

Uma Ferramenta para Geração de Gateways e Scripts em um Ambiente CORBA

Carlos Maurício S. Figueiredo, Eduardo Freire Nakamura,
Antonio A.F. Loureiro

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
{carlosf, nakamura, loureiro}@dcc.ufmg.br

Resumo

O gerenciamento de redes de computadores e telecomunicações tem sido fundamental devido à necessidade de rápida integração de novos, modernos e complexos equipamentos de telecomunicações de múltiplos fabricantes ao sistema de gerenciamento. Além disto, é necessário que o ambiente de gerência de redes e serviços seja eficiente e dinâmico para facilitar a integração de várias tecnologias e softwares de telecomunicações. A tecnologia CORBA vem sendo muito utilizada para a integração desse sistemas, trazendo muitas facilidades e vantagens. Assim, este artigo dá um enfoque na construção de gateways para a integração de recursos gerenciados de outros modelos de gerência ou proprietários, e propõe uma ferramenta para a geração desses gateways de forma facilitada.

Abstract

The management of computer and telecommunication networks plays an important role in an environment where new, complex, and different network elements and services are added to the plant. In this heterogeneous scenario, CORBA has been applied in the integration and rapid deployment of such elements and services. This paper presents a tool that builds automatically CORBA gateways that integrate elements and services to a management platform. Furthermore, the tool supports the management by delegation paradigm.

Palavras-chave: Gerenciamento de redes, Gateway CORBA, Gerenciamento por delegação

1 Introdução

Diante da necessidade de integração de complexos equipamentos de múltiplos fabricantes ao ambiente de gerência, alguns modelos foram adotados para realizar o gerenciamento de redes e sistemas, os mais tradicionais são o SNMP e o CMIP [2, 6]. Porém, o SNMP não atende os requisitos de tarefas de gerenciamento mais complexas e o CMIP, devido a sua

complexidade e lento processo de padronização, não obteve a mesma aceitação que o SNMP. CORBA (Common Object Request Broker Architecture) [5, 17, 19, 20, 22] se tornou uma alternativa interessante para este propósito, pois permite realizar funções que nem o SNMP e nem o CMIP realizam.

O padrão CORBA é baseado em objetos distribuídos que vem sendo largamente adotado, e que pode ser encontrado em vários sistemas de telecomunicações [14]. Isto se deve ao fato de ser uma arquitetura que permite a evolução de um sistema proprietário para um sistema multiplataforma, por permitir a integração com outros softwares de telecomunicações e por possibilitar a transparência da informação sem a preocupação de saber em que plataforma de software ou hardware ela reside e onde está localizada na rede de gerência.

O padrão foi desenvolvido pelo OMG (Object Management Group) [12], que provê um modelo de interação de objetos em um ambiente distribuído. O OMG é um consórcio internacional de software, estabelecido em 1989 e suportado por mais de 700 empresas envolvidas com tecnologia da computação, incluindo operadoras e fornecedores de serviços de telecomunicações.

Muitos benefícios podem ser adquiridos com um sistema de gerenciamento baseado em CORBA (tais como portabilidade, escalabilidade e interoperabilidade) [3], podendo este coexistir com outros modelos e aplicações legadas de gerência. Isto é desejável devido ao fato de muitos investimentos terem sido realizados em modelos antigos e essa base instalada poder ser muito grande. A coexistência de um sistema baseado em CORBA com outros modelos de gerência, com aplicações legadas e com recursos gerenciáveis com interfaces proprietárias (ex. centrais telefônicas) deve se dar por intermédio de gateways CORBA. Estes gateways traduzem as requisições e as informações do domínio dos recursos gerenciados para o ambiente CORBA, tornando transparente para as aplicações de gerenciamento suas particularidades.

Nesse contexto, este trabalho apresentará um estudo de aplicação de gateways e, a partir daí, apresentará uma proposta de uma ferramenta de geração de gateways e scripts em um ambiente CORBA de forma a facilitar o trabalho dos desenvolvedores de sistemas de gerência.

Este trabalho está organizado da seguinte forma. As seções 2 e 3 apresentam os módulos de geração de gateways e scripts, respectivamente, com a descrição do projeto de cada parte. Finalmente a seção 4 apresenta as conclusões deste trabalho.

