

RELATÓRIO TÉCNICO

“GERÊNCIA BASEADA NA WEB”

Suzana Ramos Teixeira¹
suzana@ravel.ufrj.br

Luís Felipe M. de Moraes
moraes@ravel.ufrj.br

José Helvécio Teixeira Jr.²
helveciot@ravel.ufrj.br

Laboratório de Redes de Alta Velocidade - RAVEL
Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
COPPE/UFRJ
Caixa Postal 68511 - CEP 21945-970
Rio de Janeiro/RJ

- (1) Aluna de Doutorado. Bolsista CAPES.
(2) Aluno de Doutorado. Bolsista FAPERJ.

RESUMO

Atualmente, o gerenciamento praticado pelas organizações envolve o uso separado de ferramentas e técnicas incompatíveis para o gerenciamento de redes, sistemas, aplicações e serviços. O que se precisa urgentemente é de uma ferramenta que integre essas áreas de gerenciamento. **Gerência baseada na Web** ou **Gerenciamento através da Web** é um enfoque promissor que pode fornecer uma solução de gerenciamento verdadeiramente integrada, uniforme e simples. Neste trabalho, enfocamos as tecnologias emergentes envolvidas no Gerenciamento através da Web.

1. Introdução

Administradores hoje em dia utilizam tipicamente diversas ferramentas de gerenciamento para monitorar e controlar redes, sistemas, aplicações e serviços, em razão dos mesmos terem infra-estruturas incompatíveis (diferentes modelos de informação, métodos de acesso de informação e protocolos de gerenciamento, por exemplo) [2].

Além das ferramentas de gerenciamento de redes, sistemas, aplicações e serviços não serem integradas, elas comumente são baseadas em um enfoque centralizado (apresentando problemas inerentes a um esquema de gerenciamento centralizado) e são limitadas no escopo de gerência (ou seja, apresentam o problema da escalabilidade, sendo adequadas para gerenciar somente ambientes de pequeno porte). Todos esses motivos contribuem para explicar o porquê da tarefa de gerenciamento ser tão complexa.

Gerência baseada na Web ou **Gerenciamento através da Web** é um enfoque promissor que pode fornecer uma solução verdadeiramente integrada, uniforme e simples para o gerenciamento de redes, sistemas, aplicações e serviços.

Soluções de gerenciamento baseadas na Web são atrativas por várias razões, dentre elas:

- o gerenciamento baseado na Web fornece uma interface de usuário uniforme, como o mesmo ver e sentir (*look and feel*), denominada *browser* (navegador) Web, que é uma interface muito bem estabelecida tanto para recuperação de dados como para execução de aplicações;
- administradores não precisam usar ferramentas diferentes para gerenciar redes, sistemas aplicações e serviços;
- navegadores Web estão disponíveis em várias plataformas;
- o pré-requisito para um gerenciamento integrado eficiente é que aplicações de gerenciamento possam ser acessadas de qualquer lugar e que agentes de gerenciamento estejam disponíveis em todos os recursos a serem gerenciados. Se navegadores Web forem usados como console de gerenciamento, aplicações de gerenciamento tornar-se-iam disponíveis de qualquer lugar; e
- soluções baseadas na Web constituem uma maneira eficiente e barata para reduzir os investimentos pesados associados com consoles dedicadas, característica de soluções proprietárias.

A usabilidade intuitiva e simples de interfaces de usuários de navegadores Web simplifica a tarefa de gerenciamento. Além do fácil uso e de estarem disponíveis em diversas plataformas, navegadores Web são atrativos para propósito de gerenciamento principalmente por oferecer um ambiente de execução para programas independentes de plataforma.

Neste trabalho, enfocamos as tecnologias emergentes envolvidas no gerenciamento baseado na Web. Na seção 2, introduzimos sucintamente a tecnologia Web e a linguagem Java, que constituem tecnologias chaves da Internet muito utilizadas no desenvolvimento de soluções de gerenciamento através da Web. Na seção 3, revemos as técnicas de gerenciamento padronizadas existentes, denominadas Gerenciamento Internet e Gerenciamento OSI (*Open Systems Interconnection*). Na seção 4, examinamos opções de gerenciamento através da Web. Na seção 5, apresentamos as arquiteturas padronizadas de gerenciamento baseada na Web. Por fim, na seção 6, enumeramos algumas de nossas considerações.

2. Tecnologias Chaves da Internet

Nesta seção, introduzimos a tecnologia Web e a linguagem Java, que constituem tecnologias muito utilizadas no desenvolvimento de soluções de gerenciamento através da Web. As tecnologias Web e Java estão revolucionando o uso da Internet nos últimos anos.

2.1. A Tecnologia Web

A *World Wide Web*, WWW ou simplesmente Web [4] consiste em uma maneira simples de organizar e acessar dados em diversas mídias usando hiperligações (*hyperlinks*) na Internet. Um *hyperlink* consiste em uma associação entre uma informação com o endereço de um outro documento que contém informações relacionadas. Relacionando documentos, a Web facilita a localização de informações para usuários.

Na Web, existem vários documentos e cada um possui um nome único, chamado URL (*Uniform Resource Locator* – Localizador de Recursos Uniformes) ou, simplesmente, endereço Web.

O protocolo nativo da Web é o HTTP (*HyperText Transfer Protocol* – Protocolo de Transferência de Hipertexto), mas a Web também suporta protocolos Internet tradicionais como o TELNET (*Telecommunications Network*), o FTP (*File Transfer Protocol*) e o SMTP (*Simple Mail Transfer Protocol*). Dessa forma, a Web torna-se *front-end* para aplicações disponíveis na Internet, integrando os principais protocolos e aplicações da Internet em uma interface fácil de ser usada.

O HTTP é um protocolo do nível de aplicação que implementa conceitos de orientação a objetos genéricos. Um aspecto importante do HTTP é que ele permite que sistemas distribuídos sejam construídos independentemente dos dados a serem transferidos.

A linguagem HTML (*HyperText Markup Language*) é a linguagem utilizada na Web para criar documentos hipertextos portáteis de uma plataforma para outra. Documentos HTML são documentos SGML (*Standardized Generalized Markup Language*) com semântica genérica apropriada para representar informações de uma vasta abrangência de domínios.

A Web é baseada na arquitetura cliente/servidor e basicamente opera da seguinte forma: documentos hipertextos a serem disponíveis na Web são preparados em HTML e tornados acessíveis através de um servidor Web; usuários desejando acessar documentos podem recuperá-los usando um cliente Web (navegador Web, por exemplo: *Netscape*,

Internet Explorer, Mosaic), que faz a conexão com o servidor Web que contém tais documentos.

