

# UM MODELO CONCEITUAL PARA OS COMPONENTES DA CAMADA DE APLICAÇÃO DO RM-OSI/ISO E ASPECTOS DE SUA IMPLEMENTAÇÃO

Flávio Morais de Assis Silva  
Edmundo Roberto Mauro Madeira  
UNICAMP - IMECC - DCC

emails: flavio@dcc.unicamp.br edmundo@dcc.unicamp.br

31 de março de 1993

## Sumário

Este artigo fornece um modelo conceitual para os componentes da camada de Aplicação do RM-OSI/ISO (Reference Model - Open Systems Interconnection / International Organization for Standardization). Este modelo consiste de um refinamento da estrutura apresentada pela ISO para esta camada e é baseado na análise dos conceitos presentes nos protocolos de Aplicação e das operações realizadas por eles em suas especificações. Pretende-se, com este modelo, esclarecer os relacionamentos que há entre os componentes desta camada e estabelecer uma base conceitual para o desenvolvimento de implementações. Aspectos da implementação de um sistema específico baseada nestas idéias são discutidos.

## 1 Introdução

O modelo de referência RM-OSI/ISO (*Reference Model - Open Systems Interconnection / International Organization for Standardization*) [5] é um modelo de sistemas de comunicação conceitual e funcional, definido com o objetivo de servir como uma base para o desenvolvimento de soluções padrões para os problemas que surgem na tarefa de interconexão de sistemas heterogêneos.

Este modelo é estruturado em sete camadas, cada uma responsável por tarefas específicas. A de número 7, chamada de camada de Aplicação, trata de problemas que estão mais fortemente relacionados com as aplicações reais dos usuários de um sistema de comunicação.

Muitos protocolos foram definidos para esta camada, como, por exemplo, o RDA (*Remote Database Access*), para acesso a bancos de dados remotos, o FTAM (*File Transfer, Access and Management*), para a execução de operações em arquivos remotos, o CCR (*Commitment, Concurrency and Recovery*), para o processamento de ações atômicas, e o MMS (*Manufacturing Message Specification*), que suporta a transmissão de mensagens em ambientes da manufatura. Embora estes protocolos tenham funções muito bem definidas, os relacionamentos entre eles se tornaram bastante complexos. Funcionalidades adicionais foram definidas no RM-OSI para modelar estes relacionamentos [6], mas de uma maneira bastante abstrata.

Com este artigo pretende-se definir melhor estas funcionalidades, procurando facilitar a compreensão das funções de cada componente da camada de Aplicação e de seus relacionamentos. O artigo propõe um modelo conceitual para os componentes desta camada, que guia as implementações dos protocolos de Aplicação em um sistema de comunicação desenvolvido na Universidade de Campinas (UNICAMP).

Na seção 2, os conceitos principais relacionados com a camada de Aplicação são introduzidos. Na seção seguinte é feita uma interpretação de alguns aspectos destes conceitos. Na seção 4 apresenta-se como funcionam alguns aspectos comuns ao funcionamento de vários protocolos da Aplicação. Esta apresentação é útil para a compreensão de alguns componentes do modelo proposto. A seção 5 constitui a principal parte deste artigo e é onde o modelo proposto é mostrado. Na seção 6 comenta-se sobre aspectos de como uma implementação guiada por este modelo foi estruturada.

## 2 Definições Conceituais da Camada de Aplicação

Nesta seção, será dada uma visão geral dos conceitos relacionados à estrutura da camada de Aplicação do RM-OSI, como definida em [5] e [6].

### 2.1 Entidade de Aplicação, Elemento de Serviço de Aplicação e Associação de Aplicação

As atividades de processamento de informação para uma determinada aplicação em um sistema real são representadas no modelo RM-OSI por um *Processo de Aplicação*, ou, simplesmente, *AP* (*Application Process*). Pode-se ter, por exemplo, APs para reservas de passagens aéreas ou para transferência de um arquivo de uma máquina para outra. Os APs são os usuários dos serviços fornecidos pela camada de Aplicação.

No modelo RM-OSI os elementos ativos das camadas são representados pelas *entidades*. Na camada de Aplicação há, portanto, as *Entidades de Aplicação*,

ou, abreviadamente, *AEs*, de *Application Entities*. A um AP estão associadas uma ou mais AEs. As AEs constituem todos os aspectos do AP que se referem a capacidades de comunicação. Uma AE só pode estar ligada a um AP. A cada AE está associado o seu tipo, chamado *Tipo da Entidade de Aplicação*, que é uma descrição das capacidades de comunicação da AE.

As AEs incorporam um conjunto de *Elementos de Serviços de Aplicação*, ou, abreviadamente, *ASEs*, de *Application Service Elements*. Os ASEs representam uma coleção de capacidades de comunicação da camada de Aplicação, para a execução de tarefas de comunicação específicas. Há ASEs, por exemplo, que fornecem serviços para o acesso a arquivos remotos (o ASE FTAM) ou acesso a bancos de dados em outros sistemas (o ASE RDA). Utilizando os protocolos da camada de Aplicação, um ASE local e um remoto trocam *APDUs* (*Application Protocol Data Units* - Unidades de Dados de Protocolo da Aplicação). Uma APDU representa informações trocadas entre os sistemas para o controle de um determinado protocolo. Uma AE utiliza as tarefas específicas de comunicação de seus ASEs para a composição de serviços de comunicação mais genéricos. Os ASEs FTAM, ACSE (*Association Control Service Element*) e CCR, por exemplo, podem formar uma AE para a transmissão de arquivos (funcionalidade do FTAM) de forma confiável (funcionalidade do CCR). A função do ACSE será comentada logo abaixo.

Cada uso particular de um AP é modelado por uma *Invocação de AP*, ou, simplesmente, *API*, de *Application Process Invocation*. De modo análogo, o uso das capacidades de comunicação de uma AE em uma ocasião específica é representado por uma *invocação desta AE*. Usa-se a abreviação *AEI*, de *Application Entity Invocation*, para se referir a invocações de AE. Uma AEI modela o agrupamento de um sub-conjunto (ou o conjunto inteiro) das funções de comunicação da camada de Aplicação e a informação de estado relacionada a uma instância de uso destas funções por uma instância de AP. Se um AP representa um programa que transfere um arquivo de uma máquina para outra, cada execução deste AP para a transferência de um arquivo específico representa uma invocação deste AP. Este AP utiliza funções de comunicação, como, por exemplo, o uso de um protocolo específico para transferência de arquivos (o FTAM), que estão representadas por uma AE. Uma invocação desta AE vai representar o uso, pela API, destas funções.

A comunicação entre APIs é modelada no RM-OSI pela comunicação entre suas AEIs. Pelo modelo as APIs se interagem com AEIs locais e estas se comunicam com AEIs remotas, utilizando protocolos. No sistema remoto, da mesma forma, AEIs se interagem com a API remota (figura 1).

