O ACESSO A VARIAVEIS MMS UTILIZANDO A INTERFACE DE APLICAÇÃO MAP

Verônica Lima Pimentel de Sousa

Departamento de Computação e Automação
Faculdade de Engenharia Elétrica
Universidade Estadual de Campinas
CP 6101 13081 Campinas - SP
E-mail: Veronica@lebion.fee.unicamp.ansp.br

RESUMO

O trabalho apresenta uma experiência de implementação de uma interface de aplicação (API) especificada pelo comitê do projeto MAP, da General Motors.

O modelo da interface é descrito de forma suscinta e a especificação da interface para uso do protocolo de aplicação MMS é referenciada em seus aspectos principais.

Detalhes de implementação da API-MMS são discutidos dentro do escopo de serviços de acesso a variáveis MMS remotas, onde são apresentadas as estruturas de dados envolvidas através de um exemplo simples de leitura de variável.

ABSTRACT

This work presents a practical implementation of the Application Interface (API), proposed by the General Motors Committee - the MAP project.

The interface model is briefly described and the specification for MMS protocol is referred highlighting their major features.

Implementation details of API-MMS are discussed concerning the remote variable access services scope, where data structures are presented in a simple example of variable reading operation.

1. INTRODUÇÃO

Há alguns anos, na área de Redes Locais em Automação Industrial, tem-se voltado a atenção para a viabilidade de uso dos padrões ISO/OSI e do protocolo MMS (Manufacturing Message Specification) na comunicação fabril.

O primeiro aspecto é indiscutível nos dias atuais: o padrão ISO/OSI tem saído dos centros de pesquisas e já é o lema de inúmeros vendedores em vários países. O segundo aspecto refere-se ao protocolo MMS que constitui a gama de serviços visando a comunicação dos

equipamentos programáveis de um ambiente fabril.

O protocolo MMS tem demonstrado poder atender a comunicação dos mais diversos tipos de equipamentos podendo ser implementado em simples robôs, CLP's, NC's, etc como MMS-servidores, como também em estações monitoras que implementam MMS-clientes.

O problema atual tem sido a complexidade e o grande número de atividades (controles, gerenciamento) deixado ao usuário final, ou seja, ao usuário MMS.

Surgiu desse contexto, um projeto assessorado pela General Motors objetivando um padrão aberto, porém simples, para a comunicação de equipamentos heterogêneos num ambiente automatizado de manufatura, visando a migração à interconexão aberta integrada - CIM (Computer Integration Manufacturing). Esses estudos resultaram numa especificação de redes com arquitetura OSI fazendo uso dos padrões OSI/ISO, considerando as necessidades específicas da automação industrial, ao qual se denominou MAP (Manufacturing Automation Protocol) [1].

A idéia de abstração se faz fundamental numa especificação de modelo aberto, tornando possível a heterogeneidade física e a desvinculação de particularidades de um determinado equipamento. Daí o MMS adotar o conceito de VMD (Virtual Machine Device) para amenizar a problemática da diversidade de dispositivos, permitindo que softwares aplicativos não se restrinjam a um dispositivo específico.

Igualmente importante é a flexibilidade no crescimento da automação, tanto para permitir a introdução de novos dispositivos como para expandir a funcionalidade dos elementos existentes a novas integrações. A tendência é o surgimento de "interfaces" para absorver essas adaptações e expansões de funcionalidade.

O comitê MAP considerou essa situação e elaborou um projeto incluindo a especificação de uma interface de aplicação genérica capaz de promover abstração aos softwares aplicativos descrevendo uma visão externa bastante ampla e capaz de mapear adequadamente essa visão nas funções de serviços específicos. A interface para programas de aplicação - API (Application Program Interface) [2], [3], [4] e [5], foi então modelada, considerando alguns aspectos essenciais à comunicação aberta, entre eles: generalidade, independência de hardware, padronização e facilidade de uso.

A especificação de interface para o protocolo MMS baseia-se nas particularidades deste protocolo levando em conta seu modelo (modelo de interação Cliente/Servidor) e os serviços associados.