2 O módulo de geração de gateways

Esta seção descreve o módulo responsável pela geração de gateways CORBA.

2.1 Breve análise da aplicação de gateways CORBA

Gateways são componentes de software, ou objetos, que integram ao ambiente de gerência baseado em CORBA, elementos de rede de outros modelos de gerência, tais como SNMP e CMIP, ou proprietários, como ocorre com algumas centrais telefônicas.

Esses gateways devem traduzir informações e requisições entre o domínio do elemento de rede e o ambiente de gerência CORBA, tornando transparentes para as aplicações de gerenciamento as particularidades dos elementos de rede.

Os gateways podem ser implementados de forma a acrescentar funcionalidades ao elemento de rede que não existiam antes, devido ao legado, ou funcionalidades importantes para o ambiente de gerência, tais como informações pertinentes ao gerenciamento de objetos CORBA. Esses também podem ser implementados para usar outros recursos do ambiente de gerência, tais como os Serviços CORBA: Event Service [11], usado para emissão de falhas, muito importante para a gerência do elemento de rede; e Persistent State Service [13], que pode ser usado para manter estados do gateway ou informações de histórico (Log) de ações executadas.

Para atender requisições de aplicações de gerência, esse gateways devem ser servidores CORBA, ou seja, objetos disponíveis no ORB. O processo de criação de servidores CORBA consiste dos seguintes passos:

1. Definição da interface IDL do gateway. Descrição das operações acessíveis do objeto;
2. Compilação da IDL: Para a geração do stub, usado pelas aplicações que acessarão o gateway, e skeleton, usado para a implementação do objeto;
3. Implementação do objeto: Implementação das operações definidas na sua IDL;
4. Criação da Aplicação Servidora: Aplicação que tornará o objeto implementado disponível no ORB para receber requisições.

A figura 1 mostra a disposição de gateways em um ambiente de gerência baseado em CORBA. Nela podemos perceber a existência de gateways para diversos elementos de redes, entre eles agentes SNMP e CMIP que já possuem um mapeamento especificado para CORBA pelo JIDM [7, 8].

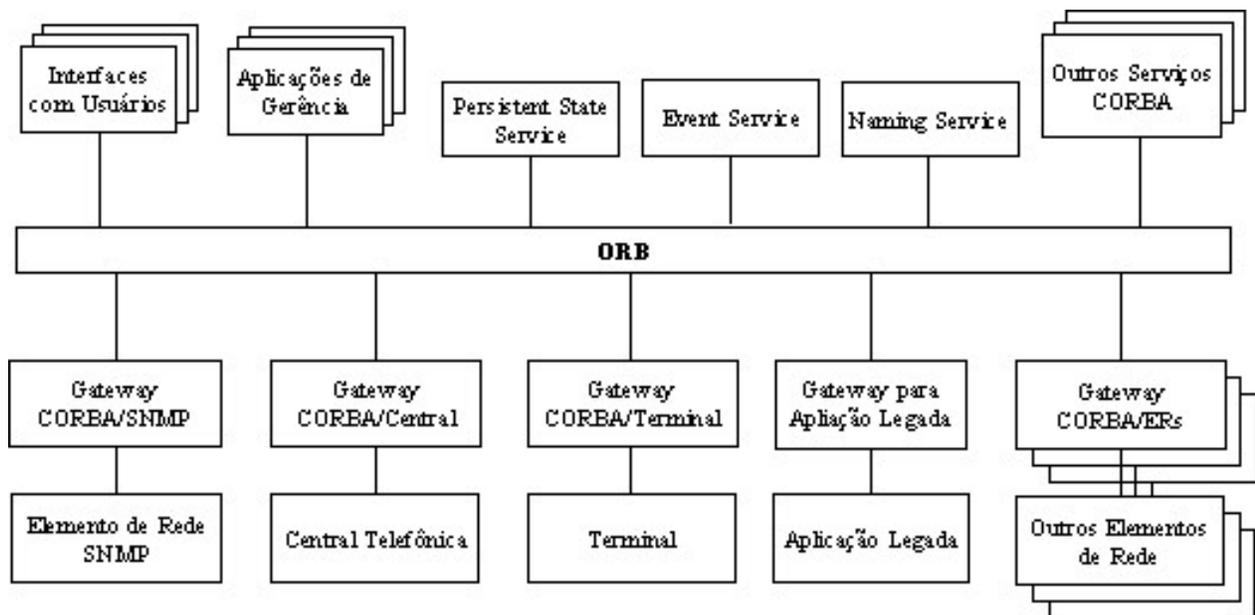


Figura 1: Disposição de gateways em um ambiente de gerência baseado em CORBA.

