

## Análise da Utilização das API XOM/XMP no Desenvolvimento de Aplicações de Gerência para a Rede de Telecomunicações e Computadores

*Patrícia V. Corrêa Bicalho*

TELEMIG

Belo Horizonte, Minas Gerais

e-mail: patricia@sis.dcc.ufmg.br

URL: <http://www.sis.dcc.ufmg.br/~patricia>

*Benedito Luís Fayan*

CPqD – TELEBRÁS

Campinas, São Paulo

e-mail: fayan@cpqd.br

*José Marcos Silva Nogueira*

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

30161-970 CP 702 Belo Horizonte, Minas Gerais

e-mail: jmarcos@dcc.ufmg.br

URL: <http://www.dcc.ufmg.br/~jmarcos>

### Resumo

Este artigo apresenta uma análise da utilização das API XOM/XMP no desenvolvimento de aplicações de gerência, abordando principalmente os aspectos de implementação e portabilidade das aplicações que utilizam essas API para acesso a infra-estruturas de comunicação. Essas API estão sendo consideradas padrão *de facto*, sendo oferecidas pela maioria das plataformas de aplicações de gerência comerciais. Este trabalho é resultado de experiências práticas com a implementação de aplicações que utilizam essas API, realizadas em paralelo por dois grupos. De um lado a TELEMIG e o Departamento de Ciência da Computação da UFMG, que visam dotar o SIS (*Sistema Integrado de Supervisão*) de uma interface TMN (Q3). De outro lado o CPQD TELEBRÁS, que possui um laboratório TMN especialmente voltado para o estudo e análise dos aspectos de implementação da arquitetura TMN nas empresas operadoras de telecomunicações.

### Abstract

This paper presents an analysis of the use of XOM and XMP API's in the development of management applications addressing mainly implementation and portability aspects. These applications use these API's to access the communication infrastructure. These API's are being considered *de facto* standards, and are made available in many commercial application management platforms. This work results from practical experiences in the implementation of applications by two groups. One is the company TELEMIG and the Computer Science Department of Universidade Federal de Minas Gerais which have the goal to improve a system to integrate supervisory systems called SIS with TMN interfaces (Q3). The other is the CPQD TELEBRÁS which has a TMN laboratory that studies and analysis the implementation aspects of TMN architecture applied to telecommunications enterprises.

## 1 Introdução

A contínua evolução dos sistemas de telecomunicações deu origem a um ambiente de rede bastante heterogêneo devido à variedade de fornecedores de equipamentos e produtos que são normalmente gerenciados por sistemas proprietários e incompatíveis entre si [FIFL93].

Neste ambiente tão heterogêneo e disperso, em um cenário cada vez mais competitivo, as empresas almejam implementar gerência integrada de redes e serviços, que visa principalmente melhorar a qualidade do serviço prestado e reduzir custos operacionais, através do rápido provisionamento e recuperação de falhas da rede.

Um dos requisitos para se implementar a gerência integrada de redes e serviços é que os sistemas de gerência ou aplicações de gerência possam trocar informações entre si, de modo a possibilitar um fluxo contínuo e automatizado na realização de atividades envolvendo múltiplos sistemas e aplicações, para que se possa obter uma visão integrada fim-a-fim dos serviços [RAM93].

A fim de prover uma arquitetura organizada para a interconexão de sistemas ou aplicações de gerência, através de interfaces padronizadas que incluem protocolos e mensagens, o ITU-T definiu a arquitetura TMN (*Telecommunications Management Network*) para a rede de telecomunicações.

Entretanto, não só o ambiente de telecomunicações tornou-se mais complexo. O ambiente computacional também evoluiu, permitindo que as aplicações pudessem ser implementadas de forma distribuída, tornando-se um ponto importante no desenvolvimento de sistemas de gerência destinados a dar suporte a gerência integrada de rede das empresas operadoras de telecomunicações.

Neste contexto, de forma a nortear o desenvolvimento e a obtenção de sistemas, buscando obter benefícios tais como reusabilidade, modularidade e portabilidade das aplicações de gerência, surgiu o conceito de *Plataformas de Aplicações de Gerência*. Uma Plataforma de Aplicações de Gerência consiste de uma camada de software que reside entre o sistema operacional e as aplicações de gerência. Torna transparente para o usuário os detalhes tecnológicos do ambiente computacional, oferecendo um acesso padronizado aos serviços prestados por ela, e também um suporte a um ambiente distribuído.

O acesso aos serviços prestados pelas plataformas às aplicações de gerência deve ser feito através de Interface de Programa de Aplicação ou API (*Application Program Interface*). Uma API deve fornecer um conjunto consistente de primitivas e estruturas de dados ao desenvolvedor de aplicações, substituindo as interfaces proprietárias de cada plataforma para acesso aos serviços.

Duas entidades atuam fortemente na padronização de API: o POSIX (*Portable Operating System Interface for Computer Environment*) e a X/Open. A X/Open definiu as API XMP e XOM. A API XMP provê uma interface comum para os serviços oferecidos pelo SNMP e pelo CMIS. A API XOM é utilizada para a definição dos argumentos a serem utilizados pelo conjunto de funções que fazem parte da definição da API XMP. Nos últimos anos começaram a surgir no mercado produtos que implementam os conceitos de Plataforma de Aplicações de Gerência, disponibilizando aos desenvolvedores de aplicações o acesso aos serviços de comunicação através das API XOM/XMP.