Para os protocolos da Aplicação até agora definidos pela ISO, para que uma comunicação entre duas AEIs se processe, uma *associação de aplicação*, ou, simplesmente, *associação* (também mostrada na figura 1), deve ser estabelecida previamente. Uma associação modela o comportamento cooperativo entre as AEIs, para suportar a troca de informação e a coordenação de suas atividades. Representa também um canal entre as AEIs dentro do qual as trocas de mensa-

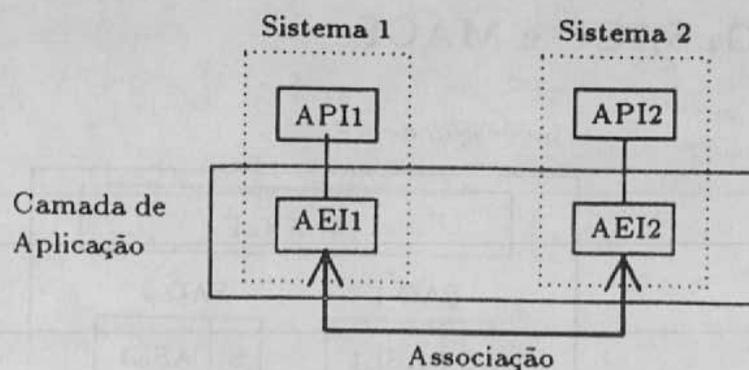


Figura 1: Sistemas, APIs, AEs, Associação e Camada de Aplicação

gens se darão. O ACSE é o protocolo da Aplicação que fornece meios para que associações sejam estabelecidas e liberadas.

Uma invocação de AE pode ter nenhuma, uma ou mais de uma associação em um dado momento.

## 2.2 Contexto de Aplicação

Para que uma comunicação através de uma associação possa ser realizada, as AEs comunicantes têm que estar cientes e seguirem um conjunto de regras que coordene a comunicação através desta associação. Dá-se o nome de *Contexto de Aplicação* [6] a um conjunto de tais regras. Exemplos de aspectos da comunicação a que um contexto de aplicação se refere são:

1. quais AEs serão usados em uma associação. Cada AE é responsável por funções de comunicação específicas e, portanto, para se processar a tarefa de comunicação em uma associação, as funcionalidades de mais de um AE podem ser necessárias;
2. se haverá concatenação de APDUs ou não na associação;
3. regras para estabelecer o seqüenciamento correto de eventos, considerando-se os diferentes AEs utilizados em uma associação;
4. regras para o uso dos recursos de camadas inferiores, como *tokens* e pontos de sincronização da camada de Sessão.

Uma associação tem somente um contexto de aplicação e o contexto exato a ser usado é negociado na fase de estabelecimento da associação, embora esta negociação seja limitada. Uma vez tendo-se determinado um contexto de aplicação, não se pode mais substituí-lo por outro. O contexto, no entanto, pode-se alterar durante o tempo de vida da associação. As alterações que pode sofrer são definidas por regras presentes nele próprio.

## 2.3 SAO, SACF e MACF

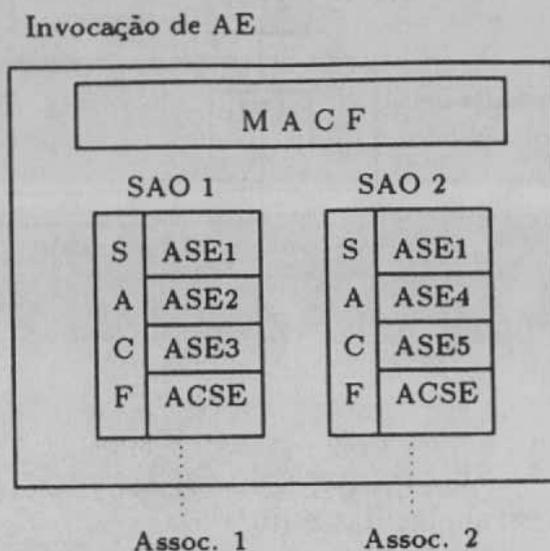


Figura 2: Invocação de AE com MACF, SAOs, SACFs, ASES e associações

A figura 2 mostra como uma invocação de AE é internamente estruturada [6]. Nesta figura:

1. há um objeto chamado *SAO* (*Single Association Object*, Objeto de uma Única Associação) acoplado a cada associação. Uma invocação de AE pode ter nenhum, um ou mais de um SAO em um dado instante, um para cada associação.

Cada SAO é dividido em:

- um conjunto dos ASES que serão usados na associação. O ACSE é sempre um elemento deste conjunto; e
  - o *SACF* (*Single Association Control Function*, Função de Controle de uma Única Associação), que é a parte do SAO responsável pelo controle das interações entre os ASES que participam da associação e do uso dos recursos das camadas inferiores;
2. e há um outro componente, chamado *MACF* (*Multiple Association Control Function*, Função de Controle de Múltiplas Associações), que modela as funções necessárias ao controle das atividades que envolvem mais de uma associação.

Neste artigo será mostrado mais detalhadamente o que faz cada um destes componentes de controle.

## 3 Interpretação dos Relacionamentos entre os Conceitos

Nesta seção serão comentados aspectos de algumas das relações entre os conceitos discutidos na seção anterior.

A um AP pode estar associada uma ou mais de uma AE, cada uma sendo de um tipo diferente. Por [6] é possível ter duas AEs do mesmo tipo associadas a um AP. Esta situação não é considerada aqui, porque a associação de AEs a APs serve para indicar as capacidades de comunicação que o AP utiliza. Estas funções são as fornecidas pelas AEs associadas a ele (AEs modelam agrupamentos de capacidades de comunicação da camada de Aplicação). Uma AE de um determinado tipo já é o bastante para que as capacidades de comunicação que ela representa sejam atribuídas ao AP.

Uma invocação de AP, por sua vez, pode estar associada a muitas invocações de AE de um mesmo tipo ou de tipos diferentes, já que este tipo de associação representa ocasiões de uso das capacidades de comunicação da AE pela API.

As AEs servem para descrever as capacidades de um sistema de comunicação, formadas por capacidades básicas (os ASEs). Por exemplo, não basta saber que um sistema possui implementados os protocolos TP (*Transaction Processing*) e RDA para poder usá-los em uma única transação. É preciso saber se há uma AE que permita usá-los em conjunto. As AEs de um sistema deveriam modelar os tipos de inter-relacionamentos possíveis entre os protocolos da camada de Aplicação que o sistema suporta. Desta maneira não deveria haver uma AE que suportasse tanto associações para transferência de arquivo (FTAM) e acesso a bancos de dados (RDA) se estas associações não se relacionarem. Não há inter-relacionamento a ser modelado. Deveria haver duas AEs, uma para transferência de arquivos e outra para acesso a banco de dados. A estruturação de AEs segundo estas idéias seria modular.

Como uma AEI modela informações de estado relacionadas a uma ocasião específica de uso de uma AE, pode-se concluir que as atividades executadas nas associações desta AEI deveriam ter alguma relação entre si e que esta relação fosse em termos de informação de estado na própria AEI. Com isto quer-se dizer que o estado de uma associação deveria depender do estado de um sub-conjunto das outras associações e que esta dependência fosse em termos de aspectos que a própria AEI deva tratar, não o AP. Se duas associações de uma AEI não se relacionam desta maneira, deveriam estar em AEIs distintas.