A API-MMS [5], incorpora um modelo de interação entre usuários MMS (requisitante e respondedor) com o objetivo de troca de informações de forma cooperante. Ela fornece subsídios para a realização das atividades de comunicação e de manuseio dos objetos MMS, dentre eles as variáveis MMS, tornando o papel do usuário menos sobrecarregado. Nos ítens seguintes descrevemos sucintamente o modelo genérico da Interface de Aplicação, sua especificação para o protocolo MMS [5] e apresentamos as funções específicas da interface para o tratamento dos serviços MMS [6] relacionados com o acesso a variáveis remotas, mostrando-se as estruturas de dados envolvidas, seguindo uma sequência de implementação real.

2. MODELO DA INTERFACE DE APLICAÇÃO - API

Uma Interface de Aplicação é uma fronteira entre o ambiente real de aplicações, onde residem os Processos Aplicativos (AP's) e o ambiente padronizado e estruturado de rede, formulado pelo sistema de comunicação. O papel da Interface de Aplicação (API) visa deixar o programa aplicativo independente do sistema de comunicação e do sistema operacional utilizados, como ilustra a figura 1.

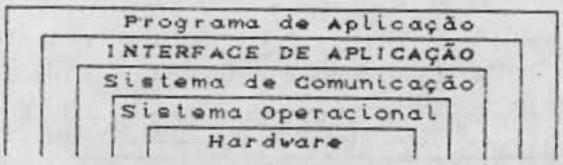


Figura 1 - O papel da Interface de Aplicação num modelo hierárquico

Vários aspectos do modelo podem ser ressaltados, como:

- A característica de generalidade da API: Diz respeito a sua capacidade de operar com múltiplas invocações de AE's (Application Entities). Isso permite total interação e a execução de múltiplas atividades distintas para alcançar uma atividade de âmbito global. A interface pode gerenciar todas as invocações de AE's e suas associações através de funções específicas de gerenciamento de associação.
- O modelo arquitetural constituído de blocos funcionais: Revela a possibilidade de se modular a interface de acordo com o universo de discurso comum a ambos AP's comunicantes. A API pode adequar-se às necessidades, evitando penalidades de espaço/tempo com capacidades adicionais desnecessárias.
- O aspecto de independência de hardware: Imprescindível na abordagem de portabilidade de aplicativos. A API conta com algumas funções de suporte para garantir o isolamento de AP's do hardware e/ou sistema operacional.

Na tentativa de minimizar as implicações provocadas pela heterogeneidade dos sistemas reais a ISO formulou um modelo que envolve estruturas e atividades padronizadas. Dentro desse mesmo modelo foi também projetada a API com a finalidade de suportar a grande variedade de aplicações reais. A arquitetura funcional da API reflete o papel de elo entre o ambiente real do usuário e o ambiente padronizado de rede como descreve a figura abaixo:

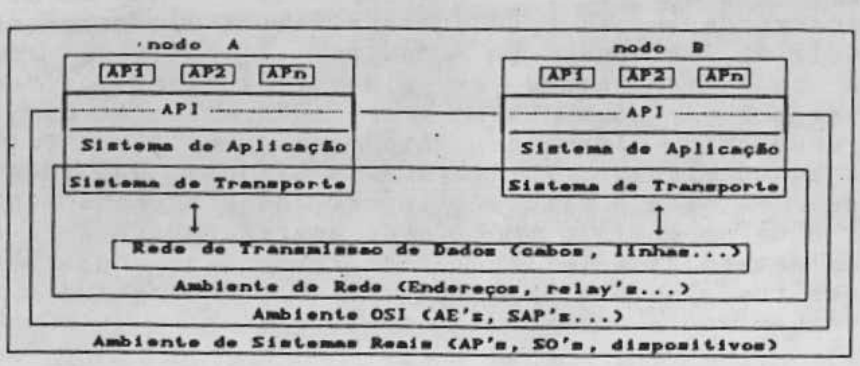
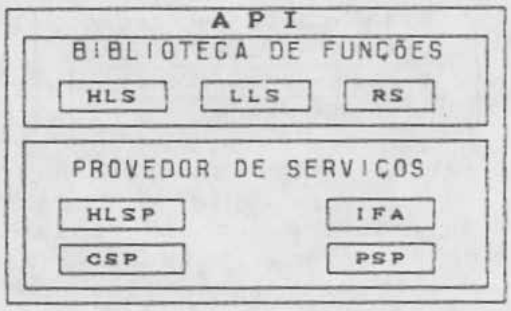


Figura 2 - A API no ambiente OSI e no ambiente de sistemas reais.