## 1.1 O Contexto do Desenvolvimento do Trabalho

Paralelamente à padronização dessas API, assim como à padronização da própria arquitetura TMN, a TELEMIG, empresa operadora do Sistema TELEBRÁS, decidiu desenvolver um sistema de gerência de rede denominado SIS – *Sistema Integrado de Supervisão* [NM96], dando início a implementação da sua gerência integrada de rede. Este sistema integra sistemas de supervisão já existentes na planta de telecomunicações da empresa, implementando principalmente a gerência de falhas dos elementos pertencentes à rede de telecomunicações da empresa. Este projeto está sendo desenvolvido através de uma parceria entre a TELEMIG e o Departamento de Ciência da Computação da UFMG e está em sua terceira fase de desenvolvimento.

A primeira fase do projeto, de 1991 à 1993, teve como objetivos principais a construção de uma plataforma básica para o sistema e a integração, através de uma interface de comunicação proprietária [BIC93, CN93, NMC93], de sistemas de supervisão da planta analógica da empresa. Na segunda fase, de 1993 à 1995, foi incorporado ao sistema o paradigma gerente-agente adotado pelas normas ISO e pelo ITU-T, para a gerência de redes de computadores e gerência de redes de telecomunicações, respectivamente. Utilizando este paradigma, foram integradas ao SIS as centrais digitais AXE/Ericsson, Trópico RA/CPQD, Elcom/Batik, EWSD/Siemmens, e a central de comutação da telefonia celular da Northern Telecom. Essa interface gerente-agente [NBMD94] levou em consideração vários aspectos abordados pela padronização. Assim, embora ainda suportada por um protocolo proprietário da camada de aplicação, foi definido um modelo de informação próximo dos padrões adotados pelo ITU-T para a gerência de redes de telecomunicações. A plataforma básica do SIS já se encontra instalada e em operação nos centros de gerência de rede da empresa desde setembro de 1993. Atualmente supervisiona quase a totalidade da planta de transmissão analógica e um grande número das centrais digitais dos tipos mencionados acima.

A terceira fase de desenvolvimento, de 1995 à 1997, tem como um de seus objetivos principais dotar o SIS de uma interface Q3, conforme definido pela recomendação M.3010 [CCI92] que descreve os princípios da arquitetura TMN. De acordo com esta recomendação, a troca de informações de gerência entre as entidades gerente e agente deve ser feita através do uso do CMIS [CCIa] e do CMIP [CCIb]. Através desta interface, poderão ser integrados elementos de rede que possuam interfaces padronizadas, e até mesmo fazer a integração do SIS a outros sistemas de suporte a operação, os quais também darão suporte a gerência integrada de rede da empresa. A figura 1 mostra a evolução das interfaces de comunicação utilizadas para a integração de elementos da rede de telecomunicações ao sistema SIS, para as três fases de desenvolvimento. Para auxiliar a implementação da interface TMN (Q3), está sendo utilizado o produto *SunLink*, que oferece o acesso aos serviços da infra-estrutura de comunicação através das API XOM/XMP.

Outra iniciativa está sendo levada em paralelo. O Departamento de Sistemas de Operações do CPQD TELEBRÁS conta com um laboratório, chamado Laboratório TMN, especialmente voltado para o estudo e avaliação dos aspectos de implementação da TMN nas empresas operadoras de telecomunicações. Um dos objetivos deste laboratório é prover o Sistema TELEBRÁS com especificações técnicas que orientem no desenvolvimento e na obtenção de sistemas de gerência, visando benefícios tais como a interoperabilidade, reusabilidade e portabilidade das aplicações de gerência. As API XOM/XMP estão sendo analisadas como parte desse conjunto de especificações a serem utilizadas para os sistemas de gerência TMN.

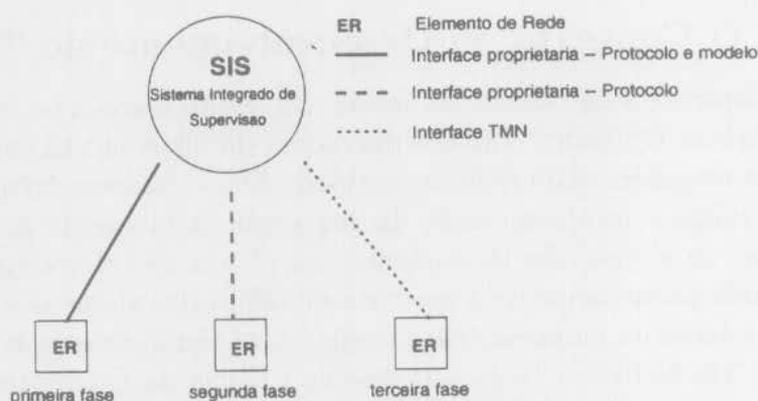


Figura 1: Evolução das interfaces para integração de elementos de rede ao SIS

## 1.2 Organização do Artigo

As duas próximas seções deste artigo descrevem as API XOM/XMP e apresenta uma experiência prática da utilização dessas API através do desenvolvimento de uma aplicação gerente-agente. A última seção apresenta uma conclusão, abordando principalmente os aspectos de implementação e portabilidade das aplicações que utilizam essas API para acesso aos serviços de comunicação de uma plataforma.