Para ilustrar, suponha que um AP tenha que ler dois arquivos, cada um em um sistema remoto diferente, e deva concatenar os conteúdos destes arquivos. Neste exemplo deveria haver um AP associado a apenas uma AE. Esta AE forneceria as funções de comunicação necessárias para ler o arquivo (especificamente, esta AE conteria o ASE FTAM). Seguindo a interpretação acima, a invocação de AP estaria associada a duas invocações de AE, cada uma com apenas um SAO. Cada SAO suportaria a associação com um sistema remoto (figura 3a).

Desta maneira ficaria excluída uma modelagem na qual haveria uma invocação de AE com dois SAOs, um para cada associação (figura 3b). A primeira opção é preferível porque as associações conceitualmente só têm *dependências de estado* no AP, ou seja, as associações só se relacionam em aspectos que devem ser tratados pelo AP, portanto, fora da camada de Aplicação.

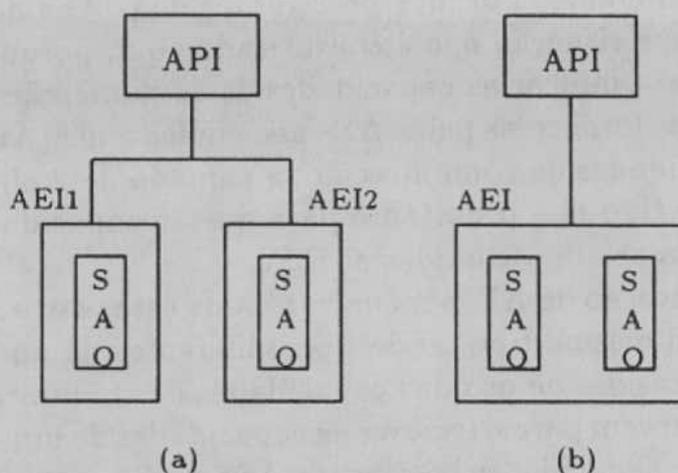


Figura 3: Relacionamento entre APIs e AEIs

O RM-OSI, no entanto, não define precisamente que tipos de relações de estado e coordenação deve haver entre associações de uma AEI, permitindo, com isto, as duas modelagens da figura 3 e ainda outras. Porém a definição de entidade que aparece em [7] e que, segundo uma nota neste documento, passará a ser a definitiva em uma revisão de [5], parece sugerir que as AEs devam fornecer um serviço de comunicação composto coerente, modular, seguindo, assim, as idéias descritas nos parágrafos anteriores.

Um outro exemplo desta situação é quando uma AEI mantém estabelecida uma série de associações com outras AEIs, após o uso delas. Isto evita que tenham que ser novamente estabelecidas associações com estas mesmas AEIs, para os mesmos fins, em uma ocasião futura. Com isto ganha-se em eficiência. O uso futuro de algumas associações pode não ter relacionamentos com o de outras e, portanto, tem-se o mesmo caso anterior. Esta idéia é utilizada nos *pools* de associações dos protocolos TP [11] e RDA [10].

## 4 Características de Procedimentos Comuns aos Contextos de Aplicação

Nesta seção serão discutidos aspectos de endereçamento, estabelecimento de associação e uso de recursos fornecidos pelas camadas inferiores, pois são funcionalidades que influem no modo de se estruturar a camada de Aplicação e que

são usadas por todos os contextos de aplicação.

## 4.1 Endereçamento na camada de Aplicação

Para serem endereçadas as AEs se ligam a *P-SAPs* (*Presentation-Service Access Points* - Pontos de Acesso a Serviços da Apresentação). Os P-SAPs representam, conceitualmente, os pontos na interface entre a camada de Aplicação e a de Apresentação onde as entidades destas camadas trocam primitivas de serviços. Um P-SAP pode estar ligado a no máximo uma AE e uma entidade de Apresentação. A uma AE podem estar ligados mais de um P-SAP. Estas regras de associação fazem com que cada P-SAP identifique uma única entidade de Aplicação e uma de Apresentação. Este relacionamento é dinâmico.

APs e AEs são identificados por *títulos de APs* e *títulos de AEs*, respectivamente. APIs e AEIs são identificadas através destes títulos também, mas somados a *identificadores* de invocação. Em cada instante de tempo há um *endereço de apresentação* associado a cada título de AE. Este endereço identifica o conjunto de P-SAPs ligados à AE (figura 4).

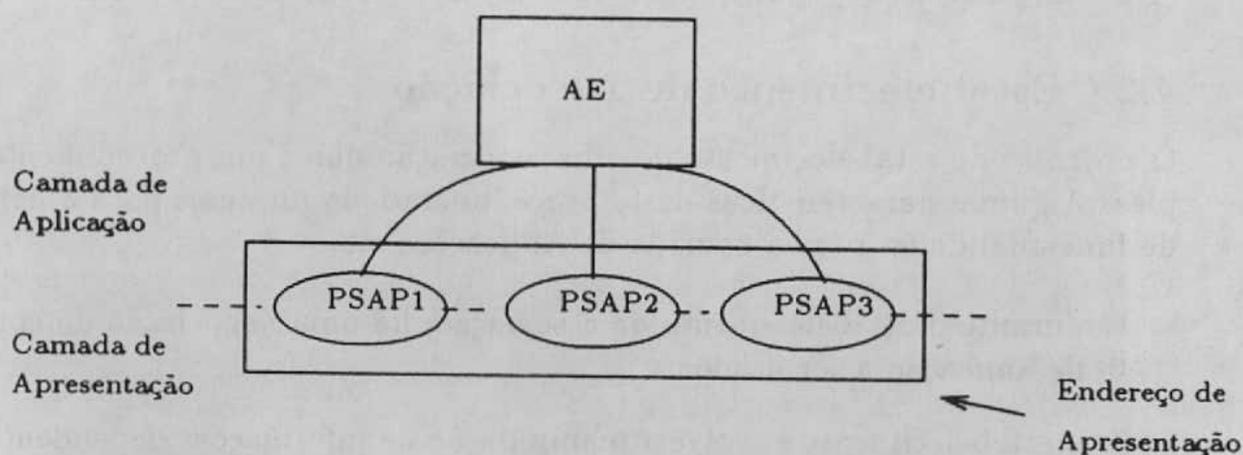


Figura 4: Relacionamento entre AE, PSAPs e Endereço de Apresentação

Note que, uma vez endereçado um P-SAP, as entidades à qual este P-SAP está ligado ficam também endereçadas. Pelo relacionamento entre AEs e APs, endereçando-se uma AE, está-se endereçando também um único AP. Os endereços dos APs serão, portanto, endereços de apresentação.