A API pode ser esquematizada dentro de duas grandes regiões: uma pertencente ao ambiente do usuário, constituída de uma biblioteca de funções e a outra pertencente ao ambiente de rede, que é o próprio provedor de serviços de rede. Ambas as regiões são constituídas de blocos funcionais, na seguinte forma:



HLS - High Level Service
LLS - Low Level Service
RS - Responder Service

HLSP - High Level Service Provider
CSP - Confirmed Service Provider
IFA - Indication Filter & Arbitrator
PSP - Primitives Service Provider

Figura 3 - Arquitetura da API

Esses módulos funcionais são usados para descrever a funcionalidade que ocorre ao se prover serviços para um usuário. A interação entre os módulos é determinada pelo tipo de serviço solicitado. O modelo de interação da interface é descrito minunciosamente em [2] e analisado sob o enfoque de uma aplicação, o MMS, em [10].

3. ESPECIFICAÇÃO DA API-MMS

Os serviços MMS dizem respeito à comunicação e ao interprocessamento de informações entre dispositivos programáveis do ambiente industrial. Através de modelos abstratos é possível a representação da interconexão de dispositivo heterogêneos e a comunicação entre eles, com suas propriedades estáticas (regras, localização, nomeação, formatos, etc) e dinâmicas (atividades, operações, estados, eventos, interação). A figura do VMD (Virtual Machine Device) retrata o dispositivo com todos os seus domínios e funções. Dentro de um domínio VMD, todos os recursos e serviços a ele vinculado numa determinada instância, representando os objetos MMS (variáveis, eventos, semáforos, invocação de programa, etc). Os serviços MMS atuam sobre um VMD de forma a produzir transações em seus objetos.

O MMS define um modelo abstrato que incorpora as interações entre esses elementos e apresenta suas funcionalidades aos AP's,

adotando a relação cliente/servidor, através da qual realizam-se os serviços requisitados uns dos outros.

Para permitir total interação dos AP's associamos um tipo de primitiva de serviço (request, indication, response, confirm) à uma situação específica.

A API-MMS corresponde a esse modelo de interação, provendo serviços dos tipos requisitantes e respondedores, fazendo uso de suas funções de provedor de serviços representados pelos módulos funcionais: HLSP, CSP IFA e PSP. O Provedor de Serviços de Primitivas exerce o papel de manipulador de primitivas MMS, enviando-as e recebendo-as para/da Máquina de Protocolo MMS.

Além do manuseio de primitivas MMS, a API-MMS conta com outras funções necessárias ao gerenciamento e controles internos, provendo serviços adicionais ao usuário, absorvendo assim algumas de suas atividades. Dentre suas funções podemos mencionar as seguintes:

- simplificar as etapas de estabelecimento da comunicação entre AP's.
- interfacear-se entre usuário e sistema operacional local.
- desempenhar funções auxiliares que diminuam a necessidade de conhecimento, por parte do usuário, da complexa máquina que constitui o protocolo de aplicação MMS.
- formatar para o usuário, etapa por etapa, as primitivas de serviços que ele necessite utilizar.

Todas essas funções são colocadas à disposição do usuário na Biblioteca de Funções, de forma padronizada, tornando mais amigável a utilização do protocolo MMS.

O usuário MMS utiliza os serviços da API-MMS através de chamadas de funções, fornecendo os parâmetros requeridos no formato estipulado. O formato das chamadas de funções segue uma forma padronizada e é obtida com uso de DCB's (Data Control Block). A estrutura dos DCB's contribui para a padronização e embutem as particularidades na descrição de parâmetros opcionais inerentes de uma descrição genérica.