## 2 Interface de Programa de Aplicação – API

As especificações para a gerência de redes definidas por organismos de padronização, tais como ISO e ITU-T dão ênfase à definição de padrões para as interfaces interoperáveis entre dois sistemas de gerência. Estas interfaces são baseadas no serviço CMIS e no protocolo de gerência CMIP, em conjunto com estruturas padronizadas para representação e troca da informação de gerenciamento representada na forma de objetos gerenciados, definidos de acordo com a recomendação X.722 [CCId] do ITU-T, chamada de *Guidelines for the Definition of Managed Objects* (GDMO). A figura 2 mostra a comunicação entre uma entidade agente e outra gerente através de uma interface interoperável. Ilustra também o mapeamento de recursos reais em objetos gerenciados. Estes objetos são vistos pelo sistema de gerenciamento através da interface.

Nenhum padrão recomendado pela ISO ou pelo ITU-T define *como* uma aplicação de gerência deve ser implementada de forma a se ter uma interface interoperável. Para os desenvolvedores de aplicações isso é um fator de dificuldade. Especificamente são mostrados abaixo alguns pontos que são ou não são objetos de padronização.

- As normas utilizadas na camada de apresentação do modelo OSI definem como um valor, descrito na notação ASN.1 [Ros89], é transmitido através de uma interface interoperável, mas não existem normas que definem como este dado deve ser manipulado pela aplicação;
- O serviço CMIS define um conjunto de primitivas de gerenciamento, mas não define como uma aplicação tem acesso a estes serviços;
- As normas definem o conceito de objeto gerenciado como sendo uma abstração do recurso do mundo real que deve ser gerenciado. Definem também as propriedades dos

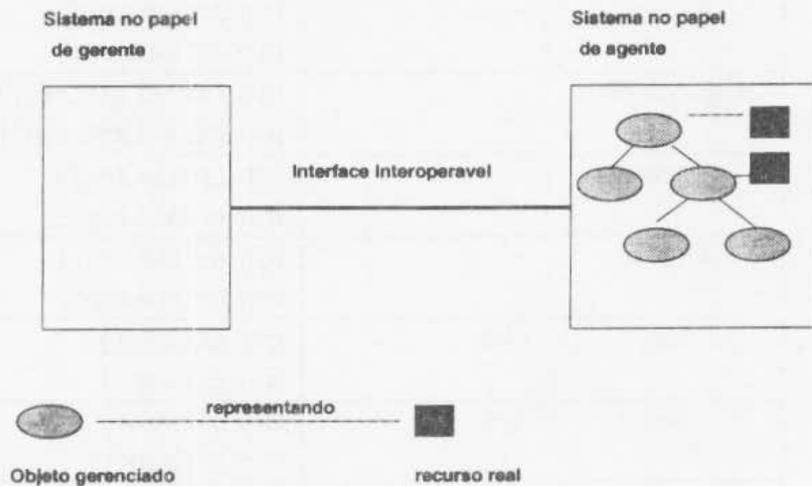


Figura 2: Interface interoperável

objetos gerenciados que são visíveis através de uma interface interoperável (atributos, operações, comportamentos e notificações). Entretanto não definem como estas propriedades são implementadas ou acessadas por uma aplicação.

O objetivo de uma API é prover um conjunto consistente de primitivas e estruturas de dados ao desenvolvedor de aplicações, substituindo as interfaces proprietárias de cada plataforma para acesso aos seus serviços. Assim, a definição de uma API facilita o desenvolvimento de aplicações e permite a portabilidade das mesmas em um ambiente multifornecedor, já que o acesso aos serviços se torna padronizado.

Foram com estes objetivos que a API XMP (*X/Open Management Protocol*) foi especificada, sendo hoje um padrão *de facto* para as plataformas de aplicações de gerência. A API XMP oferece aos desenvolvedores de sistemas de gerência um mecanismo comum de acesso, tanto para o serviço CMIS, utilizado principalmente em sistemas de gerência de redes de telecomunicações, como para o serviço SNMP, utilizado principalmente em sistemas de gerência de redes de computadores. Juntamente com a API XMP a X/Open definiu a API XOM (*X/Open Abstract Data Manipulation*) para a manipulação de tipos de dados definidos em ASN.1 [FOR]. Estas duas API serão descritas nas próximas seções.

## 2.1 A API XMP

A API XMP (*X/Open Management Protocols*) consiste de uma biblioteca de funções que mapeiam os serviços dos protocolos de gerência CMIP e SNMP.

As chamadas aos serviços providos pela API XMP podem ser síncronas ou assíncronas. A API XMP utiliza a API XOM para criar, examinar, modificar e destruir os argumentos utilizados pelas suas funções. A interface XMP é simétrica, provendo todas as funcionalidades requeridas para implementar todos os tipos de gerente e agente.

As funções da API XMP suportam os sete serviços CMIS e os quatro serviços oferecidos pelo SNMP, tanto no modo *requestor*, onde a aplicação solicita o serviço, quanto no modo *responder*, onde a aplicação recebe uma solicitação de serviço para ser executada. O mapeamento dos serviços oferecidos pelos protocolos de gerência nos serviços oferecidos pela API XMP são mostrados na tabela 1.