O endereço de apresentação é usado para endereçar AEs, não invocações de AEs. Na fase de estabelecimento de uma associação, se for necessário, a AEI que requisita o estabelecimento pode fornecer também um identificador de API e/ou um identificador de AEI, para especificar os objetos exatos com os quais se quer comunicar. Um título de AE ou título de AP pode também ser fornecido, para se verificar se a AE e o AP com que se deseja comunicar são os que realmente são acessados através dos P-SAPs. Esta verificação pode ser necessária pelo

fato de as ligações entre AEs e P-SAPs serem dinâmicas e porque endereços de apresentação podem-se tornar inválidos. Informações análogas sobre o sistema que está requisitando o estabelecimento da associação são também enviadas à entidade remota, para o endereçamento no sentido de comunicação inverso.

O RM-OSI assume a existência de uma *Facilidade de Diretório de Títulos da Aplicação*, que, a partir de títulos de APs e AEs, fornece informações de endereçamento utilizadas nas camadas superiores. Há uma outra facilidade de diretório usada na camada de Rede, que fornece informações de endereçamento adicionais usadas nas camadas inferiores.

No RM-OSI as funções de uma camada que tratam de endereçamento são chamadas de *Funções de Diretório*. Na camada de Aplicação estas funções são realizadas pelas AEs no momento de estabelecimento de uma associação. Nesta camada as funções de diretório irão fazer mapeamentos entre títulos de entidades da Aplicação e endereços de apresentação. Servem para determinar, por exemplo, qual é o endereço de apresentação remoto a partir do título da AE. Algumas informações adicionais podem ser necessárias para a execução destas funções. Se estas informações estiverem em sistemas remotos, o protocolo DS (*Directory Services* - Serviços de Diretório) [2] é usado para obtê-las.

## 4.2 Estabelecimento de Associação

O processo de estabelecimento de uma associação não é um procedimento simples. Algumas características deste procedimento, de interesse para a definição de funcionalidades para a camada de Aplicação, são:

1. durante o estabelecimento da associação, há uma negociação do contexto de aplicação a ser usado;
2. o estabelecimento envolve a manipulação de informações dependentes dos seguintes elementos:
  - (a) da API associada à AEI na qual a associação está sendo estabelecida, como, por exemplo, os identificadores da API e da AEI com os quais a associação é para ser estabelecida;
  - (b) do contexto de aplicação que será usado, como valores iniciais para pontos de sincronização, ou como deve ser a posse inicial de *tokens* da Apresentação;
  - (c) da AE como um todo, como quais contextos de aplicação a AE suporta, a qual AP a AE está ligada ou quais identificadores de invocação de AE e AP são válidos;
  - (d) das camadas de Apresentação e Sessão, pois depende de quais unidades funcionais estão disponíveis para serem utilizadas em uma associação;

- (e) da camada de Transporte, pois depende da qualidade de serviço que pode fornecer;
- 3. APDUs de protocolos podem ser enviadas como dados de usuário da primitiva que estabelece a associação (A-ASSOCIATE.req) [1]; e
- 4. pode ser necessário utilizar uma outra associação da mesma AEI para o estabelecimento. Esta outra associação seria usada para se obter informações sobre endereçamento, quando disponíveis em outros sistemas. O protocolo DS seria utilizado para este fim.

O item 2b indica que o estabelecimento de associação implica em controle sobre recursos oferecidos pela camada de Apresentação (ou através dela), enquanto o item 3 implica em que deve haver um controle sobre as interações entre os ASEs que serão utilizados. Estas duas funcionalidades são típicas de SACFs. O item 4, no entanto, indica que este procedimento pode implicar em relacionamentos entre duas associações, o que é uma função típica de MACF. Os itens 2c, 2d e 2e indicam que não bastam apenas informações que se referem unicamente a um SAO ou a uma AEI. Embora 2a também indique isto, as informações indicadas neste item são fornecidas pela API através de primitivas de serviços. Nos itens anteriores as informações teriam que ser conseguidas pela própria AEI.

O item 1 informa que há regras necessárias ao procedimento de estabelecimento de associação que são à parte do contexto de aplicação que for ser estabelecido. Estas regras seriam aquelas que definem o comportamento dos componentes da AEI para o estabelecimento em si da associação, até que se negocie o contexto de aplicação a ser usado. Estas regras incluem a interação com o ASE ACSE.

Em resumo conclui-se que o procedimento de estabelecimento de associação inclui funções de SACF, antes e depois de se determinar o contexto de aplicação para a associação, funções de MACF e trata informações que não se restringem a uma única AEI. Esta conclusão será utilizada posteriormente para situar esta funcionalidade na estrutura da camada de Aplicação.

### 4.3 Uso dos Recursos das Camadas Inferiores

Dá-se o nome aqui de *recursos* a unidades funcionais, atividades, unidades de diálogo, *tokens* e pontos de sincronização da camada de Sessão, unidades funcionais e contextos de apresentação da camada de Apresentação e qualidade de serviço da camada de Transporte. Estes são recursos fornecidos através da camada de Apresentação e selecionados e/ou controlados pela camada de Aplicação.

São nos contextos de Aplicação que estão definidas regras determinando quais são as unidades funcionais das camadas de Apresentação e Sessão necessárias, ou como deve ser a estruturação do diálogo em termos dos pontos de

sincronização. Quando dois ou mais ASEs invocam primitivas de serviços que afetam a posse de *tokens* ou a determinação de pontos de sincronização, devem existir regras também que coordenem este uso, evitando, por exemplo, que um ASE requisiite um *token* que já foi requisitado por um outro.

Quanto a contextos de apresentação, a camada de Aplicação deve controlar, através de interação com a camada de Apresentação, o conjunto dos contextos válidos na associação. Um conjunto inicial é negociado no momento de estabelecimento de uma associação e depois este conjunto pode ser alterado, a partir de solicitações dos APs, utilizando primitivas de serviço dos ASEs. Estas solicitações geram requisições de primitivas da camada de Apresentação para este fim.

O fornecimento de valores para os parâmetros de qualidade de serviço é geralmente feito pelos APs, através de parâmetros correspondentes em primitivas dos ASEs. Estes valores são fornecidos pela AEI quando ela mesma solicita o estabelecimento de uma associação para seu uso exclusivo, como para suportar canais no TP [11].

O SACF é o elemento da camada de Aplicação que aplica regras e armazena informações sobre estes recursos. Pode, por exemplo, requisitar a posse de um *token* quando algum ASE precisar, ou determinar o número seqüencial dos pontos de sincronização. O armazenamento de informações serve, por exemplo, para controlar certos serviços dos ASEs que dependem da posse de um determinado *token*, ou para restaurar parte do estado de algum diálogo, quando uma ressincronização ocorrer, como saber quais *tokens* possuía no ponto no qual a associação será restaurada ou quais eram os contextos de apresentação que estavam ativos no momento.

## 5 Estrutura dos Componentes da Camada de Aplicação

Nesta seção serão mostradas as funcionalidades definidas para os componentes da camada de Aplicação. Esta definição é baseada na análise do funcionamento dos protocolos e procedimentos relacionados que suportam este funcionamento. As idéias gerais foram mostradas nas seções anteriores, mas alguns conceitos serão ainda mais detalhados nesta seção, para que as funcionalidades apresentadas sejam melhor compreendidas.