O formato das chamadas de funções API deve seguir o seguinte padrão:

```
Nome_da_função (parâmetros_de_entrada_explícitos, DCB_de_entrada,
                DCB_de_saída, parâmetros_de_saída_explícitos)
```

4. IMPLEMENTAÇÃO DOS SERVIÇOS DE ACESSO A VARIÁVEIS

Para fins de implementação foi considerado um projeto com fins didáticos, em desenvolvimento no departamento DCA/FEE, chamado SISDI-MAP (Sistema Didático - MAP) que incorpora protocolos de aplicação, interfaces de aplicação, processos aplicativos e uma interface de operação, seguindo uma estrutura hierárquica típica OSI. O SISDI-MAP será utilizado como ferramenta de estudo e observação do comportamento de uma rede local MAP [9].

Cada um dos componentes do sistema (protocolo MMS, protocolo ACSE, interface API-MMS, processos aplicativos e protocolos níveis 6 e 7) foi especificado de forma a constituírem módulos autônomos (processos), com suas funcionalidades distintas e recursos

independentes, tendo por base uma estrutura de operação multi-usuário.

O ambiente de implementação para todos esses processos foi um micro-computador IBM PC/AT e a linguagem de programação adotada foi C. Cada um desses processos foi implementado por implementadores distintos, constituindo suas teses de mestrado.

A comunicação entre os processos é feita através de mail-boxes, via portas unidirecionais dedicadas. O núcleo provê ainda o suporte de controle das tarefas, seus escalonamentos, intercomunicação, tratamento de sincronização e gerenciamentos necessários. Na próximaa versão, o SISDI-MAP estará sobre o suporte Unix, e todas essas funções estarão embutidas no próprio sistema operacional.

Como foi mencionado, a implementação da API-MMS consistiu em estruturá-la como um processo independente, autônomo, dentro do modelo hierárquico e modular do SISDI-MAP.

Para compor a API-MMS dentro desse sistema, foi feita a seguinte consideração: restringir a implementação a apenas um subconjunto de funções que compusesse uma entidade funcional coerente e completa em funcionalidade. Dentre os diversos serviços dois conjuntos de serviços foram selecionados:

- Gerenciamento de Conexão, já que esse protocolo opera com comunicação assíncrona, e
- Acesso a Variáveis.

O primeiro conjunto de serviços diz respeito ao estabelecimento, manutenção e término de associações entre AP's [12]; o segundo suporta operações sobre variáveis MMS num VMD associado a um AP [13].

4.1 O ACESSO A VARIÁVEIS MMS

O principal objetivo dos serviços de Acesso a Variáveis visa a realização de operação de leitura e escrita de variáveis. Para realizar uma operação de leitura, por exemplo, são necessárias informações que descrevam a variável, descrevam o método de acesso (se por endereço ou nome) e informações que associem a descrição da variável à descrição dos dados. Essas informações são devidamente utilizadas para converter a representação real (aplicação local) numa forma que possa ser transmitida e reconhecida pelas aplicações de rede. A API contribui nessa tarefa auxiliando o usuário MMS a compor a estrutura completa requerida na descrição da variável e descrição dos dados, através de funções auxiliares que são chamadas pelos usuários de forma sucessiva, sequencial, passo a passo.

Supondo que um usuário local deseja acessar variáveis remotas existentes num certo dispositivo e considerando que a associação a esse elemento já foi estabelecida (utilizando os serviços de gerenciamento de conexão), seria necessário que o usuário utilizasse mais dois tipos de serviços específicos da API-MMS:

- serviços para auxiliá-lo na construção de definições dos parâmetros requeridos na primitiva de Request, e
- o serviço propriamente dito para realizar a leitura da variável ou variáveis em questão (ReadRequest).

Na primeira etapa, as funções auxiliares da API-MMS são utilizadas de forma coerente com os protocolos para especificar tipos

abstratos, métodos de acessos, etc. Exemplificando nosso caso específico, em que uma variável simples deve ser acessada por seu nome (variável nomeada), o usuário faria, antes do pedido de leitura desta variável, uma especificação de acesso a variável e uma especificação que descreva a própria variável. Como isso é feito?