O CGI (*Common Gateway Interface*) é um padrão para interfaceamento de aplicações externas com servidores Web. Um documento HTML que um *daemon* Web recupera é estático, o que significa que ele existe num estado constante (um arquivo texto que não muda). Um programa CGI, por outro lado, é executado em tempo real, podendo assim produzir informações de saída dinâmicas. Por exemplo: suponhamos que você deseje acoplar um banco de dados UNIX à Web para permitir que pessoas possam pesquisá-lo. Basicamente, você precisará criar um programa CGI que o *daemon* Web irá executar para transmitir informações para o motor do banco de dados, recuperar resultados e mostrá-los para o cliente. O programa CGI funciona pois como um *gateway*. A Figura 1 ilustra o princípio do CGI.

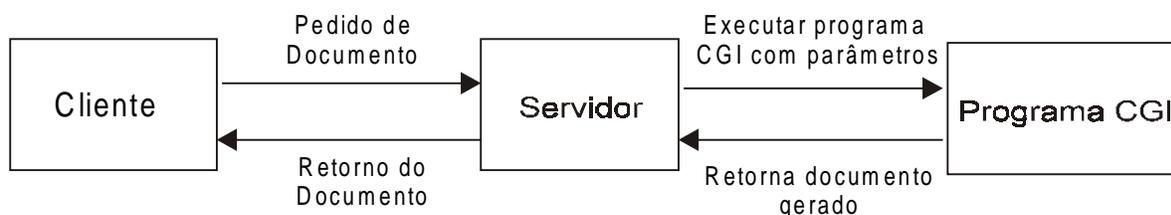


Figura 1: Princípio do CGI

Basicamente, os desenvolvedores da Web usam *scripts* CGI para permitir que os usuários se comuniquem com um aplicativo, como um programa de banco de dados, que normalmente reside no servidor Web.

Não existem limites ao que se possa acoplar à Web. Muitas das soluções de gerenciamento baseadas na Web usam CGI, que agem como *gateway*, para gerenciar agentes.

Vários esforços têm sido propostos utilizando a tecnologia Web no gerenciamento de redes, sistemas, aplicações e serviços. Usando a tecnologia Web como interface de integração, uma apresentação unificada de todas as informações de gerenciamento é fornecida e um estilo consistente de invocar operações de gerenciamento é seguido.

2.2. A Linguagem Java

Desenvolvida pela *Sun Microsystems*, Java [6, 10, 11] é uma linguagem orientada a objeto que tornou-se uma linguagem de grande interesse para a comunidade Internet quando a *Sun* lançou o *HotJava*, um *browser* (navegador) Web que podia executar pequenos programas Java embutidos, chamados *applets*, dentro de páginas Web.

Hoje, as *applets* Java estão presentes nos *sites* Web mais sofisticados, podendo ser executadas por qualquer *browser* Web com recursos Java.

Os programas Java rodam dentro de máquinas virtuais, as quais ficam dentro do computador no qual eles estão rodando. Um programa Java não tem contato com o computador real; ele conhece apenas a máquina virtual. Quando se escreve um programa Java e o compila, ele está pronto para ser executado em qualquer computador que contenha uma máquina virtual Java. A máquina virtual resolve o que os programas Java podem ou não fazer e age como uma *firewall* (porta corta-fogo) entre o computador anfitrião (*host*) e o programa Java. Um programa Java nunca acessa os dispositivos de entrada e/ou saída, o sistema de arquivos ou mesmo a memória de um computador. Em vez disso, ele pede à máquina virtual que os acesse.

Antes do Java, a maioria dos programas relacionados com a Web tinha de ser executada na máquina do servidor Web. A execução de *applets* no computador cliente, com *browser* Web e máquina virtual Java, é uma grande inovação na programação para a Web.

Java não é notável apenas porque *applets* do Java podem ser executadas dentro de páginas Web. Entre outros motivos também porque:

- fornece aos programadores, através de APIs (*Application Programming Interfaces*), a possibilidade de desenvolver programas independentes de plataforma. Programas Java são compilados para *bytecodes* independentes de plataforma. Esses *bytecodes* podem ser executados em qualquer máquina com sistema de execução (*runtime system*) Java, ou seja, um interpretador Java;
- foi projetada para suportar aplicações em rede. Para possibilitar que aplicações Java executem em qualquer lugar da rede, o compilador Java gera um arquivo de formato neutro (independente de plataforma). Assim, o código compilado pode ser executado em processadores distintos, dada a presença do sistema de execução (*runtime system*) Java. Isto é importante não só para redes, mas também para distribuição de software. Com Java, uma mesma versão de uma aplicação executa em várias plataformas;

- software desenvolvido em Java é portátil através de várias plataformas e pode ser distribuído e acessado por navegadores Web;
- oferece os benefícios da orientação a objetos, o que permite o desenvolvimento de códigos mais próximos dos problemas do mundo real. Para tornar a linguagem Java mais simples, os projetistas decidiram não implementar sobrecarga de operadores e herança múltipla;
- suporta multiprocessamento com suas classes *thread* (linhas de execução);
- executa automaticamente a coleta de lixo (*garbage collection*), liberando memória que não está mais sendo usada;
- através do conceito de máquina virtual, implementa um mecanismo de segurança. A máquina virtual incapacita os programas Java de causar qualquer dano ao computador para o qual eles são descarregados;
- é uma linguagem dinâmica. A máquina virtual liga os programas Java em tempo de execução (*runtime*), eliminando a necessidade de ligar bibliotecas em tempo de compilação;
- dispõe de um conjunto de funcionalidades que permite interação com a Internet (em Java, tem-se uma extensiva biblioteca de rotinas para interação com protocolos da Internet como TCP/IP, HTTP e FTP). É possível com APIs usar abstrações de alto nível, como *os URLs (Uniform Resource Locators*– Localizadores de Recursos Uniformes), ou implementar comunicações em níveis muito baixos simplesmente usando pacotes. Aplicações em Java podem acessar objetos em rede via URLs.

Java é, pois, uma linguagem de programação orientada a objetos que é dita ser um subconjunto de C++, sem os aspectos confusos de C++ (operadores de sobrecarga, herança múltipla, por exemplo).

Java foi projetada para ser usada em ambientes distribuídos. Em razão disso, foi dada muita ênfase a aspectos de segurança no seu projeto. Java possibilita a construção de sistemas livres de vírus. Técnicas de autenticação são baseadas na criptografia de chave pública.

O uso da linguagem Java no desenvolvimento de aplicações de gerenciamento baseado na Web deve-se principalmente ao fato que software desenvolvido em Java é portátil através de várias plataformas e pode ser distribuído e acessado por navegadores Web. A instalação de novas versões de software de gerenciamento em uma rede é um processo complexo e caro. *Downloading* transparente e rápido via um navegador Web Java constitui uma alternativa atrativa.

3. Técnicas de Gerenciamento Existentes

Nesta seção, nós revemos sucintamente as técnicas de gerenciamento padronizadas existentes, denominadas Gerenciamento Internet e Gerenciamento OSI. Mais informações sobre as referidas técnicas podem ser encontradas em [13, 14, 15] .