A partir do momento que um elemento de rede está disponível no ambiente através do uso de um gateway, tem-se o seu acesso de forma homogênea pelas aplicações de gerência e de interface, e pode-se reaproveitar outros componentes do ambiente, como os serviços CORBA, para implementar algumas funcionalidades dos objetos ou para realizar a interação entre eles. Por exemplo, um elemento de rede pode mandar mensagens de falha para uma aplicação de gerência de forma desacoplada através do uso do Serviço de Eventos, e a aplicação de gerência, por sua vez, utiliza o serviço de persistência para armazenar um histórico da ocorrência de falhas.

Concluindo, podemos observar a vantagem de se integrar elementos de rede ao ambiente de gerência baseado em CORBA, com a possibilidade de se acrescentar funcionalidades a ele, inclusive através do uso de serviços do CORBA.

2.2 Descrição da ferramenta

A ferramenta proposta implementada, realiza todos os procedimentos para a criação do gateway de forma semi-automática, e disponibilização deste no ORB, conforme descrito anteriormente. Basicamente, ela permite que se escolha de uma lista o elemento de rede para o qual se deseja obter o gateway, que sejam acrescentadas funcionalidades a ele a partir de uma biblioteca de funções, e permite que se defina um procedimento para emissão de eventos ao ambiente através do uso do Event Service, muito utilizado para a monitoração de falhas, que é uma característica muito importante de gerência de redes.

Inicialmente, deve-se considerar uma linguagem de programação alvo para a ferramenta, pois o gateway a ser gerado deve ter sua implementação e eventuais modificações feitas pelo desenvolvedor nessa linguagem. A linguagem escolhida, tanto para a implementação da ferramenta quanto para a geração do gateway foi Java [15, 18] devido à sua excelente capacidade de integração com CORBA e sua portabilidade.

Vale ressaltar que a escolha do Java para o gateway implica na necessidade de se usar um Java ORB para sua disponibilização. Existem vários Java ORBs, tais como o Borland Visibroker for Java [1] e o OpenGroup OpenORB [4], mas na implementação realizada foi usado este último, que é um ORB de código aberto, satisfaz a especificação CORBA 2.3 e possui vários Serviços implementados. Como a ferramenta permite que se altere características do código do gateway a ser gerado, conforme será mostrado posteriormente, a escolha do ORB não impactará no seu funcionamento.

As funcionalidades da ferramenta estão descritas a seguir

Inclusão de classes auxiliares ao gateway. Classes auxiliares são classes Java que podem implementar alguma funcionalidade específica, como por exemplo alguma função estatística ou uma biblioteca de acesso a um elemento de rede específico. Os métodos dessas classes desejados para se tornarem disponíveis na interface IDL do gateway devem ser marcados.

A figura 2 mostra como são incluídas essas classes ao gateway, lembrando que o desenvolvedor precisa definir o nome do objeto que será instanciado de cada classe no código gerado.

A ferramenta é extensível por permitir que sejam inseridas nela, novas classes auxiliares para se tornarem disponíveis para a criação de gateways. Assim, a partir do momento

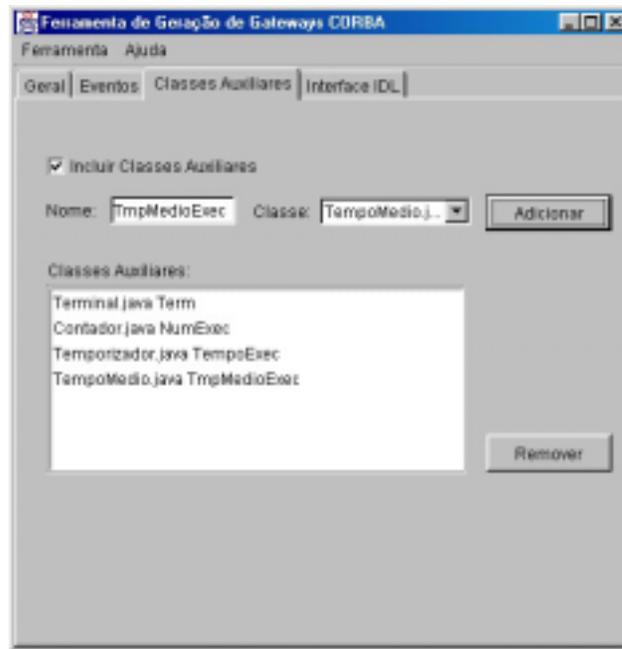


Figura 2: Inclusão de classes auxiliares no gateway.