4.2 ESPECIFICAÇÃO DO SERVIÇO

O protocolo para especificação de acesso a variáveis [7] estabelece que seu acesso poderá ser feito ou fornecendo o nome de uma lista de variáveis previamente definida como um objeto MMS, ou fornecendo a própria lista de variáveis que conterá a especificação da variável. Esta especificação encontra-se descrita na linguagem abstrata ASN.1 (Abstract Sintax Notation ONE) e, portanto, deve ser traduzida para uma linguagem concreta. Na primeira versão do SISDI-MAP essa tradução foi feita manualmete para a linguagem C; na próxima versão, um compilador ASN.1 fará essa tarefa de forma automática.

Para ilustrar, apresentamos as definições das estruturas de dados envolvidas numa operação de leitura de variável, através da qual mostramos o mapeamento entre a linguagem abstrata e concreta.

Em ASN.1, temos a primitiva de ReadRequest definida como tal:

```
ReadRequest ::= SEQUENCE
   { SpecificationWithResult          [0] IMPLICIT BOOLEAN,
                                          DEFAULT FALSE
     VariableAccessSpecification      [1] variable accessspecification
   }
```

Em C, traduzimos como uma estrutura:

```
struct {
        Boolean                              spec_with_result;
        Var_access_specification             var_acc_spec;
      } Read_request
```

4.3 CONSTRUINDO AS DEFINIÇÕES

A especificação de acesso a variável remota requer que se forneça ou o nome de uma lista ou o endereço da lista construída pelo usuário contendo as especificações de variáveis.

O tipo "CHOICE" do ASN.1 é traduzido para o C como "UNION". Uniões fornecem um modo de manipular diferentes tipos de dados numa única área de armazenamento. Resulta-se no seguinte:

```
union {
        List_of_variable                     var_list [Max_element];
        Object_name                          list_name;
      } Var_access_specification;
```

Consideremos para o nosso exemmplo que desejamos ler várias variáveis (ex: var1, var2, var3 e var4). A primeira opção será usada.

Uma lista de variáveis pode conter uma ou mais especificações

de variável. Isso significa que múltiplas variáveis podem ser acessadas numa só operação de leitura, através da interface o que minimiza as chamadas remotas na rede.

Vale a pena observar o outro tipo de especificação para o acesso de variáveis MMS: nome da lista de variáveis. Para isso, fornece-se uma especificação de nome de objeto.

Objetos MMS são previamente definidos para que sejam reconhecidos globalmente pela rede.

Dentro da classe de serviços de acesso a variáveis, existem aqueles serviços destinados à definição dos objetos relacionados a variáveis, que são: Define Named Variable, Define Named Type, Define Scattered Access e Define Named Variable List. Este último deve ter sido utilizado para efetuar a associação de uma lista de variáveis a um nome com validade dentro de um determinado escopo MMS. Maiores informações sobre objetos e seus escopos de existências são encontrados em [6].

Voltando ao exemplo e considerando que a especificação de acesso tenha sido feita fornecendo a lista de variável, veremos agora a especificação da lista de variável. Cada elemento da lista deve ser composto por duas informações: a especificação da variável e opcionalmente uma especificação de acesso alternado. Esta última especificação provê um modo de se acessar uma variável composta ou estruturada e seus elementos podem ser ora um vetor (array) ou parte de um vetor, ora uma estrutura ou elementos de uma estrutura, garantindo o acesso parcial a variáveis compostas, onde as definições se embutem umas nas outras, formando muitas combinações.

A lista de variáveis seria então definida como segue:

```
struct {
          Var_specification          *var_spec;
          Boolean                    alter_acc_valid;
          Alternate_access           *alter_access;
        } List_of_variable;
```

Continuando, veremos agora o protocolo de especificação de variável. Uma variável MMS pode ser especificada fornecendo ou seu nome, ou seu endereço, ou uma descrição de variável ou uma descrição de acesso disperso. Essa definição em C é uma escolha entre 4 tipos. Sendo assim, o tipo union é utilizado:

```
union {
          Object_name                var_name;
          Address                    var_address;
          Var_description            var_descr;
          Scattered_access_description  *scat_acc_desc;
       } Var_specification;
```

Para concluir as definições do nosso exemplo, a primeira opção seria selecionada, fornecendo uma especificação de nome de objeto MMS. Como foi explicado anteriormente, uma requisição de definição de variável nomeada já deve ter sido realizada (Define Named Variable). No caso do exemplo, em que são quatro (4) as variáveis a serem lidas, são necessárias quatro definições da estrutura exposta acima (Var_specification).