Serviço CMIS	Serviço SNMP	Funções XMP
Action	-	mp_action_req() mp_action_rsp()
Cancel-Get	-	mp_cancel_get_req() mp_cancel_get_rsp()
Create	-	mp_create_req() mp_create_rsp()
Delete	-	mp_delete_req() mp_delete_rsp()
Get	Get	mp_get_req() mp_get_rsp()
Set	Set	mp_set_req() mp_set_rsp()
-	Get-Next	mp_get_next_req() mp_get_next_rsp()
Event Report	Trap	mp_event_report_req() mp_event_report_rsp()

Tabela 1: Mapeamento dos serviços de gerência nos serviços da API XMP

Além de prover uma interface para os serviços oferecidos pelos protocolos de gerência, a API XMP define uma série de serviços adicionais que são necessários para a gerência de alguns serviços de suporte. Alguns destes serviços são mostrados na tabela 2.

Funções XMP	Descrição do Serviço
mp_bind()	Abre uma sessão com a infra-estrutura de comunicação
mp_initialize()	Inicializa um <i>workspace</i> XOM
mp_shutdown()	Destroi o <i>workspace</i> criado
mp_negotiate()	Associa pacotes OM com o <i>workspace</i> da aplicação

Tabela 2: Serviços adicionais da API XMP

## 2.2 A API XOM

Para a utilização dos serviços oferecidos pela API XMP, o desenvolvedor de aplicações de gerência deve manipular tipos de dados definidos em ASN.1. De forma a evitar que o programador tenha que definir e manipular estruturas de dados grandes e complexas utilizando ASN.1, foi definida pela X/Open a API chamada de XOM (*X/Open Abstract Data Manipulation*).

O principal objetivo da API XOM é esconder a complexidade da notação ASN.1, provendo um mecanismo geral de representação e manipulação de dados abstratos, sendo especialmente idealizada para uso com outras API OSI. A API XOM define uma interface orientada a objeto, embora não incorporando todas as características desta abordagem, tais como a propriedade de encapsulamento.

Desta forma, as variáveis e parâmetros utilizadas nas chamadas às primitivas que implementam os serviços da API XMP são definidos de acordo com a API XOM, sendo que as funções da API XOM são utilizadas para criar, examinar, modificar e destruir os argumentos das funções definidas pela API XMP.

A API XOM provê uma representação dos dados, definindo uma estrutura chamada de objetos OM. Um objeto OM é uma estrutura definida em linguagem C, usada para representar tipos de dados abstratos que refletem uma representação dos dados em ASN.1, que são utilizados pela infra-estrutura de comunicação.

### 2.2.1 Arquitetura da Informação

Nesta seção são descritas as estruturas básicas de dados que a aplicação cliente troca com o provedor de serviço, que são mantidas por este e que se tornam acessíveis para a aplicação cliente. O serviço é entendido aqui como o software que implementa a API. Os nomes das estruturas definidas na API XOM sempre têm a sigla OM associada. Isto é útil para evitar confusão entre termos tais como objeto gerenciado (abstração de um recurso) e objeto OM (estrutura de dados).

- **OM Attribute:** Um atributo OM é o componente básico de um objeto OM. Ele é representado por uma estrutura em C chamada de *OM\_descriptor*, que é composta de três campos: o tipo do atributo, sua sintaxe e o seu valor.
- **OM Object:** Um objeto OM é constituído de uma lista de atributos OM e é representado, na linguagem C, por um vetor de estruturas de dados do tipo *OM\_descriptor*. O número de atributos, o tipo e a sintaxe de cada um dos atributos que compõem um objeto é especificado pela classe a qual este objeto pertence.
- **OM Class:** Uma classe OM define os seguintes elementos: o nome da classe, a identificação de sua superclasse, a definição dos atributos específicos da classe e uma indicação se a classe é abstrata, isto é, não instanciável, ou concreta, isto é, instanciável.
- **Packages:** Um pacote XOM é um conjunto de classes OM que possuem funcionalidades correlatas. Quando se utiliza a API XOM em conjunto com a API XMP são disponíveis quatro pacotes básicos: *OM package*, *Common package*, *CMIS package*, *SNMP package*. Os pacotes *OM* e *Common* são automaticamente incluídos nos pacotes *CMIS* e *SNMP*. Pode-se definir outros pacotes XOM para se criar classes de objetos OM específicas para uma determinada aplicação.
- **Workspace:** Um workspace é definido como sendo um repositório de instâncias de objetos. Os objetos instanciados em um workspace são pertencentes a classes de objetos. Estas classes, por sua vez, fazem parte de pacotes associados ao próprio workspace.