que um desenvolvedor criou uma classe para acesso a um elemento de rede específico com uma interface proprietária, por exemplo, ele pode disponibilizá-la para a geração de outros gateways para esse elemento de rede.

Definição do Procedimento de Emissão de Eventos. Cada elemento de rede ou ambiente de gerência pode exigir uma estrutura de evento diferente, ou seja, os campos de informação do evento podem variar. A ferramenta gera todo o procedimento de emissão de eventos usando o Serviço de Eventos do CORBA em uma classe, a partir da descrição da sua estrutura pelo desenvolvedor.

Esse procedimento é baseado em um arquivo de modelo (template) que contém uma classe que já abstrai todos os detalhes para o uso do serviço CORBA. Esse arquivo pode ser modificado por um desenvolvedor mais especializado para usar abordagens diferentes de emissão de eventos, como por exemplo os métodos Push ou Pull definido pelo Serviço CORBA.

A estrutura do evento é descrita com os tipos básicos do Java, pois quando o desenvolvedor for implementar a interação dos objetos do gateway, ele programará a emissão de eventos nessa linguagem.

A figura 3 mostra como são definidos os campos do evento que será lançado pelo gateway.

Operações definidas pelo desenvolvedor. Nem todas as funcionalidades desejadas para o gateway podem estar disponíveis na ferramenta, assim, o desenvolvedor pode querer implementar suas próprias operações.

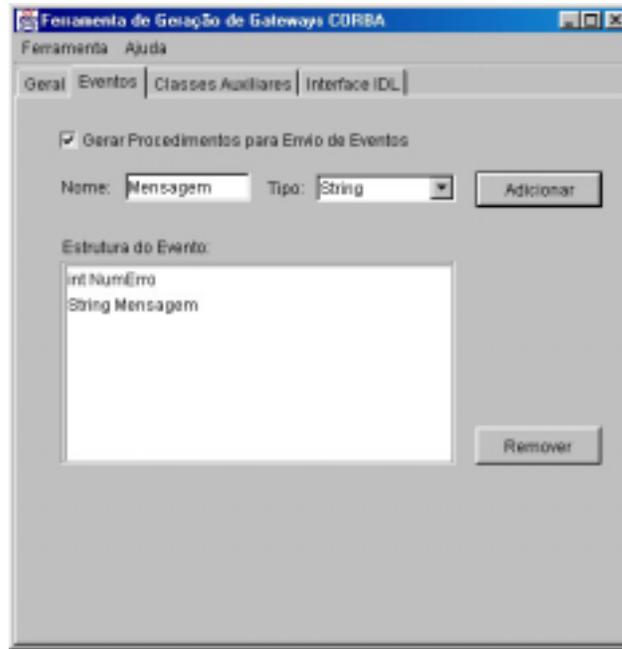


Figura 3: Definição da estrutura do evento do gateway.

A ferramenta permite que o desenvolvedor inclua suas próprias operações na interface IDL do gateway, e essas são descritas por ele com os tipos do Java, pois será com essa linguagem que a implementação dessas operações será realizada. A figura 4 mostra como são definidas essas operações.

Geração do gateway e da aplicação servidora. Após a inclusão de todas as funcionalidades desejadas para o gateway, conforme descrito nos itens anteriores, a ferramenta gerará todo o código possível de ser gerado automaticamente para o gateway.

Será gerada a interface IDL do gateway com todas as operações provenientes das classes auxiliares inseridas e com as operações definidas pelo desenvolvedor. Os tipos do Java serão convertidos automaticamente para os tipos IDL conforme o mapeamento descrito em [17, 21]. Em seguida, a IDL será compilada gerando o stub e o skeleton.