O Gerenciamento Internet e o Gerenciamento OSI correspondem, respectivamente, ao padrão de fato (*de facto*) e ao padrão oficial (*de jure*) para o gerenciamento de redes de computadores.

3.1. Gerenciamento Internet

O Gerenciamento Internet foi desenvolvido em decorrência da necessidade de se gerenciar a rede Internet, que se utiliza da *suite* (conjunto) de protocolos referenciado como TCP/IP (*Transmission Control Protocol/Internet Protocol*) [12] para prover comunicação entre redes de tecnologias distintas.

O Gerenciamento Internet baseia-se no uso do protocolo SNMP (*Simple Network Managemen Protocol*), originalmente proposto no final dos anos 80 pelo IETF (*Internet Engineering Task Force* - grupo de padronização para o TCP/IP).

O SNMP define um protocolo para troca de informações de gerenciamento, um formato para representação das informações de gerenciamento, uma infra-estrutura para organizar sistemas distribuídos em sistemas de gerenciamento, e estruturas de bases de dados específicas. Em razão do SNMP englobar várias definições, o mesmo é comumente referenciado como “modelo SNMP”.

O modelo SNMP utiliza: o paradigma gerente-agente (*manager-agent*) para troca de solicitações e informações de gerenciamento; uma estrutura de informações de gerenciamento (SMI – *Structure Management Information*) para definição de objetos gerenciados; uma base de informações de gerenciamento (MIB – *Management Information Base*), organizada em uma árvore hierárquica, que especifica quais itens de dados de um elemento (objeto) gerenciado devem ser mantidos (armazenados) e quais operações (por exemplo, leitura e/ou escrita) são permitidas nesses dados; apenas as operações de gerenciamento *get-request*, *get-next-request*, *get-response*, *set-request* e *trap*; e um método muito primitivo, conhecido como comunidade, para sua segurança.

A popularidade do modelo SNMP no final dos anos 80 e início dos anos 90 deixou claras as suas deficiências, as quais fizeram com que fossem propostos os modelos SNMPv2

(SNMP versão 2) e SNMPv3 (SNMP versão 3), fazendo como que o modelo SNMP passasse, então, a ser referenciado como modelo SNMPv1 (SNMP versão 1).

Mesmo com os modelos SNMPv2 e SNMPv3, ainda existem deficiências. Por exemplo, a SMI não é orientada a objetos, ou seja, não suporta herança.

Muito trabalho tem sido realizado desde que o Gerenciamento Internet foi proposto. No entanto, o uso do SNMP para gerenciamento de sistemas, aplicações e serviços não foi explorado em profundidade e pode não ser apropriado.

3.2. Gerenciamento OSI

A necessidade de uma arquitetura de gerenciamento capaz de atender a uma enorme diversidade de elementos gerenciáveis existentes em uma rede fez com que a ISO (*International Organization for Standardization*) e o ITU-T (*International Telecommunications Union – Telecommunication Standardization Sector*) estabelecessem um conjunto de ferramentas e serviços para a supervisão (monitoramento) e controle de recursos de comunicação dentro de um ambiente OSI (*Open Systems Interconnection*), em adição ao modelo de referência OSI.

O modelo de informação do Gerenciamento OSI é baseado no paradigma de orientação a objetos (OO), o qual utiliza, intrinsecamente, os princípios de abstração, encapsulamento, herança e polimorfismo.

Conceitos do modelo de informação do Gerenciamento OSI são introduzidos no documento ISO10165-1, denominado *Management Information Model*, e detalhados no documento ISO10165-4, denominado *Guidelines for Definition of Managed Objects* (GDMO). Esses documentos tratam da Estrutura de Gerenciamento de Informação (SMI - *Structure of Management Information*) do Gerenciamento OSI, que descreve o cenário no qual objetos gerenciados podem ser definidos. Esse cenário inclui a definição do conjunto de operações que pode ser realizado sobre os objetos gerenciados e o comportamento desses objetos mediante a execução dessas operações.

Na SMI, objetos gerenciados são definidos como estruturas de dados usando a notação de sintaxe abstrata ASN.1 (*Abstract Syntax Notation One*). O GDMO apresenta extensões à linguagem ASN.1 no tratamento da sintaxe das definições de informações gerenciadas. Uma nova estrutura de linguagem - *template* - é introduzida para combinar definições. Uma classe de objetos gerenciados é um modelo ou uma *template* para objetos gerenciados que compartilham os mesmos atributos, as mesmas notificações, e as mesmas

operações de gerenciamento. As *templates* resumem os elementos que devem ser incluídos em uma definição de objeto gerenciado e definem também as ferramentas de notação que são recomendadas para uso com esta definição.

Diálogos entre entidades de aplicação de gerenciamento são realizados com ajuda de um protocolo da camada de aplicação, denominado *Protocolo de Informação de Gerenciamento Comum* (CMIP - *Common Management Information Protocol*).

O Gerenciamento OSI emprega funções de segurança como autenticação e controle de acesso e fornece as operações de gerenciamento *get, set, create, delete, action e event-report*.

Uma das principais desvantagens do Gerenciamento OSI é sua complexidade. Comparado ao Gerenciamento Internet, ele é muito difícil de ser implementado. Agentes de Gerenciamento OSI tipicamente requerem muito dos recursos do sistema.

4. Gerenciamento através da Web

O **Gerenciamento através da Web** ou **Gerência baseada na Web** é um enfoque promissor que pode fornecer uma solução verdadeiramente integrada, uniforme e simples para o gerenciamento de redes, sistemas, aplicações e serviços.

O gerenciamento baseado na Web fornece uma interface de usuário uniforme, denominada *browser* (navegador) Web, que é uma interface muito bem estabelecida tanto para a recuperação de dados como para execução de aplicações.

Tecnologias usadas no gerenciamento baseado na Web já foram utilizadas na Internet e são familiares mesmo aos usuários comuns. Além do mais, gerenciamento baseado na Web pode facilmente acomodar muitas ferramentas de gerenciamento padrão como o SNMP e o CMIP.

O pré-requisito para um gerenciamento integrado eficiente é que aplicações de gerenciamento possam ser acessadas de qualquer lugar e que agentes de gerenciamento estejam disponíveis em todos os recursos a serem gerenciados. Se navegadores Web forem usados como console de gerenciamento, aplicações de gerenciamento tornar-se-iam disponíveis de qualquer lugar.