4.4 OUTRAS DEFINIÇÕES

Bom, mas nem sempre se deseja acessar a variável simples e associada a um nome. O protocolo de acesso a variáveis MMS conta com a possibilidade de se acessar qualquer tipo de variável complexa e com diversos modos de acesso.

A interface MMS deve garantir facilidade do uso desse protocolo com suas mais diversas possibilidades.

A API de fato conta com funções de definições de tipos complexos, construções de listas, definições de variáveis dispersas, ou seja, variáveis formadas por múltiplas definições fisicamente dispersas, definições de variáveis selecionadas, ou seja, variáveis obtidas de partes de uma variável mais estruturada, etc.

O resultado pode ser uma estrutura de definições aninhadas englobando as definições mais internas, como ilustra a figura 4.

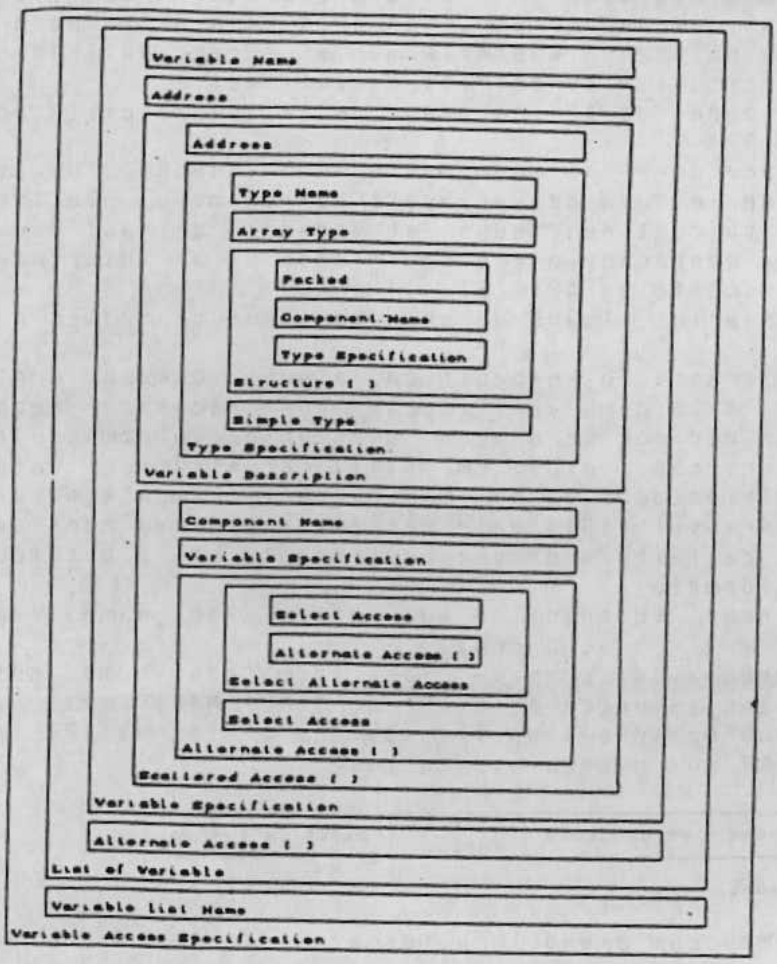


Figura 4 - Definições aninhadas no MMS

Obviamente, definições previamente definidas são utilizadas na construção de novas definições.

4.5 CONSTRUÇÃO X INTERPRETAÇÃO DE DEFINIÇÕES

Correspondendo a construções de definições, feitas pelo usuário (requisitante), outras funções da API-MMS são utilizadas pelo AP-remoto (respondedor) de forma a decompor e interpretar as definições recebidas. Dessa forma a API oferece funções de interpretações de definições MMS, através das quais o usuário interpreta as associações, os métodos de acessos, as especificações de variáveis, os tipos associados, etc.