Dentre as características de orientação a objeto, a API inclui os conceitos de herança, superclasse, classe e subclasses. Para exemplificar esses conceitos, seja a função XMP *mp-get\_req*, que é utilizada tanto para operações no serviço CMIS como no serviço SNMP. Para possibilitar seu uso para os dois serviços, ela tem como parâmetro a classe abstrata OM *Get-Argument*. Esta classe é superclasse das classes concretas *CMIS-Get-Argument* e

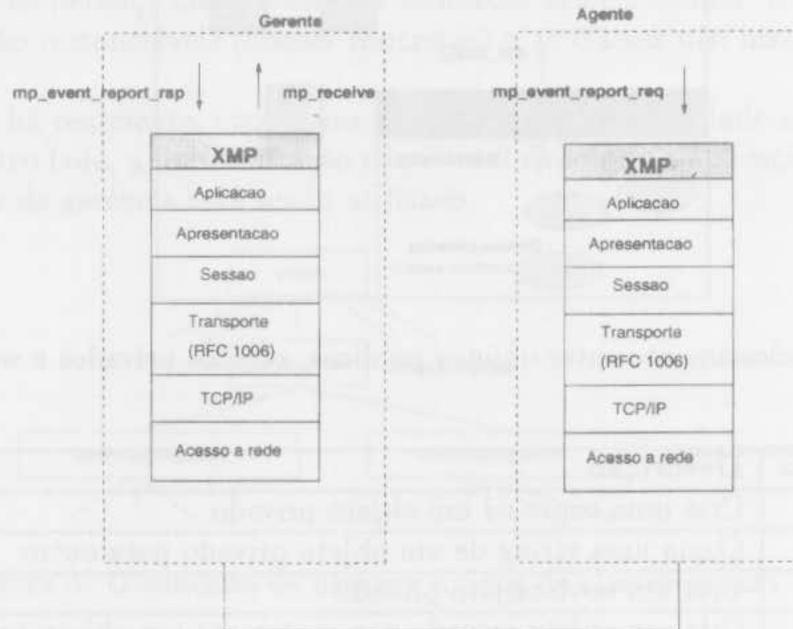


Figura 5: Arquitetura de comunicação de uma interface gerente-agente utilizando API XMP

agente através do uso da função *mp\_negotiate*, definida pela API XMP. Para sanar esta dificuldade, foi utilizado o pacote de software de domínio público ISODE, versão 8.0, conforme descrito mais adiante. Na versão 7 da API XMP, novos pacotes podem ser definidos, contendo novas classes de objetos OM.

Na implementação da aplicação gerente ou agente, um conjunto de regras foram aplicadas garantindo o correto funcionamento da interface. São elas:

- Inicialização do espaço de trabalho ou *workspace* através da utilização da primitiva *mp\_initialize*;
- Abertura de uma ou mais sessões utilizando a primitiva *mp\_bind*.
- Realização das interações de gerenciamento, operações e notificações, utilizando as funções de interface oferecidas.
- Fechamento das sessões abertas através da primitiva *mp\_unbind*.
- Descarte do workspace através da primitiva *mp\_shutdown*.

O diagrama de estados mostrado na figura 6 apresenta uma transação particular entre as entidades gerente e agente. Os estados deste diagrama são descritos na tabela 4. Transições que saem do estado IDLE indicam o início de uma transação, enquanto as transições que chegam ao estado IDLE indicam que uma transação foi completada.

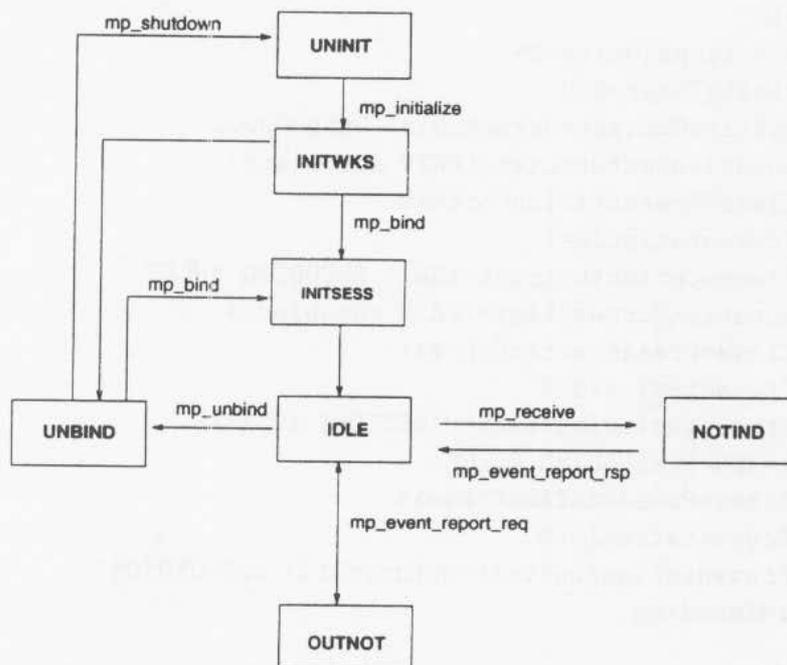


Figura 6: Diagrama de estados da aplicação gerente ou agente que usa as funções da API XMP

Estado	Descrição
UNINIT	workspace não inicializado
INITWKS	workspace inicializado
INITSESS	Sessão inicializada
UNBIND	Sessão fechada
IDLE	Nenhuma interação de gerenciamento foi inicializada
NOTIND	Uma notificação foi recebida
OUTNOT	Uma notificação foi enviada

