxílio. Os casos

por ventura não satisfeitos podem ser resolvidos através de soluções particulares do usuário. Para algumas das situações não satisfeitas através de uma solução estática (por ex., questões de tamanho) serão fornecidas ferramentas para uso em tempo de execução.

V. Referências

- (1) CHAPPELL, David. A tutorial on abstract syntax notation one (ASN.1). Open Systems DATA TRANSFER, Dec. 1986;
- (2) DAVISON, E.. Upper layer architecture. Open Systems DATA TRANSFER, feb. 1986;
- (3) ISO. Information processing systems - open system interconnection - basic connection oriented presentation protocol definition. 1986. (ISO/DIS 8823).
- (4) ISO. Information processing systems - open system interconnection - presentation service definition. 1986. (ISO/DIS 8822).
- (5) ISO. Information processing systems - open systems interconnection - specification abstract syntax notation one (ASN.1). 1985. (ISO/DIS 8824).
- (6) ISO. Information processing systems - open systems interconnection - specification of basic encoding rules for abstract syntax notation one (ASN.1). 1985. (ISO/DIS 8825).
- (7) LARMOUTH, John. Presentation layer services for open systems interconnection. Open Systems DATA TRANSFER, Aug. 1985;
- (8) MANTELMAN, lee. Upper layers: From bizarre to bazaar. Data Communications, jan. 1988.
- (9) ROSE, Marshal T. The ISO Development environment: User s Manual, vol. 1, 2, 3 & 4, The Wollongong Group, Palo Alto CA USA, nov. 1988.

```

EX DEFINITIONS ::= BEGIN
  Registro ::= [APPLICATION 0] SET
  (
    Classes [0] IMPLICIT ISO646STRING,
    Info    [1] IMPLICIT INTEGER
          {
            Tiposabstratos (0),
            Tiposconcretos (1)
          }
  )
END -- DEFINITIONS

```

Definição de uma estrutura através da ASN.1 (Entrada do TRADUTOR)

REGISTRO	TAM	CONTEUDO				
60	0A					
		CLASSES	TAM	CONTEÚDO		
		A0	07			
				ISO646STRING	TAM	CONTEÚDO
				16	02	" "
		INFO	TAM	CONTEÚDO		
		A1	01	0		

Codificação através do emprego da BER

```

typedef struct
(
  char    tag$registro,
          tam$registro,
          tag$classe,
          tam$classe,
          tag$string,
          tam$string,
          classe[2],
          tag$info,
          tam$info,
          info
) registro$tipo

registro$tipo registro = (60,0A,07,16,02,"",A1,01,0)

```

Condições :

- ISO646STRING foi definido com apenas 2 elementos;
- INT foi definido com apenas 1 byte.

Estrutura em "C" (Saída do TRADUTOR)