Duas opções básicas para gerenciamento baseado na Web estão disponíveis [1]:

- os próprios recursos são instrumentados (manuseados) usando servidores Web e acessados diretamente através de um navegador Web. Neste caso, o HTTP é usado como protocolo de gerenciamento. Esta opção é referenciada como **técnica embutida para gerenciamento baseado na Web**. Vide Figura 2; e

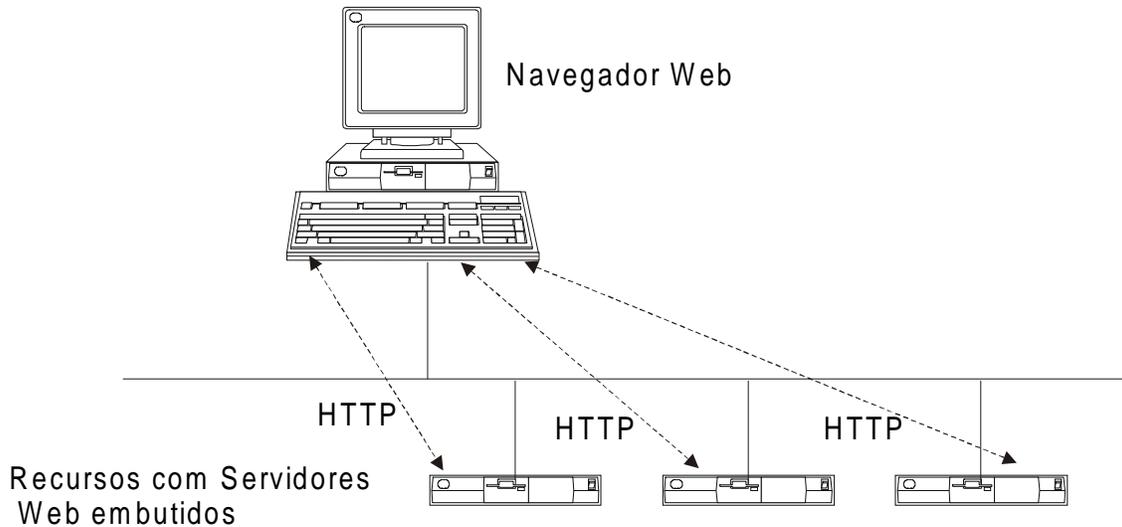


Figura 2: Técnica embutida para gerenciamento baseado na Web

- uma plataforma de gerenciamento é equipada como um servidor Web que permite a cada navegador Web ter acesso à plataforma. A comunicação com os recursos é feita através de um protocolo de gerenciamento padrão (SNMP ou CMIP) ou proprietário. Neste caso, o HTTP é usado apenas como protocolo GUI (*Graphical User Interface* – Interface Gráfica do Usuário). Esta opção é referenciada como **solução procuradora** (*proxy solution*). Vide Figura 3.

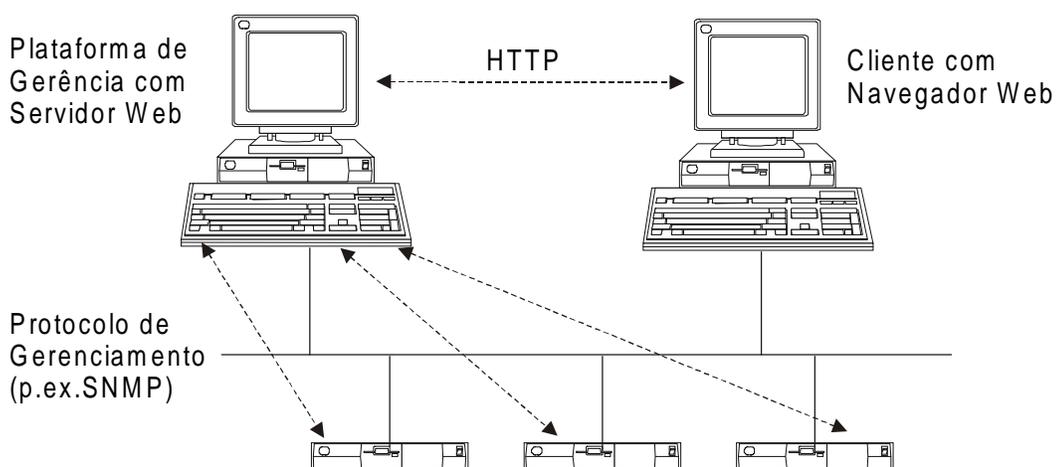


Figura 3: Solução Procuradora (*proxy solution*)

Hoje, muitos fabricantes estão equipando plataformas e ferramentas de gerenciamento isoladas com servidores Web. Assim, as mesmas podem ser acessadas através de um navegador Web. Ferramentas de gerenciamento geralmente comunicam-se com os recursos através de um protocolo de gerenciamento padrão ou proprietário. Isso pode ser visto como um caso especial de *proxy solution*.

O principal objetivo de se usar a **técnica embutida para gerenciamento baseado na Web** é a expectativa de custos mais baixos para soluções de gerenciamento. Para tarefas simples e ambientes pequenos, existe uma chance de plataformas de gerenciamento caras e complexas serem descartadas.

A principal motivação atrás da **solução procuradora** (*proxy solution*) é a proteção de investimentos em outras arquiteturas e produtos em razão dos padrões existentes. Esta opção é particularmente atrativa para ambientes nos quais a existência de facilidades para coleta, correlação e processamento de grandes volumes de dados é essencial.

É claro que as duas técnicas (**técnica embutida para gerenciamento baseado na Web** e **solução procuradora**) não são mutuamente exclusivas. Elas podem ser facilmente combinadas. Uma migração para uma dessas técnicas é indicada desde que a transição de consoles especializadas para navegadores Web não cause problemas para os operadores. Em princípio, soluções baseadas na Web podem ser facilmente escaláveis para infra-estruturas de tamanhos diferentes.

5. Arquiteturas de Gerenciamento baseadas na Web

Vários esforços têm sido propostos visando a utilização da tecnologia Web no gerenciamento de redes, sistemas, aplicações e serviços. A proposta WBEM (*Web-Based Enterprise Management*) [1, 2, 3, 5] é um resultado recente que envolve trabalhos conjuntos da Microsoft, Intel, Cisco, Compaq, BMC Software e outras empresas. Outro esforço, iniciado pela Sun, resultou no JMAPI (*Java Management API*) [1, 2, 3], onde classes Java são estendidas para implementar funções de gerenciamento.

5.1. JMAPI

A API de Gerenciamento Java (JMAPI – *Java Management API*) foi introduzida pela Sun para suportar o uso da linguagem Java na implementação de aplicações de gerenciamento. JMAPI é um rico conjunto de objetos e métodos para o desenvolvimento de soluções de gerenciamento de rede, sistemas, aplicações e serviços para redes

heterogêneas. JMAPI também pode ser considerada como uma extensão das classes básicas de Java.