Tabela 4: Descrição dos estados de uma aplicação agente ou gerente

A abertura de uma sessão, realizada através da chamada à primitiva *mp\_bind*, é uma ligação entre a aplicação e a API XMP. Se a chamada à função *mp\_bind* for bem sucedida, ela retorna um objeto de sessão contendo todos os parâmetros necessários para o estabelecimento de uma associação, conforme apresentado abaixo.

```

AGENT: SESSION RETURNED
.   Class=Session
.   FileDescriptor=0
.   RequestorAddress=(PRIV subobject)
.   .   Class=PresentationAddress
.   .   NAddresses=0x
.   .   PSelector=0x72666335
.   .   SSelector=0x707273
.   .   TSelector=0x
  
```

```

.   Role=15
.   CMISFunctionalUnits=23
.   InactivityTimer=600
.   ApplicationContext=UnknownOid: 0x59000002
.   PresentationContextList=(PRIV subobject)
.     .   Class=PresentationContext
.     .   PresentationId=1
.     .   PresentationAbstract=ASN.1 ENCODING RULES
.   PresentationContextList=(PRIV subobject)
.     .   Class=PresentationContext
.     .   PresentationId=3
.     .   PresentationAbstract=DIRECTORY ACSE AS
.   Rules=ASN.1 ENCODING RULES
.     .   Class=PresentationContext
.     .   PresentationId=5
.     .   PresentationAbstract=UnknownOid: 0x590101041
.     .   Class=Encoding

```

No momento da chamada à primitiva *mp\_bind*, a interface que implementa a API XMP não acessa a pilha de protocolos. O acesso a esta pilha só é realizado no momento em que uma operação de gerenciamento ou uma notificação é recebida pela interface que implementa a API. A supervisão da associação é feita de modo transparente pela API XMP.

São criados também, pelas aplicações gerente e agente, objetos de contexto. Estes objetos definem características particulares de uma operação de gerenciamento ou de uma notificação. Vários detalhes administrativos estão contidos nos objetos de contexto, como por exemplo se a operação é síncrona ou assíncrona, se o serviço é com confirmação ou sem confirmação, a prioridade da requisição e etc. É mostrado abaixo um objeto de contexto criado pela aplicação agente.

```

AGENT: CONTEXT TO BE USED
.   Class=Context
.   Mode=1
.   TimeLimit=600
.   Priority=3
.   ResponderAddress=(PRIV subobject)
.     .   Class=PresentationAddress
.     .   NAddresses=0x96a40709
.     .   PSelector=0x72666330
.     .   SSelector=0x707273
.     .   TSelector=0x
.   Asynchronous=False

```

Após a inicialização do *workspace* e após a criação dos objetos de contexto e de sessão, a aplicação agente cria uma instância de um objeto OM da classe *CMIS-Event-Report-Argument*, que irá conter as informações relacionadas ao evento que se quer notificar. Estas informações são representadas na forma de atributos da instância do objeto. Um objeto da classe *CMIS-Event-Report-Argument* herda os atributos da sua superclasse denominada *Event-Report-Argument*. A classe *Event-Report-Argument* é uma classe abstrata, portanto não instanciável, que herda os atributos de sua superclasse denominada *Object*.

Além dos atributos herdados, uma instância de um objeto desta classe possui os atributos adicionais mostrados na tabela 5. A primeira coluna dessa tabela indica o nome de cada atributo. A segunda coluna indica a sintaxe do atributo. O nome de certas sintaxes são construídos a partir de *templates de sintaxe*. Um template de sintaxe é uma construção léxica na forma *identificador-primário(identificador-secundário)*. Um template de sintaxe compreende um grupo de sintaxes relacionadas. Qualquer membro de um grupo é identificado pelo identificador primário. Um membro particular é representado pelo identificador secundário associado ao template. A terceira coluna indica o número de valores que o atributo pode assumir.

Atributo	Sintaxe	Número de valores
managed-Object-Class	Object(Object-Class)	1
managed-Object-Instance	Object(Object-Instance)	1
eventTime	String (Generalized-Time)	0-1
event-Type	Object (Event-Type-Id)	1
event-Info	any	0-1

Tabela 5: Atributos adicionais do objeto da classe *Event-Report-Argument*

Conforme mostrado na tabela 5, o atributo *event-Info* é do tipo *any*. Isto significa que a sintaxe e a semântica deste parâmetro depende do tipo de evento que está sendo reportado ao gerente. No caso desta aplicação, este atributo contém a informação de um alarme ocorrido em um objeto gerenciado genérico. A sintaxe da informação de alarme utilizada nesta aplicação está descrita na recomendação X.721 [CCIC] do ITU-T sendo do tipo construído ASN.1 **SEQUENCE**, conforme mostrado a seguir.

```
AlarmInfo ::= SEQUENCE {
    probableCause          ProbableCause,
    specificProblems      [1] SpecificProblems OPTIONAL,
    perceivedSeverity     PerceivedSeverity,
    backedUpStatus        BackedUpStatus OPTIONAL,
    backedUpObject        [2] ObjectInstance OPTIONAL,
    trendIndication       [3] TrendIndication OPTIONAL,
    thresholdInfo         [4] ThresholdInfo OPTIONAL,
    notificationIdentifier [5] NotificationIdentifier OPTIONAL,
    correlatedNotifications [6] CorrelatedNotifications OPTIONAL,
    stateChangeDefinition [7] AttributeValueChangeDefinition OPTIONAL,
    monitoredAttributes    [8] MonitoredAttributes OPTIONAL,
    proposedRepairActions [9] ProposedRepairActions OPTIONAL,
    additionalText         AdditionalText OPTIONAL,
    additionalInformation  [10] AdditionalInformation OPTIONAL
}
```

Como a versão 3 da API XMP não permite a negociação de novos pacotes entre as aplicações gerente e agente, contendo a definição de novas classes de objetos, a solução adotada foi tornar transparente para a API a estrutura da informação de evento que trafega na interface gerente-agente. Assim, o valor do atributo *event-info* foi transformado

pela aplicação em uma sequência de bits que representa a informação de alarme codificada utilizando BER (*Basic Encoding Rules*). A geração desta sequência de bits foi feita utilizando-se o ISODE.