Deve-se ressaltar novamente que a estrutura mostrada nesta seção é CONCEITUAL, tratando-se de um refinamento das estruturas propostas pela ISO em [6]. Estruturas apropriadas para implementações podem ser obtidas a partir deste modelo.

## 5.1 As AEs

Quando se estabelece uma associação os objetos que estarão se comunicando são invocações de AEs e, não, AEs em si. No entanto o esquema de endereçamento usado no RM-OSI é baseado em SAPs, que são ligados a AEs. Especificam-se endereços de AEs, mas localmente é preciso haver a determinação de uma invocação desta AE para efetivamente tratar a comunicação. Para determinar a invocação, pode-se portanto considerar que a cada AE está associado um módulo, chamado *SELECIONADOR DE AEI*, que seria o responsável por esta tarefa. A figura 5 mostra a estrutura resultante para cada AE. O módulo *SELECIONADOR DE AEI* utiliza o ASE ACSE, pois as informações sobre AEs, AEIs, APs e APIs vêm em APDUs deste ASE. Ele trata a primitiva *A-ASSOCIATE.ind* do ACSE e faz a determinação da AEI. Caso esta determinação não seja possível, este módulo não aceita o estabelecimento da invocação. Em caso contrário o controle do estabelecimento é passado à AEI determinada, que prosseguirá o tratamento do estabelecimento.

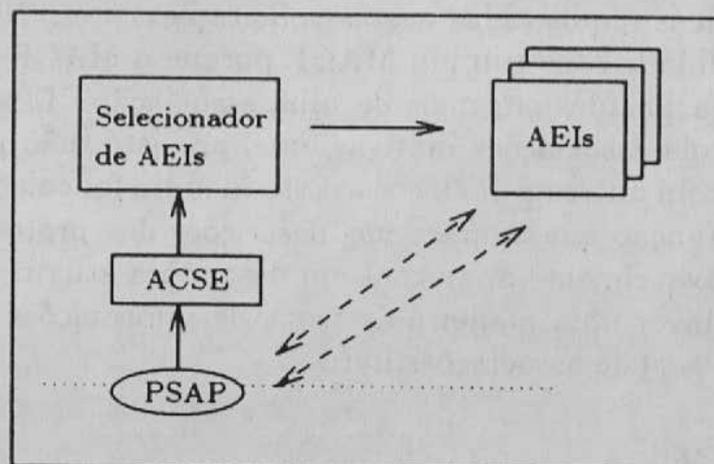


Figura 5: Modelo para AEs

## 5.2 AEIs

De acordo com o que foi comentado anteriormente sobre os tipos de relacionamentos entre as associações de uma AEI, pode-se estruturá-la de acordo com a figura 6.

Nesta figura estão representados os possíveis vários grupos de associações relacionadas, controladas por MACFs, cada conjunto sendo independente dos outros; um módulo chamado *INFO* que representa informações associadas à AEI, como o seu identificador de invocação e o identificador de invocação da API a que está associada; e um conjunto de associações controladas por um módulo chamado *GERENCIADOR DE ASSOCIAÇÕES LIVRES*. Este conjunto forma

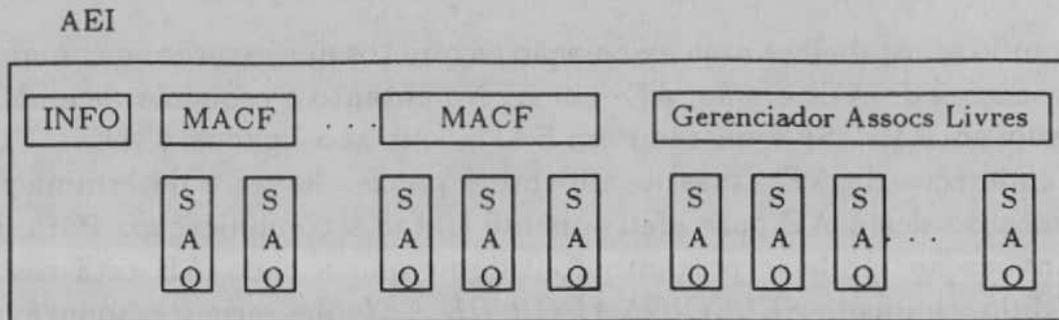


Figura 6: Modelo para AEIs

uma coleção (*pool*) de associações da AEI mantidas conectadas para usos futuros. O módulo gerenciador deste *pool* é o encarregado de controlar estas associações, executando, por exemplo, atividades como atender solicitações de MACFs de pedidos de associações livres, selecionar uma associação que atenda às características requisitadas nestas solicitações, etc.. Não se considerou que esta funcionalidade constitui um MACF porque o MACF modela o controle de atividades relacionadas em mais de uma associação. Esta funcionalidade adicionada controla associações inativas, que, por isto, não podem ter atividades relacionadas com outras associações. Este módulo foi colocado na estrutura por exercer uma função que aparece nas descrições dos protocolos RDA [10] e TP [11] e que provavelmente aparecerá em descrições atualizadas de outros protocolos. Deve haver uma maneira de troca de informações entre os MACFs e o gerente deste *pool* de associações livres.

### 5.3 MACF

Como mencionado na seção 2.3, a ISO define MACF como sendo o conjunto das funções necessárias ao controle das atividades que ocorrem em mais de uma associação de uma AEI. Mais detalhadamente, no entanto, possui funções como:

1. serve como uma interface entre o protocolo e o AP, recebendo de e enviando primitivas para ele e interagindo com os SAOs. O MACF pode gerar uma ou mais primitivas de serviços (enviadas aos SAOs) a partir de uma só primitiva do usuário e vice-versa;
2. é responsável por manter as regras de seqüenciamento das primitivas em mais de uma associação;
3. mantém informações sobre o estado global de atividades executadas em múltiplas associações;
4. executa algumas atividades específicas do protocolo, por exemplo, coletar "votos" no TP [11];

5. trata erros e toma as atitudes necessárias nas diversas associações.