FIG. I EXEMPLO DOS ELEMENTOS MANIPULADOS PELO TRADUTORr

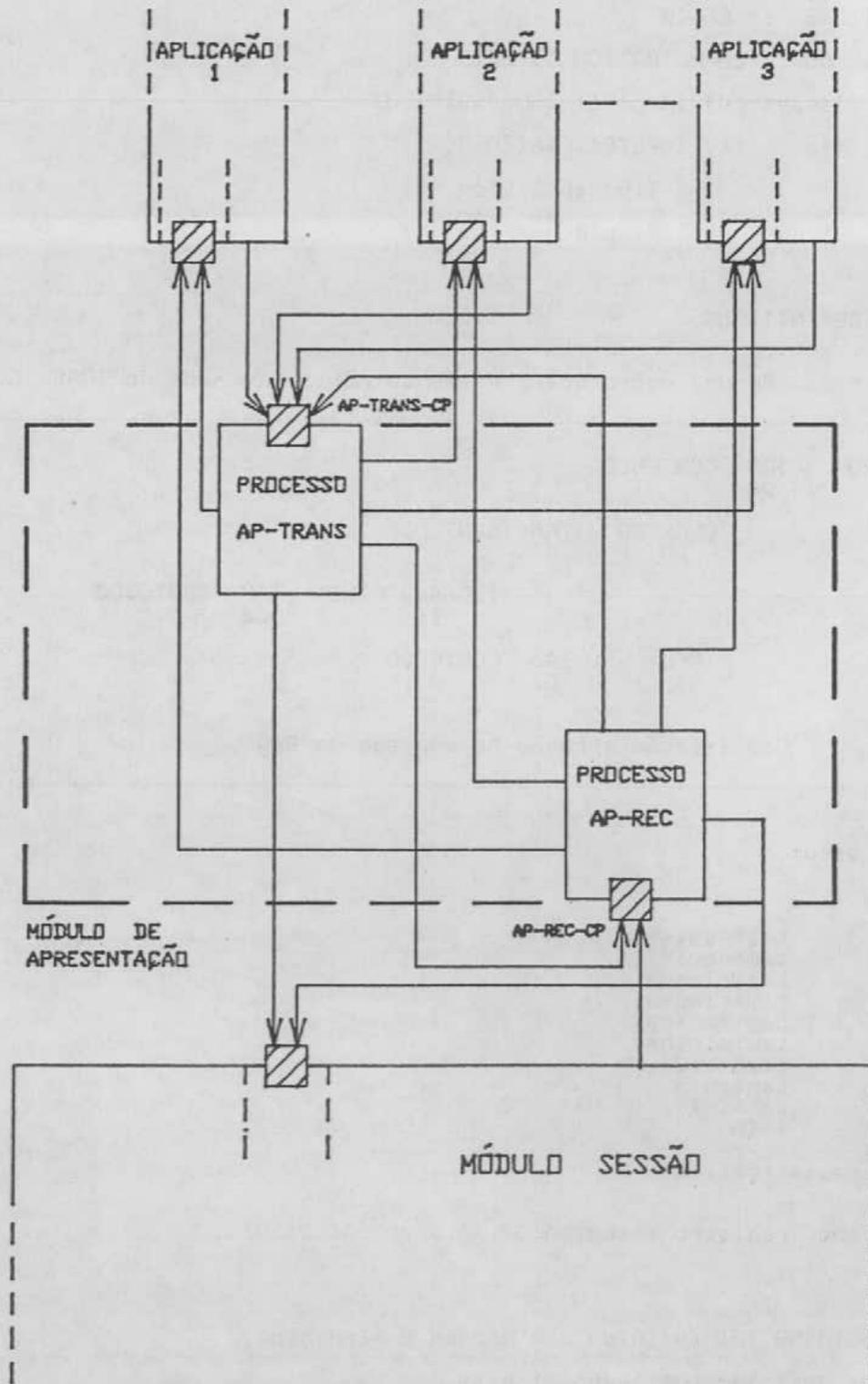


FIG.II - MÓDULO DE APRESENTAÇÃO

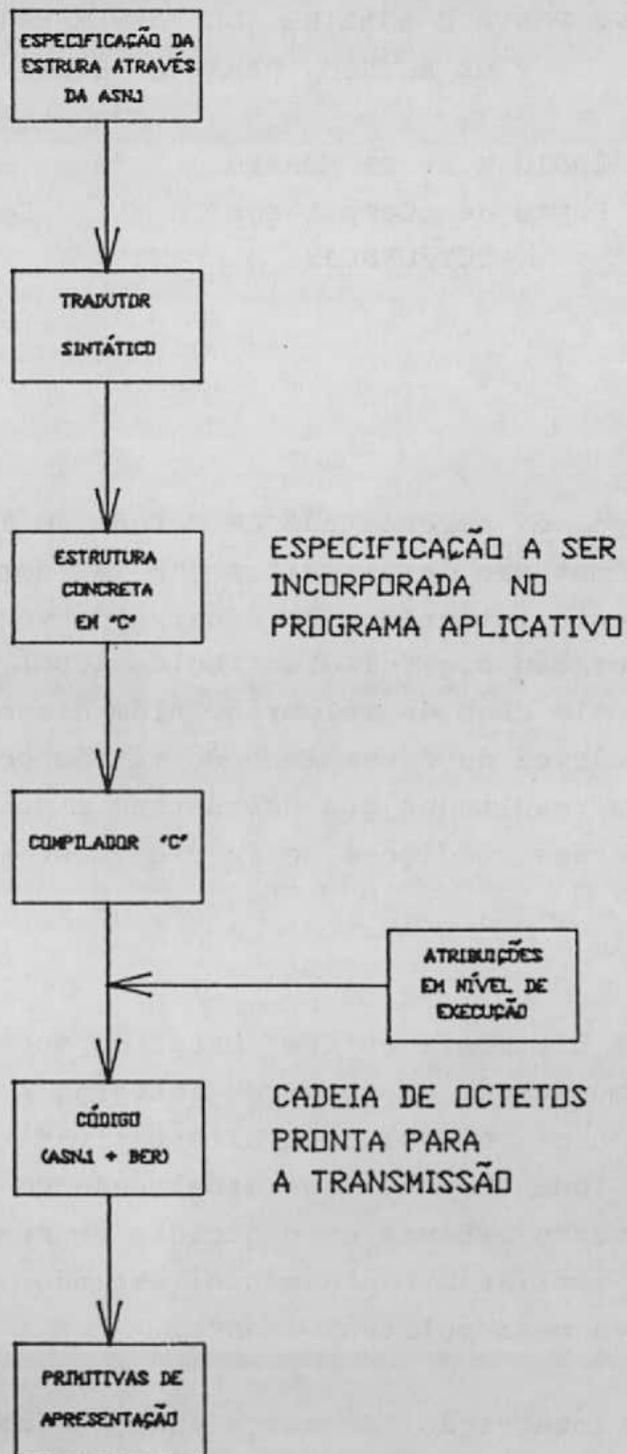


FIG.III - ESTRUTURA UTILIZADA PELO TRADUTOR