Após a compilação da IDL, o código do gateway deve ser implementado. Basicamente a ferramenta já realiza os seguintes procedimentos:

- Insere código para a instanciação de todas as classes inseridas pelo desenvolvedor, que são as classes auxiliares e a classe com procedimentos para emissão de eventos;
- Insere a declaração de todas as operações que devem ser implementadas, que são as provenientes das classes auxiliares e as definidas pelo desenvolvedor;
- No corpo das operações provenientes das classes auxiliares, a ferramenta já inclui a chamada dos métodos das classes instanciadas correspondentes.

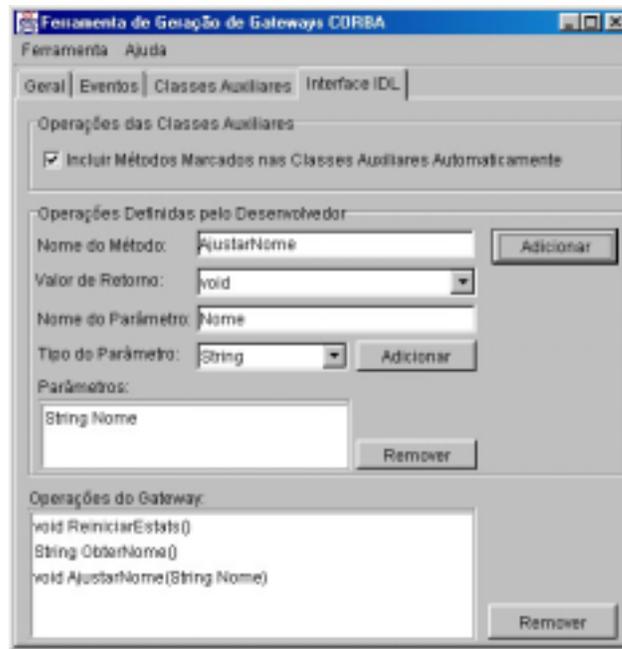


Figura 4: Definição de operações definidas pelo desenvolvedor.

Veja que alguma implementação manual é necessária, e basicamente consiste da implementação das operações definidas pelo desenvolvedor e interação entre as diversas classes, ou funcionalidades, inseridas no gateway. Esse trabalho só não precisará ser realizado se o gateway for composto de apenas uma classe que já implementa todas as suas funcionalidades, neste caso a ferramenta simplesmente funciona como se estivesse encapsulando a classe inserida em um objeto CORBA.

A geração da aplicação servidora do gateway também é baseada em um arquivo de modelo (template) que contém código para todas as inicializações do ORB necessárias e disponibilização do gateway implementado. Novamente, é possível a um desenvolvedor mais especializado realizar modificações nesse arquivo para alterar características da disponibilização do gateway, como por exemplo inclusão do uso do Serviço de Nomes.

A figura 5 mostra a interface da ferramenta para a entrada dessas informações, início e acompanhamento, através de mensagens mostradas em uma caixa de texto, dos processos de geração.

A figura 6 ilustra todo processo de geração de gateways com a ferramenta proposta.

2.3 Benefícios conquistados com o uso da ferramenta

Um dos benefícios mais visíveis obtidos da ferramenta foi tornar o processo de criação de gateways mais fácil e rápido através da eliminação do trabalho repetitivo relativo à criação de objetos CORBA. Além disso, a ferramenta permite ao desenvolvedor agregar ao gateway novas funcionalidades, sejam elas funções obtidas de uma biblioteca de classes com funções estatísticas ou específicas para um elemento de rede, ou ainda, funcionalidades obtidas da

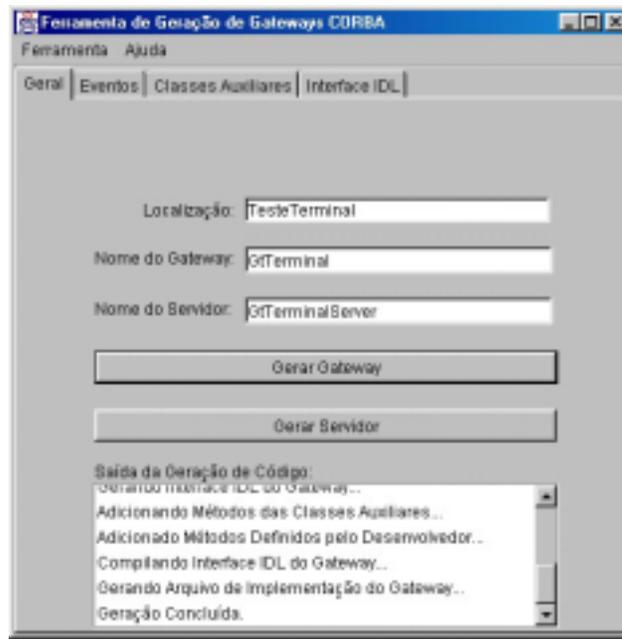


Figura 5: Geração do gateway e sua aplicação servidora.