A arquitetura de gerenciamento JMAPI (Figura 4) contém muitos elementos centrais da solução procuradora (*proxy solution*) para gerenciamento baseado na Web, descrita anteriormente. Mas, por outro lado, ela é mais que uma definição de uma arquitetura. Ela também especifica um modelo de referência de implementação detalhada e implementa componentes fundamentais na forma de bibliotecas de classes.

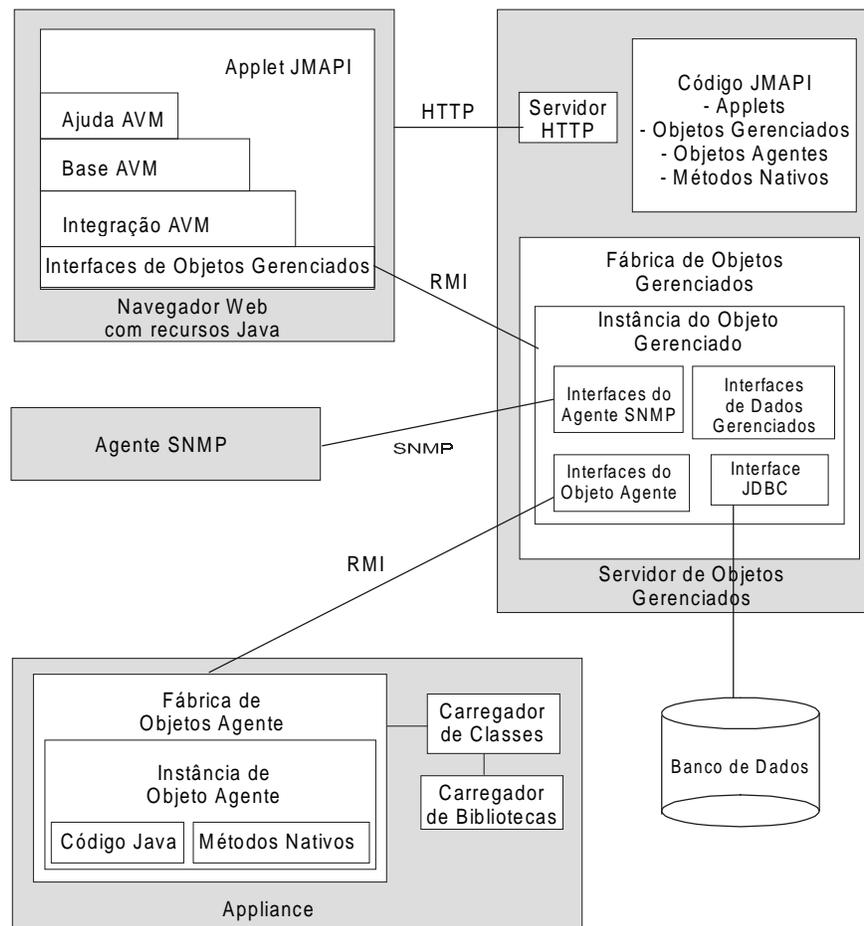


Figura 4: Arquitetura JMAPI

A arquitetura JMAPI foi primeiramente publicada em 1996. A versão 1.0 foi anunciada em 1998. Conceitualmente, a arquitetura JMAPI separa o sistema de gerenciamento em dois componentes: console de gerenciamento e plataforma de gerenciamento. O sistema gerente (*manager system*) é estruturado de acordo com o princípio cliente/servidor. A console de gerenciamento (Navegador Web com Recursos Java - *Java-enabled Web Browser*) assume o papel de cliente. A plataforma de gerenciamento (Servidor de Objetos Gerenciados - *Managed Object Server* –) assume o papel de servidor.

Em adição, a arquitetura JMAPI tem dois tipos diferentes de agentes, denominados *SNMP agents* e *appliances*. *SNMP agents* correspondem a agentes SNMP e *appliances* correspondem a agentes de gerenciamento implementados em Java.

Assim, a arquitetura JMAPI (Figura 4) consiste de quatro elementos fundamentais, apresentados com mais detalhes a seguir:

a) **Servidor de Objetos Gerenciados (*Managed Object Server*)**: também chamado de **Módulo de Execução Administrativo (*Admin Runtime Module*)**, representa a infra-estrutura básica da plataforma de gerenciamento. Seu núcleo consiste de um componente construtor denominado Fábrica de Objetos Gerenciados (*Managed Object Factory*) que permite as aplicações coordenarem a criação, manipulação e deleção de objetos gerenciados. A Fábrica de Objetos Gerenciados incorpora as seguintes interfaces para esse propósito:

- *Interfaces de Dados Gerenciados (Managed Data Interfaces) e Interface JDBC (Java Database Connectivity)*: interfaces que permitem acesso à base de dados com o objetivo de armazenar objetos gerenciados;
- **Interfaces do Objeto Agente (*Agent Object Interfaces*): interfaces para comunicação com agentes implementados em Java; e**
- *Interfaces de Agentes SNMP (*SNMP Agents Interfaces*)*: interfaces para comunicação com agentes que utilizam SNMP para acessar MIBs.

O **Servidor de Objetos Gerenciados** também armazena um código Java (*Java Code*) para implementação de aplicações de gerenciamento ou agentes que, se necessário, podem ser carregados nas máquinas virtuais Java das consoles de gerenciamento ou dos agentes. O código Java para aplicações de gerenciamento é usualmente *applets* Java que estão disponíveis nos servidores HTTP. O código Java para agentes é aplicações Java.

b) **Navegador Web com Recursos Java (Java-enabled Web Browser):** : corresponde a console de gerenciamento, constituindo a interface gráfica do usuário, onde o administrador emite operações de gerenciamento. Operações de gerenciamento podem ser invocadas através de um navegador Web ou através de uma aplicação isolada.

O *Java-enabled Web Browser* consiste de um Módulo de Visão Administrativa (*Admin View Module - AVM*), Interfaces de Objetos Gerenciados (*Managed Object Interfaces*) e um Navegador Web.

O AVM compreende classes chaves do lado do cliente para o desenvolvimento de *applets* baseados em JMAPI. A principal função dessas classes é fornecer uma interface de usuário e funcionalidades no nível de aplicação. As classes AVM são divididas em três partes: Ajuda AVM (*AVM Help*), Base AVM (*AVM Base*) e Integração AVM (*AVM Integration*). *AVM Help* fornece um ambiente de ajuda de propósito geral. Classes *AVM Base* são usadas para implementar um modelo usuário que constrói no navegador Web um estilo hipertexto de navegação. Classes *AVM Integration* fornecem integração entre classes *AVM Base* e Interfaces de Objetos Gerenciados (*Managed Object Interfaces*). *AVM Help* e *AVM Base* não são classes específicas de gerenciamento.