Como exemplos dessas funções, citamos algumas delas mostrando a correspondência:

mm_new_type ..... mm_interpret_type
mm_make_list ..... mm_extract_elem_list
mm_new_data ..... mm_interpret_data
mm_new_vaccess ..... mm_interpret_vaccess
etc.

Toda a explanação do ítem anterior refere-se à etapa de "preparação" numa operação de leitura/escrita de uma variável, tanto no AP local como no remoto. Nos AP's essa etapa pode ser feita inicialmente uma só vez e posteriormente serem aplicadas inúmeras requisições de leitura e/ou escrita da variável.

E qual o papel da API na etapa de leitura/escrita propriamente dita da variável?

A interface opera em modo assíncrono. Quando o usuário insere uma requisição de leitura de variável, por exemplo, ele imediatamente fica liberado para continuar suas atividades de aplicação. Ao se completar toda a operação, ele é notificado pela interface, atraves das funções de suporte da API [3].

São necessários alguns gerenciamentos e controles para efetuar tal função.

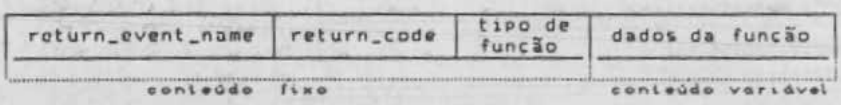
Numa implementação específica alguns desses controles são decisões locais, tais como verificação dos recursos requeridos, o gerenciamento de pedidos pendentes, a comunicação com usuário e com o provedor de serviços, etc. Em [11] encontram-se detalhadas as decisões de implementação da API-MMS. Especificamente para o serviço de leitura de variável (13), vale ressaltar algumas considerações.

A função de leitura da variável consta na biblioteca da API com o seguinte formato:

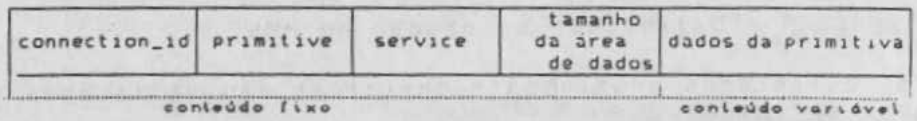
```
mm_vread (connection_id, return_event_name, read_i_dcb,
          read_o_dcb)
```

Esse padrão de chamada foi formatado numa estrutura de mensagem que transita entre AP e API no SISDI-MAP.

Há 2 tipos de mensagens fluindo nas portas da API:
a mensagem API-AP com o seguinte formato:



a mensagem API-MMS com o seguinte formato:



A chamada da função mm_vread fica embutida na mensagem API-AP com seu conteúdo:

```
struct {
        Connection_id                   connection_id;
        Vread_i_dcb_type                vread_i_dcb;
        Vread_o_dcb_type                vread_o_dcb;
      } Mm_vread;
```

Ao receber a mensagem do usuário, a API faz as verificações sobre a validade dos parâmetros, verifica o contexto em que foi inserido o pedido e se OK enfilera-o numa fila de pedidos pendentes. Nesta implementação, assumimos que o usuário deve alocar previamente espaco para obter sua resposta que virá numa primitiva de "response", via mensagem API-MMS e será mapeada no DCB de saída do tipo Vread_o_dcb_type, passado pelo usuário.

Se nesses procedimentos iniciais, não houver erros, a API faz o mapeamento dos parâmetros de entrada do usuário numa primitiva de Read Request, envia através do bloco de mensagem API-MMS ao provedor de serviços MMS e aguarda. Antes de enviar, ela gera um identificador (invoke-id) com o qual identificará na fila de serviços confirmados o serviço solicitado.

Na chegada da confirmação, a API mapeia convenientemente o resultado da solicitação. Caso haja algum erro ou a associação seja desfeita, o usuário é notificado com códigos de erros que retratam o tipo de ocorrência.