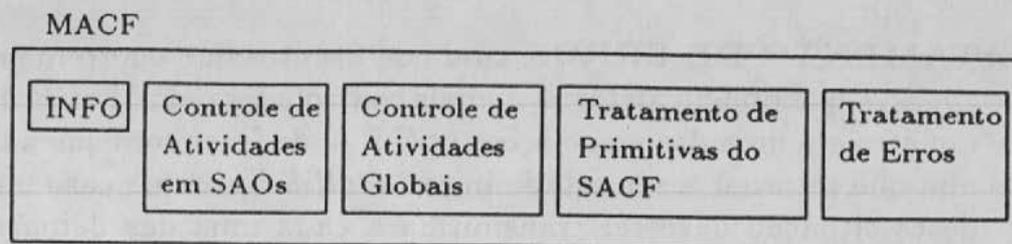


Figura 7: Modelo para MACFs

A função central do MACF é, efetivamente, o controle do estado das diversas associações. As demais atividades que exerce são aquelas que tornam possível este controle ou que dependem de seus resultados. Em termos de funcionalidades, o MACF foi dividido nos componentes que aparecem na figura 7. A descrição destes componentes é:

**INFO** é o que armazena informações necessárias ao ou decorrentes do controle das várias associações, como, por exemplo, variáveis que indiquem o estado da atividade global executada nas associações;

**CONTROLE DE ATIVIDADES EM SAOS** é o responsável por tratar ações que, embora se refiram a uma só associação, dependem do estado da atividade global, controlada pelo MACF. Por exemplo, após se decidir sobre o resultado final de uma transação no TP, diz-se que a máquina de estado do protocolo está no estado *DECIDIDO*. Neste estado as entidades comunicantes não podem mais trocar dados da aplicação. Este módulo seria o responsável, por exemplo, por detectar violações desta regra causadas pela AEI remota e tomar as medidas necessárias. Estas medidas podem envolver ações a serem executadas pelo módulo *TRATAMENTO DE ERROS*, descrito abaixo. Conceitualmente haveria no MACF uma instância deste módulo para cada SAO;

**CONTROLE DE ATIVIDADES GLOBAIS** executa as ações que se referem à atividade global que usa as várias associações. No TP, por exemplo, uma ordem de efetivação da transação atômica é dada pelo AP ao MACF e este se encarrega de distribuí-la nas diversas associações. Conceitualmente haveria somente uma instância deste módulo por MACF;

**TRATAMENTO DE PRIMITIVAS DO SACF** representa o tratamento de primitivas fornecidas pelo SACF, para o controle do SAO como um todo. Um exemplo poderia ser o tratamento de uma primitiva emitida por um SACF para indicar que o SAO em que está não precisa mais ser controlado pelo MACF, ou seja, não faz mais parte da atividade global. Algumas

destas primitivas podem ser também emitidas no sentido do MACF para o SACF, e este módulo trata destas emissões. Conceitualmente também haveria uma instância deste módulo para cada SAO;

**TRATAMENTO DE ERROS** cuida de eventos que ocorrem em uma associação e que podem afetar as demais controladas pelo MACF. Por exemplo, um erro em uma das associações no TP pode fazer com que a transação tenha que retornar a seu estado inicial (*rollback*). Neste caso uma indicação desta situação deve ser transmitida a cada uma das demais associações [11].

Estes módulos se relacionam, fazendo com que atividades exercidas por um deles possam acarretar em que atividades sejam exercidas por outros. A situação comentada no parágrafo anterior serve de um exemplo disto.

## 5.4 O SACF

A figura 8 apresenta uma visão geral dos elementos constituintes de um SAO. Nesta figura pode-se observar os ASEs associados ao SACF, da maneira apresentada em [6]. O SACF aparece dividido em dois componentes: o módulo *ESTABELECEDOR DE ASSOCIAÇÃO* e o *PARTE ESPECÍFICA DE CONTEXTO DE APLICAÇÃO*. O primeiro deles representa as funções necessárias ao estabelecimento da associação, que, uma vez estabelecida, será controlada pelo segundo módulo.

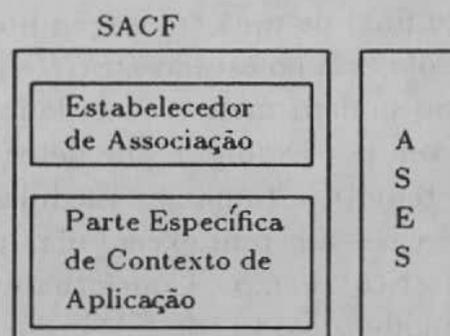


Figura 8: Modelo para SAOs

O módulo *ESTABELECEDOR DE ASSOCIAÇÃO* possui, na AEI que solicita o estabelecimento da associação, funções como coletar APDUs dos ASEs, opcionalmente concatená-las, fornecer valores para alguns parâmetros de primitivas de serviços, obter informações de endereçamento e interagir com o ACSE através de suas primitivas de estabelecimento de associação. Na AEI receptora exerce atividades como verificar se os valores de alguns parâmetros são aceitáveis, interagir com os ASEs para verificar se as APDUs enviadas também

são aceitáveis e enviar uma primitiva de serviço indicando a solicitação ao usuário do SAO (um AP ou um MACF). Controla também o envio da resposta do estabelecimento à AEI inicial.

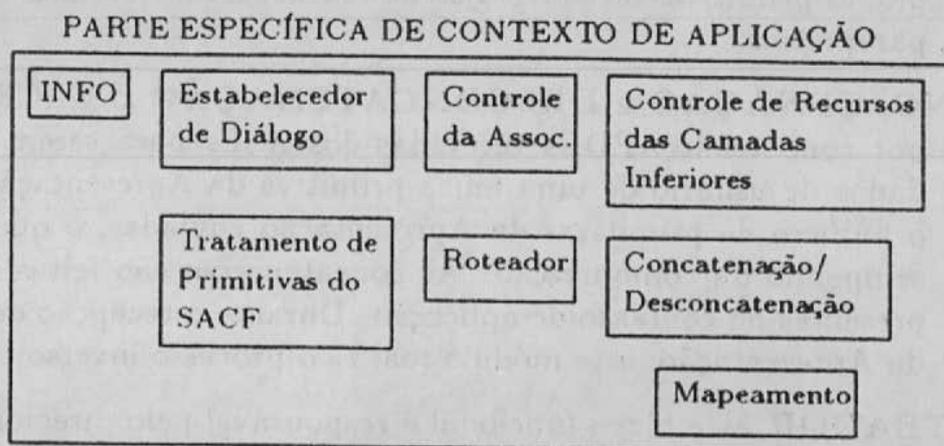


Figura 9: Modelo para a PARTE ESPECÍFICA DE CONTEXTO DE APLICAÇÃO de SACFs

Uma vez determinado o contexto de aplicação a ser usado, o módulo *PARTE ESPECÍFICA DE CONTEXTO DE APLICAÇÃO* passa a controlar definitivamente as atividades no SAO. Este módulo pode ser estruturado, funcionalmente, de acordo com a figura 9. As funções específicas exercidas pelos módulos que aparecem nesta figura dependem do contexto de aplicação negociado, mas de modo genérico são as seguintes:

**INFO** este bloco contém as informações que são úteis para o SACF, como os estados resultantes das atividades na associação, informações sobre primitivas anteriormente recebidas e enviadas ou unidades funcionais utilizadas em cada ASE;

**ESTABELECEDOR DE DIÁLOGO** este módulo é o responsável pelo estabelecimento de novos relacionamentos sobre a associação entre as invocações de AE. Exemplos deste tipo de relacionamentos são os diálogos do TP [11] e RDA [10];

**CONTROLE DA ASSOC.** este módulo é o responsável por manter o estado da associação consistente, analisando os eventos que ocorrem no SAO e determinando seus efeitos no estado da associação. As regras de seqüenciamento combinadas dos eventos de todos os ASEs e a sua sincronização também devem ser observadas por este bloco funcional;