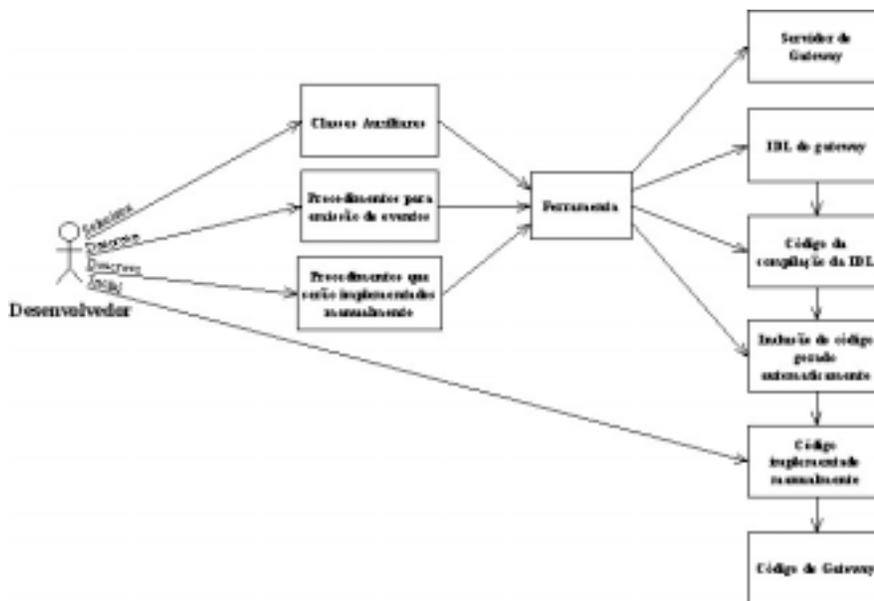


Figura 6: Processo de geração de gateways CORBA.

reutilização de serviços CORBA.

A ferramenta é extensível, pois permite que novas funcionalidades desenvolvidas sejam agregadas a ela para a geração de gateways. A ferramenta mostra-se versátil permitindo a mudança das características do código a ser gerado, inclusive detalhes referentes ao CORBA.

Isso é feito através do uso de arquivos de modelo (template), e alterações desses podem ser restritas a um desenvolvedor mais especializado.

É abstraído do desenvolvedor mais simples, detalhes CORBA de toda a descrição do gateway e das funcionalidades agregadas por ele, tornando o processo de geração de gateways menos complexo.

3 O módulo de geração de scripts

Esta seção descreve o módulo responsável pela geração de scripts de gerenciamento baseado no paradigma de gerenciamento por delegação.

3.1 Delegação de scripts

A delegação de scripts possui um gerente de delegação (Delegador) que transfere, para um determinado agente de delegação (gateway CORBA-SCRIPT), um conjunto de scripts de gerenciamento que descrevem as ações de gerência que devem ser executadas pelo agente remoto. A seguir, ele solicita que os scripts sejam executados *in loco*. Um interpretador é responsável pela execução dos scripts cujo resultado é enviado de volta para o gerente de delegação.

O ambiente de delegação e execução de scripts pode apresentar pelo menos duas configurações. Na primeira, o interpretador de scripts se encontra acoplado ao gateway CORBA-SCRIPT (figura 7). Nesta abordagem, o gateway está limitado a executar apenas os scripts da linguagem conhecida pelo interpretador. Assim, para se executar scripts de diferentes linguagens, é necessário que sejam embutidos os interpretadores específicos no gateway, ou que sejam desenvolvidos um gateway para cada linguagem script. A vantagem desta abordagem é que o gateway pode monitorar e controlar diretamente a execução do script. Na segunda, o interpretador não faz parte do gateway CORBA-SCRIPT (figura 8). Para executar um script, o gateway faz uma requisição ao sistema operacional que acionará o interpretador apropriado. Neste caso, o gateway, não possui conhecimento algum da linguagem script podendo executar qualquer script desde que exista um interpretador no sistema operacional. Definitivamente, esta abordagem é mais flexível que a anterior pois pode fazer uso de qualquer interpretador disponível. Contudo, nesta configuração não permite que o gateway controle e monitore diretamente a execução do script necessitando fazer sempre requisições ao sistema operacional.