A utilização das funções e a interpretação dos códigos de retorno são as principais preocupações de um usuário que utiliza a interface adequadamente. A API-MMS torna-se compensadora quando, numa aplicação real, certas operações são repetidamente utilizadas, como no caso de leitura/escrita de variáveis.

5. CONCLUSÃO

A utilização de uma interface abrangente, bem definida e de fácil utilização é justificada quando num ambiente aberto, equipamentos e aplicações dos mais diversos vendedores são possíveis de se interconectarem.

A padronização garante portabilidade e clareza nos sistemas adjacentes (aplicações e software de comunicação).

A API-MMS demonstrou ser de fácil assimilação na visão dos programas aplicativos, uma vez que o formato de chamadas de funções é bastante aceito no meio de sistemas computacionais. A funcionalidade da interface também é bastante razoável: a sobrecarga de tarefas deixadas ao usuário final até então pode agora ser dividida com um outro elemento que cuida de alguns gerenciamentos e controles.

Porém a ampliação da arquitetura RM-OSI sugerida pela inclusão de mais uma "camada" acarreta todas as implicações resultantes em termos de overhead, tempo de resposta e gerenciamentos redundantes.

Quanto ao serviço de Acesso a Variáveis, a utilização da interface ameniza a pesada tarefa de especificar variáveis complexas. Isso porque a API incorpora diversos tipos de definições parciais que

esclarecem o protocolo de especificação adotado pelo MMS. Ao se tratar de "variáveis de rede" estas devem ser descritas de forma abstrata e inevitavelmente requerem todas essas definições, as quais garantem sua desvinculação do equipamento em que estão implementadas.

Esperamos que a contribuição de experiência de implementação venha auxiliar futuros implementadores na tarefa de compreensão do conteúdo de protocolos, especialmente o MMS, e esclarecer os pontos deixados em aberto como "local matter" nos padrões OSI/ISO.

REFERÊNCIAS

[01] GM - "Manufacturing Automation Protocol", Versão 3.0, Julho 1987.

[02] GM-MAP/TOP - "Application Interface Model and Specification Requirements", Junho 1988.

[03] GM-MAP/TOP - "Application Interface Support Functions", junho 88.

[04] GM-MAP/TOP - "Connection Management Interface Specification", Junho 1988.

[05] GM-MAP/TOP - "MMS Application Interface Specification", Junho 1988.

[06] ISO/DIS 9506 "Manufacturing Message Specification. Part 1: Service Specification". Draft 6. Maio 1987.

[07] ISO/DIS 9506 "Manufacturing Message Specification. Part 2: Protocol Specification" Draft 6. Maio 1987.

[08] ISO/DIS 8824 "IP-OSI: Specification of Abstract Syntax Notation One (ASN.1)". Agosto 1986.

[09] Paglioni, A.; Correa, J.N.; Jacintho, D.C.A.; Lima, J.M.S.; Madeira, E.R.M.; Zabeu, M.C.; Fernandes, I.A.; Sousa, V.L.P.; Mendes, M.J.: "SISDI-MAP: Sistema Didático do Protocolo e da Interface de Aplicação MMS do MAP". Seminário Franco-Brasileiro em Sistemas Distribuídos. Florianópolis, set 1989.

[10] Madeira, E.R.M; Correa, J.N.; Sousa, V.L.P.; Mendes, M.J. "Modelamento de Interfaces de Aplicação e Exemplo de Implementação para protocolo MMS". Anais do VIII SBRC. Campinas, outubro 1990

[11] Correa, J.N.; Sousa, V.L.P.; Madeira, E.R.M.; Mendes, M.J. "Aspectos de Programação e Uso do MMS", Anais do 1o. Seminario sobre Redes de Comunicação Industrial. Sobracon. São Paulo, setembro 1990.

[12] Correa, J.N. - Aspectos de Implementação da Interface dos Programas de Aplicação para o protocolo MMS e seus Padrões Associados: Gerenciamento de Conexão e exemplo de aplicação", UNICAMP/FEE/DCA, Julho 1990.

[13] Sousa, V.L.P. - "Aspectos de Especificação e Implementação da Interface de Aplicação para o MMS: Serviços de Acesso a Variáveis", UNICAMP/FEE/DCA, Julho 1990.