**TRATAM. DE PRIMs DO SACF** trata as primitivas que controlam o SACF como um todo. De acordo com o comentado na seção anterior, o SACF pode receber e emitir primitivas deste tipo;

**CONTROLE DOS RECURSOS DAS CAMADAS INFERIORES** controla o uso de recursos das camadas inferiores, como *tokens*, números de pontos de sincronização e contextos de apresentação. Monta, por exemplo, as primitivas de solicitação de *tokens*, estabelecendo valores para seus parâmetros;

**CONCATENAÇÃO e DESCONCATENAÇÃO** o SACF é responsável por concatenar APDUs recebidas dos ASEs para serem enviadas como dados de usuário de uma única primitiva da Apresentação. Isto diminui o número de primitivas da Apresentação enviadas, o que melhora o desempenho da comunicação. As concatenações são feitas seguindo regras presentes no contexto de aplicação. Durante a recepção de uma primitiva da Apresentação, este módulo realiza o processo inverso;

**ROTEADOR** este bloco funcional é responsável pelo direcionamento de primitivas dos tipos *indication* e *confirm* geradas por ASEs, a partir de APDUs recebidas do sistema remoto, e pela camada de Apresentação ao componente da AEI que deve tratá-las. Este componente pode ser algum ASE ou algum módulo do próprio SACF, que pode, posteriormente, enviá-las ao MACF. Este direcionamento deve ser feito porque as primitivas dos ASEs podem conter APDUs de outros ASEs como dados de usuário ou não. Se houver, o SACF é o encarregado de verificar o tipo destas APDUs e enviá-las aos ASEs adequados. Se não houver, a primitiva pode, por exemplo, ser repassada ao MACF;

**MAPEAMENTO** este bloco tem a função de mapear APDUs (concatenadas ou não) no parâmetro de dados de usuário de primitivas da Apresentação e também preencher os valores de outros parâmetros destas primitivas. No contexto de aplicação há regras especificando que primitiva deve ser usada e como os parâmetros devem ser preenchidos.

O módulo *ESTABELECEDOR DE ASSOCIAÇÃO* executa atividades de alguns destes blocos funcionais. O SACF, em resumo, controla os aspectos de uma associação que dependem exclusivamente das atividades realizadas nesta associação.

## 6 Influências do Modelo Apresentado em uma Implementação

Uma implementação das camadas superiores do RM-OSI/ISO está sendo desenvolvida na Universidade de Campinas (UNICAMP). Esta implementação tem como objetivos fornecer um ambiente didático a ser usado no ensino de redes de computadores, servir como uma base para se ganhar experiência na implementação de protocolos de comunicação e constituir uma plataforma para o desenvolvimento de sistemas de comunicação que utilizem o RM-OSI.

Este sistema, chamado SISDI-OSI (Sistema Didático OSI) é constituído de todas as unidades funcionais das camadas de Apresentação e Sessão, de uma interface que mapeia o protocolo de Transporte classe 4 no TCP, e, na camada de Aplicação, dos seguintes protocolos: ACSE, CCR, ROSE (*Remote Operations Service Element*), TP, RDA, DS, NM (*Network Management*) e MMS. Embora não pertença ao RM-OSI, há ainda implementada a Interface de Programa de Aplicação do MMS [3], definida no projeto MAP/TOP. Para que o usuário possa acessar o sistema, há uma interface que permite que ele submeta APs ou serviços individuais e observe o que acontece nas camadas. Um compilador ASN.1 foi desenvolvido que, a partir de descrições em ASN.1 de PDUs, gera estruturas de dados que podem ser usadas na programação dos protocolos, além de rotinas de codificação e decodificação a serem usadas na camada de Apresentação. Parte do sistema estava implementado em máquinas compatíveis com IBM-PC, utilizando-se um núcleo de tempo real para se ter um ambiente multi-tarefa. Esta versão era chamada de SISDI-MAP [4]. Atualmente o sistema está sendo novamente projetado, agora para o ambiente UNIX, e está sendo desenvolvido em estações de trabalho SUN. A esta versão se deu o nome de SISDI-OSI. O SISDI-OSI conterà as implementações do SISDI-MAP, convertidas ao novo ambiente, e implementações de novos protocolos.

No SISDI-OSI, ASEs, MACFs e SACFs correspondem a processos UNIX, que se comunicam através de mensagens. Cada processo tem uma única fila para receber estas mensagens e simula internamente as suas instâncias de uso em cada associação, utilizando, por exemplo, tabelas.

As idéias discutidas anteriormente e o modelo de funcionalidades proposto (figuras 5, 6, 7, 8 e 9) servem como uma base para a estruturação desta nova versão do sistema (o SISDI-OSI). As únicas implementações no entanto que já estão conformes com a nova estruturação são a do protocolo CCR [8] [9] e de um SACF, que contém, por enquanto, apenas aspectos relacionados ao CCR. Os comentários a seguir se restringem portanto a apenas estas implementações.

A figura 10 ilustra o relacionamento da implementação do CCR com outros componentes do SISDI-OSI. Cada uma das caixas representa um processo UNIX. A caixa intitulada *USUÁRIO* representa os usuários de serviço do CCR. São eles que controlam o estabelecimento da associação, requisitam serviços ao CCR e aguardam as respostas. O módulo mais abaixo na figura simula as camadas inferiores do modelo. Este módulo faz mapeamentos de conexões de apresentação, de modo que se simulam diálogos entre dois sistemas diferentes. As caixas intituladas *CCR* e *ACSE* representam as implementações destes protocolos. No SACF estão implementados apenas aspectos relacionados com o CCR. Para cada contexto de aplicação a ser implementado poderá haver um processo implementando um SACF adequado. Aparecem também na figura 10 os tipos de informações trocadas entre os componentes.

Nos parágrafos seguintes se referirá ao processo *Simulador* como *camada de Apresentação*, pois as interações com este processo correspondem conceitualmente a interações com esta camada. O termo *usuário* será usado para se referir

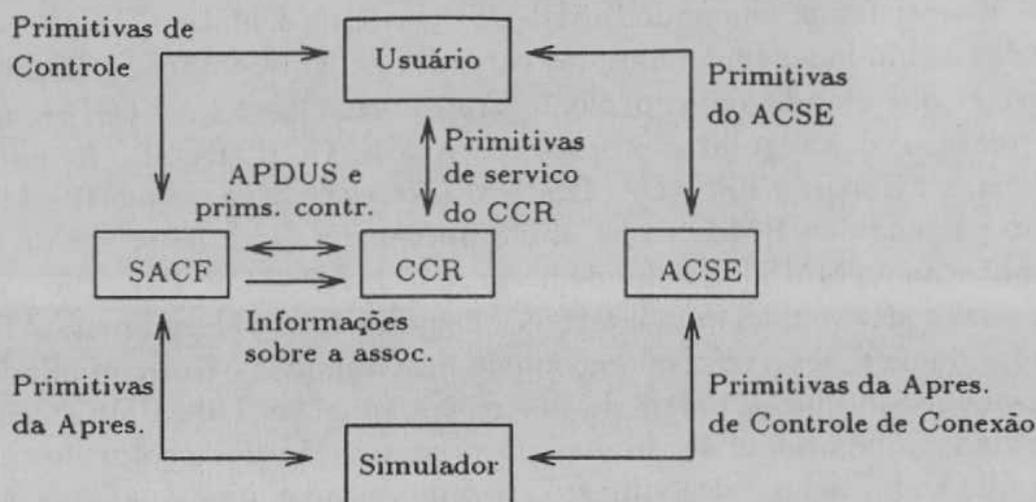


Figura 10: Relacionamentos entre CCR e outros componentes do SISDI-OSI

ao processo *Usuário*.