Interfaces de Objetos Gerenciados (*Managed Object Interfaces*) permitem acesso a objetos do Servidor de Objetos Gerenciados (*Managed Object Server*) através de RMI (*Remote Method Invocation* – Invocação de Métodos Remotos) [4]. Constituem *stubs* que podem ser usados para invocar métodos de gerenciamento remotos da Fábrica de Objetos Gerenciados (*Managed Object Factory*). Objetos gerenciados são usados para fornecer abstração de recursos.

O Navegador Web é um navegador comercial com recursos Java embutido como *Hot Java*, *Netscape* ou *Internet Explorer*. *Applets* que executam no navegador Web usam as classes AVM e as Interfaces de Objetos Gerenciados (*Managed Object Interfaces*) para executar operações de gerenciamento.

O **Navegador Web com Recursos Java (Java-enabled Web Browser)** possui uma máquina Java integrada (*integrated Java machine*) e fornece um ambiente operacional para aplicações que são *applets* Java. Esse ambiente permite que aplicações comuniquem-se com a plataforma de gerenciamento e junto com classes

AVM oferece mecanismos de coordenação e integração para diferentes aplicações executando ao mesmo tempo na console.

c) **Appliances**: *Appliances* correspondem a agentes de gerenciamento implementados em Java. Eles contêm uma função construtora para objetos gerenciados que são implementados por código Java e, em algumas circunstâncias, por partes de programas em código específico da arquitetura (métodos nativos). O código para objetos pode ser integrado dentro do agente ou carregado do Servidor de Objetos Gerenciados (*Managed Object Server*), se necessário; e

d) **Agentes SNMP**: agentes SNMP constituem agentes que utilizam o protocolo SNMP para interagir com os objetos gerenciados. O acesso a agentes SNMP é usualmente feito através das *SNMP Agents Interfaces* do Servidor de Objetos Gerenciados (*Managed Object Server*).

O modelo de informação de gerenciamento JMAPI é um refinamento do modelo de objetos Java e incorpora os princípios conhecidos de orientação a objetos. Ele separa as definições de interfaces nas quais herança múltipla é fornecida das implementações das interfaces somente com herança simples.

O modelo de comunicação do JMAPI inclui três protocolos de comunicação diferentes: HTTP, RMI e SNMP.

O HTTP é usado para carregar *applets* Java, que representam partes das aplicações de gerenciamento, de um Servidor de Objetos Gerenciados (*Managed Object Server*) para a console de gerenciamento (com navegador Web e máquina virtual Java).

Aplicações de gerenciamento usam RMI (*Remote Method Invocation* – Invocação de Métodos Remotos) para comunicar-se com objetos gerenciados no Servidor de Objetos Gerenciados (*Managed Object Server*). Por outro lado, esses objetos gerenciados usam RMI para acessar agentes implementados por código Java (*appliances*).

Para permitir integração com um grande número de recursos SNMP, a arquitetura permite que aplicações de gerenciamento acessem objetos gerenciados através do protocolo SNMP.

O objetivo da utilização de *applets* Java para implementar aplicações de gerenciamento é usar a capacidade de processamento disponível nas consoles de gerenciamento.

Execução de funções de gerenciamento não é somente delegada da plataforma de gerenciamento para os agentes, mas também para as consoles de gerenciamento. Isso fornece uma enorme flexibilidade desde que consideremos o arranjo dos módulos funcionais.

5.2.WBEM

Um consórcio de fabricantes chamado *Web-Based Enterprise Management Initiative* constituído pelas empresas Microsoft, Intel, Cisco, Compaq e outras, foi formado ao mesmo tempo que começou o desenvolvimento do JMAPI. O objetivo desse consórcio era desenvolver uma arquitetura aberta para o gerenciamento baseado na Web de toda uma infraestrutura corporativa de uma organização. A nova arquitetura, denominada gerenciamento de hipermídia (HMM - *Hypermedia Management*) mas referenciada como **arquitetura WBEM** (*Web-Based Enterprise Management*), deveria embutir diferentes técnicas de gerenciamento existentes, incluir a tecnologia Web, especialmente navegadores Web, e um protocolo de gerenciamento baseado no HTTP. Assim, a arquitetura WBEM foi projetada para:

- proporcionar o desenvolvimento de padrões industriais que permitissem aos administradores usarem qualquer navegador Web para gerenciar redes, sistemas, aplicações e serviços;
- permitir que soluções de gerenciamento fossem construídas cobrindo as áreas tradicionais de gerenciamento de configuração, gerenciamento de falhas, gerenciamento de contabilização, gerenciamento de desempenho, gerenciamento de segurança, gerenciamento de operações e planejamento;
- prover um modelo de dados que permitisse uma modelagem uniforme do ambiente a ser gerenciado; e
- atender às necessidades de uma grande conjunto distribuído de elementos de gerenciamento, fornecendo uma solução escalável.

Durante o curso do desenvolvimento da arquitetura WBEM, um protocolo de gerenciamento chamado HMMP (*Hypermedia Management Protocol* – Protocolo de Gerenciamento Hipermídia) foi especificado. Contrariamente às intenções originais, o HMMP não era baseado no HTTP, mas em protocolos de transporte, tendo muito pouco em comum com o HTTP.

Com as dificuldades em se desenvolver uma arquitetura de gerenciamento do nível pretendido, o DMTF (*Desktop Management Task Force*) [7] ficou com a responsabilidade de desenvolver a arquitetura WBEM. Com os trabalhos do DMTF, o objetivo original de se usar

o HTTP como protocolo de gerenciamento foi reestabelecido, sendo a proposta do HMMP suspensa. O DMTF também tratou de ratificar, como sendo um dos componentes principais da arquitetura WBEM, o CIM (*Commum Information Model* - Modelo de Informações Comum).

As partes principais da arquitetura WBEM são:

- CIM (*Commum Information Model* – Modelo de Informações Comum): que corresponde a um modelo orientado a objeto para descrição de informações de gerenciamento;
- MOF (*Managed Object Format* – Formato de Objeto Gerenciado): que corresponde a uma especificação sintática para objetos gerenciados modelados através do CIM; A notação MOF é uma linguagem *template*, comparável ao GDMO ou a SMI da Internet; e
- uma especificação para mapeamento do CIM em XML (*Extensible Markup Language*) [8], uma linguagem escrita em DTD (*Document Type Definition* - Definição de Tipo de Documento) que pode ser usada para representar classes e instâncias do CIM.

Através do mapeamento do CIM em XLM, qualquer navegador Web com implementação da linguagem XML pode ser usado para mostrar e capturar informações de gerenciamento. Nenhum protocolo de gerenciamento especializado é necessário nesse contexto. O HTTP é usado para transferir informações de gerenciamento.

A seguir, são vistas com mais detalhes características do CIM e da representação do CIM em XML.

5.2.1. CIM

O principal componente da arquitetura WBEM é o CIM, previamente denominado HMMS (*Hypermedia Management Schema* – Esquema de Gerenciamento Hipermissão).