Assim, a estrutura da informação de alarme a ser passada na interface gerente-agente foi definida em um módulo ASN.1. Utilizando o compilador *pepsy* do ISODE [Con93] foram geradas as funções de codificação e decodificação, assim como as estruturas de dados em linguagem C que são utilizadas pela aplicação. As funções de codificação geradas pelo compilador são utilizadas para a geração da sequência de bits que representa a informação de alarme codificada utilizando BER. Da mesma forma, as funções de decodificação decodificam esta sequência de bits do formato BER para um formato específico da aplicação.

É mostrado abaixo o conteúdo de um objeto da classe *CMIS\_Event\_Report\_Argument* instanciado pela aplicação agente.

```
AGENT: THE EVENT REPORT TO SEND TO THE MANAGER
.   Class=CmisEventReportArgument
.   ManagedObjectClass=(PRIV subobject)
.   .   Class=ObjectClass
.   .   GlobalForm=UnknownOid: 0x2b0601020506
.   ManagedObjectInstance=(PRIV subobject)
.   .   Class=ObjectInstance
.   .   DistinguishedName=(PRIV subobject)
.   .   .   Class=DSDN
.   .   .   RDNs=(PRIV subobject)
.   .   .   .   Class=DSRDN
.   .   .   .   AVAs=(PRIV subobject)
.   .   .   .   .   Class=Ava
.   .   .   .   .   NamingAttributeId=UnknownOid: 0x2b0601020506
.   .   .   .   .   NamingAttributeValue=Agente de teste
.   EventTime=19951221122618
.   EventType=(PRIV subobject)
.   .   Class=EventTypeId
.   .   LocalForm=2
.   EventInfo=0x306d0201010a01011932416c61726d6520646f2070726f636573736f
.   .   .   .   204147454e54452064652074657374652064652041504920584f4d2f584d50
.   .   .   .   003031010100042c4f757472617320696e666f726d61636f657320706f6465
```

Uma vez criados estes objetos, que estão associados a um determinado *workspace*, a aplicação agente chama a função da API XMP para enviar uma notificação, sendo que os argumentos *session*, *context* e *evReport* são os objetos criados anteriormente. É mostrado a seguir, a chamada a função *mp\_event\_report\_req* definida pela API XMP.

```
status = mp_event_report_req(session, context, evReport, &result, &id);
```

## 4 Conclusões

Alguns pontos puderam ser observados durante o desenvolvimento da aplicação descrita na seção anterior. O primeiro deles é que o número máximo de argumentos passados para as funções que mapeiam um serviço CMIS ou SNMP é cinco, sendo que os argumentos

representados pelos objetos de sessão e contexto sempre estão presentes. Assim, a chamada as funções da API XMP tornam-se aparentemente mais simples do que as chamadas de funções que implementam diretamente o serviço CMIS.

Outro ponto a ser ressaltado é que apesar da API XOM oferecer uma interface orientada a objeto, a aplicação não precisa ser necessariamente construída utilizando a metodologia de orientação a objeto.

Observa-se, também, que o desenvolvimento de uma aplicação utilizando estas API seria facilitado se a plataforma disponibilizasse uma ferramenta para a geração de pacotes. Esse tipo de ferramenta pode ser encontrado em várias plataformas comerciais e a sua finalidade é a geração automática, a partir da especificação GDMO e ASN.1, das definições das classes de objetos OM. Essas classes são agrupadas em um pacote e este pode ser associado a um workspace pertencente a uma aplicação, utilizando a primitiva da API XMP *mp\_negotiate*. Para isso, é necessário que a plataforma implemente a versão 7 da API XMP.

A partir do desenvolvimento dessa aplicação utilizando o produto *SunLink*, tentou-se executar esta mesma aplicação em um outro ambiente computacional, utilizando-se uma plataforma comercial para verificar os aspectos de portabilidade das aplicações. Esta plataforma também oferece o acesso a sua infra-estrutura de comunicação através das API XOM/XMP, sendo que esta plataforma implementa tanto a versão 3 como a versão 7 da API XMP[Net94, Hew94].

As primeiras experiências de execução da aplicação nesta plataforma indicaram que alguns fatores devem ser observados quando se objetiva tornar uma aplicação, que utiliza a API XMP, portátil para várias plataformas.