O estabelecimento da associação é solicitado pelo usuário ao ACSE. Este interage diretamente com a camada de Apresentação (simulador). As primitivas de estabelecimento emitidas pela camada de Apresentação seguem diretamente ao ACSE, que, por sua vez, envia indicações ao usuário. Assim que a associação estiver estabelecida, o usuário passará a trocar informações com o SACF e com o CCR. Observe que com este esquema um contexto de aplicação é negociado durante o estabelecimento da associação e só então se ativa o SACF que implementa este contexto. Neste procedimento o usuário está executando as funcionalidades associadas ao módulo *ESTABELECEDOR DE ASSOCIAÇÃO*, mostrado na figura 8.

Após determinado o SACF correspondente ao contexto de aplicação, o usuário envia a este SACF uma primitiva com informações iniciais de controle, que indica, por exemplo, como será a posse inicial dos *tokens* e qual foi o número inicial combinado para os pontos de sincronização. Algumas destas informações serão passadas aos ASEs (no caso, ao CCR).

Os ASEs serão implementados executando apenas suas tarefas específicas. No caso do CCR isto corresponde a praticamente gerar APDUs a partir de primitivas recebidas do usuário, gerar primitivas a partir de APDUs emitidas pelo sistema remoto (este sistema remoto é apenas simulado) e controlar o seqüenciamento correto dos eventos, segundo sua tabela de estados.

O SACF implementado executará as seguintes tarefas, de acordo com as idéias deste artigo:

- recebe APDUs vindas dos ASEs, por enquanto somente do CCR, e mapeia-as nos parâmetros de dados de usuário da primitiva da camada de Apresentação adequada. Quais são estas primitivas está especificado pelo contexto

de aplicação usado na associação;

- preenche outros parâmetros das primitivas da camada de Apresentação;
- faz algumas concatenações de APDUs. Fazer concatenações é opcional na implementação dos protocolos. No SISDI-OSI somente serão feitas algumas concatenações que fornecem funcionalidades especiais aos protocolos. O CCR especifica duas destas concatenações, que permitem que durante o processo de término de uma ação atômica já se inicie uma nova [9];
- recebe primitivas da camada de Apresentação, obtém a APDU (ou APDUs) que vier como dado de usuário, e a envia ao CCR. O próprio SACF trata também dos parâmetros da primitiva na qual a APDU veio;
- armazena informações sobre a associação, como quais *tokens* possui em um dado momento e informações de estado associadas a pontos de sincronização;
- envia, na AEI que recebeu o pedido de estabelecimento da associação, quando necessário, o título da AE par ao CCR. Isto é um exemplo do fornecimento, pelo SACF ao CCR, de informações sobre a associação.

Observe que o SACF faz o controle dos recursos da camada de Apresentação. O CCR em si não se comunica com a camada de Apresentação.

As primitivas de controle de *tokens* não foram ainda implementadas. Por enquanto o SACF está apenas mantendo informações sobre a posse delas em alguns pontos de sincronização do diálogo. Isto é necessário para tratar res-sincronizações. Informações sobre contextos de apresentação não precisam ser armazenadas, pois não haverá alteração do conjunto destes contextos nas associações do SISDI-OSI.

Em contextos de aplicação mais complexos, o SACF vai realizar outras tarefas, como, por exemplo, controlar estados para a associação como um todo, de acordo com as ações desempenhadas pelos ASEs, ou controlar roteamento, como indicado na seção 5.4. Em contextos de aplicação simples, realiza menos tarefas e pode, inclusive, ser implementado no mesmo processo que se implementa o código de um ASE.

## 7 Conclusão

Este artigo apresenta um modelo conceitual para os componentes da camada de Aplicação que consiste de um refinamento da estrutura proposta pela ISO em [6]. Procurou-se definir funcionalidades para os componentes, de forma a especificar mais precisamente qual a função de cada um deles. O modelo apresentado pela ISO é bastante abstrato na definição destas funções.

Pretende-se, com este modelo, permitir que implementadores de sistemas de comunicação que seguem o modelo RM-OSI estruturem a implementação da camada de Aplicação tendo em mente uma visão geral dos tipos de atividades que são realizadas pelos componentes desta camada e dos tipos de relacionamentos que há entre eles.

A implementação do CCR, comentada na seção 6, foi feita baseando-se nas idéias deste modelo. Obteve-se como resultado uma implementação em que o protocolo pode ser facilmente utilizado em vários contextos de aplicação, que podem inclusive se diferenciar bastante uns dos outros.

O protocolo CCR, no entanto, é simples. Outras implementações do sistema, principalmente do protocolo TP, trarão mais informações sobre a influência deste modelo na implementação do sistema. A implementação do TP já está em andamento.

O modelo mostrou que há muitas funcionalidades atribuídas a SACFs. As implementações destes componentes devem ser bastante cuidadosas, pois envolvem muitos relacionamentos e informações de vários componentes do sistema.

Procurou-se definir um modelo que refletisse situações complexas dos componentes da camada. Em sistemas mais simples, algumas das funcionalidades não precisarão ser usadas.

## Referências

- [1] CCITT. Recommendation X.217 : Association Control Service Definition for Open Systems Interconnection for CCITT Applications. CCITT, 1988.
- [2] CCITT. Recommendations X.500-X.521 : Data Communication Networks - Directory. CCITT, 1988.
- [3] E. R. M. Madeira e Manuel de Jesus Mendes. An Application Interface Model for Communication Software. *IEEE Global Telecommunications Conference - GLOBECOM '90, San Diego, USA, Dezembro 1990.*
- [4] A. Paglioni et al. Sisdi-map: Sistema Didático do Protocolo e da Interface de Aplicação MMS do MAP. *Seminário Franco-Brasileiro em Sistemas Distribuídos, Florianópolis, SC, Setembro 1989.*
- [5] ISO. Information Processing Systems - Open Systems Interconnection - Basic Reference Model - ISO/DIS 7498. ISO, 1984.
- [6] ISO. Information Processing Systems - Open Systems Interconnection - Application Layer Structure - ISO/IEC DIS 9545. ISO, 1988.
- [7] ISO. Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 3 : Naming and Addressing - ISO 7498-3. ISO, 1989.