Como dissemos anteriormante, o CIM corresponde a um modelo orientado a objeto para descrição de informações de gerenciamento. Através do CIM, aspectos e comportamentos significantes de objetos gerenciados são modelados usando conceitos de orientação a objetos.

A definição de um modelo de informações de gerenciamento completamente novo foi motivada pelo requisito central de gerenciamento integrado sem perda de informação entre as partes envolvidas. Muitos dos modelos de informação disponíveis atualmente não podem ser facilmente mapeados em outros. Quando o mapeamento é possível, as transições resultam, inevitavelmente, em perda de informação considerável. Com o CIM, a idéia é ter-se um

modelo guarda-chuva único que abranja os modelos existentes, minimizando perda de informação entre os mesmos.

O CIM fornece um conjunto de classes com propriedades e associações, sendo visto como uma ferramenta conceitual na qual é possível organizar informações sobre um ambiente gerenciado. Por ser um modelo orientado a objetos, o CIM é desvinculado de linguagem de programação. Fazendo parte do CIM, tem-se:

- um modelo de informação básica referenciado como metaesquema;
- uma sintaxe para a descrição de objetos gerenciados (MOF - *Managed Object Format* – Formato de Objeto Gerenciado); e
- duas camadas de classes genéricas de objetos gerenciados, chamadas modelo núcleo (*core model*) e modelo comum (*common model*).

O metaesquema é definido com ajuda da linguagem (*Unified Modeling Language* – Linguagem de Modelagem Unificada) [9] e contém classes, métodos, qualificadores (*qualifiers*), associações, referências, classes de eventos e esquemas. **Classes** caracterizam conjunto de objetos com propriedades comuns. Somente herança simples é possível entre as classes. **Métodos** definem procedimentos que podem ser executados em objetos de uma classe. **Qualificadores** (*qualifiers*) são usados para especificar características adicionais das classes ou propriedades e métodos das classes. Exemplos incluem [1]: opções de acesso (*READ*, *WRITE*), informações de atributo de mapeamento para uma MIB existente (*MappingStrings*), classificação de atributos (*names*) como chaves (*Key*) para entidades e identificação de classes que não podem ser instanciadas (*ABSTRACT*). **Associações** são classes (ou instância delas) que representam relacionamento entre dois ou mais objetos. Elas contêm pelo menos duas referências. **Referências** são propriedades especiais. **Classes de eventos** são usadas para definir tipos diferentes de notificação de eventos. **Esquemas** são grupos de elementos combinados para propósito administrativo.

As classes concretas de objetos gerenciados definidas estão estruturadas em três camadas hierárquicas denominadas [1]: modelo núcleo (*core model*), modelo comum (*common model*) e extensões de esquema. O **modelo núcleo** fornece um pequeno número de classes, associações, e propriedades usáveis em todos os domínios de gerenciamento como ponto de partida para todos refinamentos. Exemplos incluem classes como *ManagedSystemElement*, *PhysicalElement*, *LogicalElement*, *System* e *Service*. O **modelo comum** refina o modelo núcleo em classes não dependentes de tecnologias e implementações específicas, mas que são concretas o bastante para servirem como base para muitas aplicações

de gerenciamento. **Extensões de esquema** refinam o modelo comum para tecnologias especiais. Por exemplo, a Microsoft especificou uma extensão de esquema, denominada *Win32Schema*, que especializa as classes do modelo comum para *Windows 95/98* e *Windows NT*.

5.2.2. Representação do CIM em XML

O MOF fornece uma representação textual de informações de gerenciamento modeladas usando CIM. No entanto, essa representação sozinha não é suficiente para transferir informações de gerenciamento em ambientes heterogêneos. Para transferir informações de gerenciamento em ambientes heterogêneos, um mapeamento da representação para um protocolo de comunicação é necessário. Como mencionado antes, nos estágios iniciais da iniciativa WBEM planejou-se introduzir um protocolo de gerenciamento completamente novo para esse propósito. Quando o DTMF ficou com a responsabilidade de desenvolver a arquitetura WBEM, o objetivo original de usar o HTTP como protocolo de gerenciamento foi reestabelecido. Isso levou a especificação de um mapeamento do CIM para XML.

XML [8] é um subconjunto da linguagem SGML (*Standardized Generalized Markup Language*), usada para representar estruturas de dados (como as informações de gerenciamento) em forma textual. Em XML, um documento pode opcionalmente ter uma descrição da sua gramática em anexo. A gramática para um documento XML é descrita usando um mecanismo conhecido como DTD (*Document Type Definition* - Definição de Tipo de Documento). A DTD descreve os elementos permitidos num documento XML. Um documento que é estruturado de acordo com as regras definidas na especificação XML é dito bem formado (*well formed*). Além de bem formado, um documento XML pode ser válido. Um documento XML válido deve conter uma DTD, e a gramática do documento deve estar de acordo com o que foi especificado na DTD.

Com o objetivo de usar a XML para representar informações de gerenciamento, um vocabulário XML (por exemplo, uma DTD para classes e instâncias do CIM) deve ser definido.

Documentos XML não necessariamente contém informações sobre captura de dados. Isso pode ser alcançado com o uso de planilhas de estilo XSL (*Extensible Style Language*), que podem ser usadas tanto para capturar informações como para transformá-las em outros formatos. Qualquer número de planilhas de estilo XSL pode ser associado com um

documento XML. Por exemplo, planilhas de estilo XSL podem apresentar graficamente informações ou fornecer uma transformação para MOF.

Enfim, a definição de uma DTD para o CIM junto com as capacidades do XSL fornecem uma maneira de comunicar informações de gerenciamento do CIM em ambientes heterogêneos através do HTTP. XSL pode ser usada como uma forma padrão para capturar informações de gerenciamento.

6. Considerações Finais

Nos últimos anos, estruturas de comunicação têm sido baseadas em protocolos Internet e navegadores Web têm sido usados como terminais de usuários gráficos por serem fáceis de serem usados, por estarem disponíveis em várias plataformas e por permitirem acesso, de qualquer sistema final, às aplicações e aos dados corporativos de uma organização. Por exemplo, terminais 3270 têm sido substituídos por navegadores Web e servidores Web têm sido acoplados a aplicações herdadas como objetivo de serem acessadas via HTTP.