Um fator é a versão da API XMP utilizada. Aplicações desenvolvidas para a versão 3 não são totalmente portáteis para a versão 7 e vice-versa. Como exemplo, a versão 3 suporta apenas os objetos definidos nos pacotes básicos especificados na interface XOM/XMP (*OM*, *Common*, *CMIS* e *SNMP*). Isto significa que esta versão da API processa somente as estruturas de dados definidas nos objetos destes pacotes. Qualquer outra informação a ser transferida entre a aplicação gerente e agente deve ser necessariamente codificada (de ASN.1 para BER) e decodificada (de BER para ASN.1) pelo desenvolvedor de aplicações. Neste caso, uma alternativa é o uso de ferramentas de codificação e decodificação ASN.1 disponíveis em pacotes de domínio público, conforme mostrado na seção anterior. Quando se utiliza a versão 7 desta API esta pode ser estendida para suportar pacotes especificados pelo usuário. Deste modo, os objetos OM criados pelo usuário serão automaticamente codificados e decodificados pela API, liberando a aplicação desta tarefa.

Um outro fator que pode inviabilizar a portabilidade das aplicações é o uso de extensões da API XMP. Algumas plataformas disponibilizam funções, como por exemplo, funções de controle da associação, que não fazem parte da especificação da API.

Além disso, ao se desenvolver uma aplicação com o objetivo de torná-la portátil, deve-se analisar cuidadosamente os serviços oferecidos de forma transparente pela infra-estrutura da plataforma, que tornam o desenvolvimento da aplicação mais simples, mas que podem não existir em outras plataformas, como é o caso de serviços de diretórios para o cadastramento das aplicações. Com o uso desse serviço, o problema de endereçamento das aplicações passa a ser resolvido pela própria infra-estrutura da plataforma e não mais pela aplicação.

## Referências

- [BIC93] Patrícia Valéria Corrêa BICALHO. Metodologia de Integração de Sistemas de Telessupervisão ao Sistema Integrado de Supervisão (SIS). Master's thesis. UFMG-ICEEx-DCC, Belo Horizonte-MG, 1993.
- [CCIAa] CCITT. Recommendation X.710 (1991) — ISO/IEC 9595 : 1991, Common management information service definition for CCITT applications.
- [CCIBb] CCITT. Recommendation X.711 (1991) — ISO/IEC 9596-1 : 1991, Common management information protocol specification for CCITT applications.
- [CCICc] CCITT. Recommendation X.721 (1992) — ISO/IEC 10165-2 : 1992, Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.
- [CCIDd] CCITT. Recommendation X.722 (1992) — ISO/IEC 10165-4 : 1992, Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects.
- [CCII92] CCITT. Recommendation M.3010 – Principles For a Telecommunications Management Network, Consultative Comitee for International Telegraph and Telephone , 1992.
- [CN93] Patrícia Valéria Corrêa and José Marcos Silva Nogueira. Uma Interface para Integração de Redes de Telessupervisão a um Sistema Integrador de Supervisão. *11º Simpósio Brasileiro de Redes de Computadores*, Maio 1993. Campinas - SP.
- [Con93] ISODE Consortium. *Programmer's Guide ASN.1 Tools*, February 1993. Volume 18.
- [FIFL93] Benedito Luís FAYAN, Carlos Tadashi IMAI, Milton Ben-Hur FABER, and Paulo Sérgio LORENA. Plataforma de Suporte a Aplicações de Gerência. *Revista TELEBRÁS*, pages 110–121, 1993.
- [FOR] NETWORK MANAGEMENT FORUM. RGB Management API. Issue 1.0 Draft 1.
- [Hew94] Hewlett Packard. *HP OpenView Distributed Management Developer's Guide*, 1994.
- [NBMD94] José Marcos S. NOGUEIRA, Patrícia V.C. BICALHO, Murilo S. MONTEIRO, and João E.R. DANTAS. Interfaceamento com Elementos de Rede: Especificação da Interface Genérica com o SIS. RT SIS 3104, UFMG-DCC-ICEEx, Belo Horizonte-MG, agosto 1994. 1ª edição.
- [Net94] Network & System Management Division. *Developing Applications on the DM Platform*, 1994. HP OpenView.

- [NM96] José Marcos Silva NOGUEIRA and Dilmar Malheiros MEIRA. The SIS Project: A Distributed Platform for the Integration of Telecommunication Management Systems. In *1996 IEEE Network Operations and Management Symposium*. IEEE/IFIP, IEEE Communications Society, April 1996.
- [NMC93] José Marcos Silva Nogueira, Murilo Silva Monteiro, and Patrícia Valéria Corrêa. Integration, automatic monitoring and control in conventional tele-supervisory systems. Moscow (Russia), July 1993. AMSE, Association for the Advancement of Modelling and Simulation techniques in Enterprises. International 93 Moscow Conference.
- [RAM93] Eduardo Antônio RAMALHO. Gerência Integrada de Redes e Serviços. *Revista TELEBRÁS*, pages 07-11, 1993.
- [Ros89] Marshall T. Rose. *The Open Book - A Practical Perspective on OSI*, chapter 8, page 225 to 335. Prentice Hall, 1989.
- [Sun93a] SunConnect – Sun Microsystems, Inc. *SunLink CMIP 8.0 Management Protocol Programmer's Guide*, 1993. Version 8.0.
- [Sun93b] SunConnect – Sun Microsystems, Inc. *X/Open Management Protocols (XMP) API*, 1993. Version 8.0.
- [Sun93c] SunConnect – Sun Microsystems, Inc. *X/Open OSI-Abstract Data Manipulation (XOM) API*, 1993. Version 8.0.