A usabilidade intuitiva e simples de interfaces de usuários de navegadores Web simplifica a tarefa de gerenciamento. Além do fácil uso e de estarem disponíveis em diversas plataformas, navegadores Web são atrativos para propósito de gerenciamento por oferecer um ambiente de execução para programas independentes da plataforma. Programas (por exemplo, *applets* Java), se necessário, também podem ser carregados/instalados pelo servidor. As principais vantagens disso são:

- nenhum software tem que ser instalado nas consoles de gerenciamento e arquiteturas de hardware e software usadas em clientes não precisam ser levadas em consideração. Diferenças nas plataformas dos navegadores e nas versões dos ambientes computacionais Java, por exemplo, podem decrecer essa vantagem;
- se as funções de gerenciamento forem implementadas como *applets* Java, plataformas de gerenciamento podem delegar a execução dessas funções para as consoles de gerenciamento. Gerenciamento baseado em SNMP raramente permite que funções sejam delegadas aos agentes; conseqüentemente, a carga de trabalho criada por *polling* é freqüentemente muita alta; e
- muitos navegadores já têm ORBs (*Object Request Brokers*) embutidos. Os ORBs permitem que *applets* Java acessem diretamente objetos de um servidor CORBA.

A tecnologia Web tem revolucionado o uso da Internet nos últimos anos com sua maneira simples mas poderosa de recuperar vários tipos de dados, incluindo textos, gráficos, imagem, voz e vídeo.

O gerenciamento baseado na Web é um enfoque promissor que pode fornecer um enfoque verdadeiramente integrado para o gerenciamento de redes, sistemas, aplicações e serviços.

A linguagem Java tem alterado a maneira como as pessoas desenvolvem aplicações na Internet. O ambiente de *runtime* do Java soluciona vários problemas básicos de programação para a Internet. Como a Internet consiste de muitas plataformas diferentes, é importante escrever programas independentes de plataforma. Antes do Java, os programadores da Internet compilavam programas para cada plataforma. Software desenvolvido usando a linguagem Java é portátil através de várias plataformas e pode ser distribuído e acessado por navegadores Web. *Downloading* transparente e rápido através de um navegador Web com recursos Java constitui uma alternativa atrativa.

Aplicando as tecnologias Web e Java no gerenciamento de redes, sistemas, aplicações e serviços, pode-se fornecer soluções para os problemas correntes de consistência, segurança, confiabilidade e segurança.

Hoje, muitos fabricantes de sistemas de gerenciamento estão trabalhando com soluções baseadas na Web. Em adição a muitas soluções puramente proprietárias, duas arquiteturas abertas para gerenciamento baseado na Web existem: *Java Management API* (JMAPI) e *Web-Based Enterprise Management* (WBEM). A arquitetura JMAPI concentra-se no uso de APIs para suportar o desenvolvimento de agentes e aplicações. Seu objetivo principal é a implementação rápida e eficiente de agentes de gerenciamento reusáveis e portáteis. O propósito da arquitetura WBEM é proporcionar o desenvolvimento de padrões industriais que permitam aos administradores usarem qualquer navegador Web para gerenciar redes, sistemas, aplicações e serviços. A arquitetura WBEM concentra-se no desenvolvimento de um modelo de informação completamente novo para modelar aspectos e comportamento do ambiente a ser gerenciado. Como podemos observar, as arquiteturas WBEM e JMAPI não são diretamente tecnologias competitivas, pois possuem diferentes filosofias. Existe, no entanto, áreas onde as duas sobrepõem-se.

O gerenciamento baseado na Web pode facilmente acomodar protocolos padrões de gerenciamento de rede como o SNMP e o CMIP. Muitas soluções baseadas na Web podem, por exemplo, mapear MIBs SNMP para HTML, CGI, Java e código C que executam em

servidores Web. No entanto, administradores ainda são os responsáveis por projetar aplicações e por decidir aspectos chaves como “onde um banco de dados deve ficar localizado”.

A técnica mais comum de gerenciamento baseado na Web é adicionar um servidor Web ao dispositivo gerenciado junto com um software para mapear dados coletados do agente (*agent*) do dispositivo para o servidor. É possível, então, a realização de consultas de estado do dispositivo de qualquer localização através de um navegador Web.

Alguns vendedores usam *scripts* CGI ou mecanismos proprietários para ligar navegadores Web a gerentes (*managers*). O uso de gerentes junto com navegadores Web ainda é problemático. A utilização de Java, mesmo com seus mecanismos de segurança embutidos, apresenta efeitos colaterais. Por exemplo, *applets* não podem ler ou escrever arquivos no disco do cliente. Desempenho também continua sendo uma questão em aberto. Atrasos associados com a conversão de relatos SNMP (ou outro formato) para páginas HTML podem ser significativos.

Apesar das desvantagens técnicas correntes, acreditamos no gerenciamento baseado na Web. Usando a Web como uma interface de integração, uma apresentação unificada de todas as informações de gerenciamento é fornecida, e um estilo consistente de invocar operações de gerenciamento é seguido.

Referências Bibliográficas

- [1] Heinz-Gerd Hegering, Sebastian Abeck and Bernhard Neumair. *Integrated Management of Networked Systems – Concepts, Architectures, and Their Operation Application*. Morgan Kaufmann Publishers, Inc. San Francisco, California. 1998.
- [2] Raouf Boutaba, Karim El Guemhioui and Petre Dini. *An Outlook on Intranet Management*. IEEE Communications Magazine. Vol. 35. No. 10. Pages 92-99. October 1997.
- [3] James Won-Ki Hong, Ji-Young Kong, Tae-Hyoung Yun, Jon-Seo Kim, Jong-Tae Park and Jong-Wook Baek. *Web-Based Intranet Services and Network Management*. IEEE Communications Magazine. Vol. 35. No. 10. Pages 100 – 110. October 1997.
- [4] David G. Messerschmitt. *Networked Applications: A Guide to the New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc. San Francisco, California. 1999.
- [5] J. Patrick Thompson. *Web-Based Enterprise Management Architecture*. IEEE Communications Magazine. Vol. 36. No. 3. Pages 80-86. March 1998.
- [6] <http://java.sun.com>
- [7] <http://www.dmtf.org>.
- [8] <http://www.w3.org.XML>.
- [9] <http://www.rational.com/uml/documentation.html>.
- [10] Michael D. Thomas, Pratik R. Patel, Alan D. Hudson e Donald A. Ball Jr. *Programando em Java para a Internet – Um Guia para Criar Aplicações Dinâmicas e Interativas*. Makron Boorks. 1997.
- [11] Peter van der Liden. *Just Java – Guia Autorizado Sunsoft Press*. Makron Books. 1998.
- [12] W. Richard Stevens. *TCP/IP Illustrated, Volume 1 - The Protocols*. Addison Wesley. 1994.
- [13] William Stallings. *SNMP, SNMPv2, and CMIP – The Practical Guide to Network Management Standards*. Addison Wesley. 1993.
- [14] William Stallings. *SNMPv3: A Security Enhancement for SNMP*. IEEE Communications Surveys. Fourth Quater 1998. Volume 1. Number 1.
- [15] José Helvécio Teixeira Júnior, Jacques Philippe Sauvé, José Antão Beltrão Moura e Suzana de Queiroz Ramos Teixeira. *Redes de Computadores – Serviços, Administração e Segurança*. Makron Books. 1999.