3.2 Arquitetura proposta

A arquitetura é composta basicamente por um Delegador responsável por um ou mais gateways CORBA-SCRIPT. Cada gateway CORBA-SCRIPT é responsável pelo gerenciamento de uma sub-rede diferente (figura 9). Isto possibilita a redução do tempo de desenvolvimento e a personalização de Gerentes promovendo a integração dos diferentes modelos de gerenciamento (SNMP, OSI e proprietários) fazendo o uso de CORBA como ferramenta de integração.

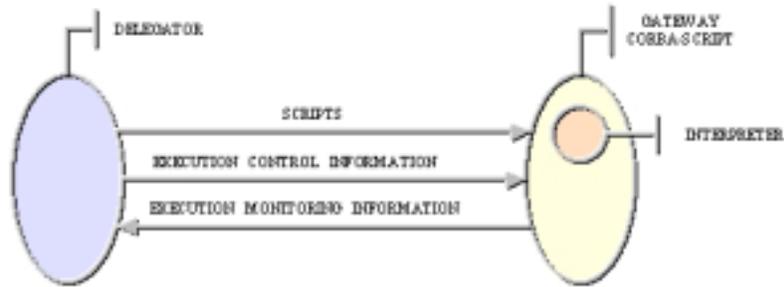


Figura 7: Delegação de scripts com interpretador acoplado ao agente delegador.

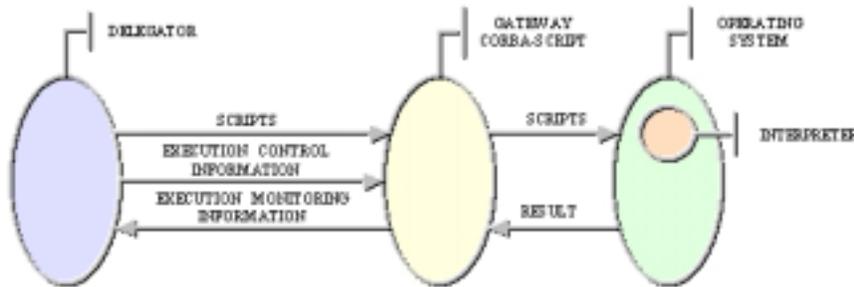


Figura 8: Delegação de scripts com interpretador desacoplado do agente delegador.

Com o objetivo de permitir a execução de scripts de qualquer linguagem, a arquitetura utilizará a delegação de scripts com ambiente de execução desacoplado do gateway CORBA-SCRIPT (figura 8). Assim, os principais componentes da arquitetura são (figura 10):

- Delegador (*Delegator*);
- Gerenciador de Entrada/Saída (*I/O Manager*);
- *Gateway* CORBA-SCRIPT;
- Sistema Operacional e Ambientes de Execução.

O delegador é o responsável pela distribuição, monitoramento e controle da execução dos scripts. A coleta e a organização das informações disponibilizadas pelos gateways CORBA-SCRIPT também é de responsabilidade do delegador. O modo de transferência de scripts poderá ser *push-model* quando for iniciado pelo delegador e *pull-model* quando for iniciado por um script em execução.

O gerenciador de Entrada/Saída define uma interface para criação, remoção, leitura, escrita e transferência de arquivos. Qualquer manipulação de arquivo solicitada pelo delegador remotamente ocorrerá através de um gerenciador de Entrada/Saída.

O gateway CORBA-SCRIPT oferece apenas uma interface comum para o controle de execução de scripts. Portanto, ele não adota nenhum Modelo de Gerenciamento específico (Internet, OSI ou proprietário) como os gateways CORBA-SNMP ou CORBA-OSI propostos pela OMG [10]. Assim, necessariamente os scripts serão orientados ao Modelo de Gerenciamento adotado pela sub-rede gerenciada.

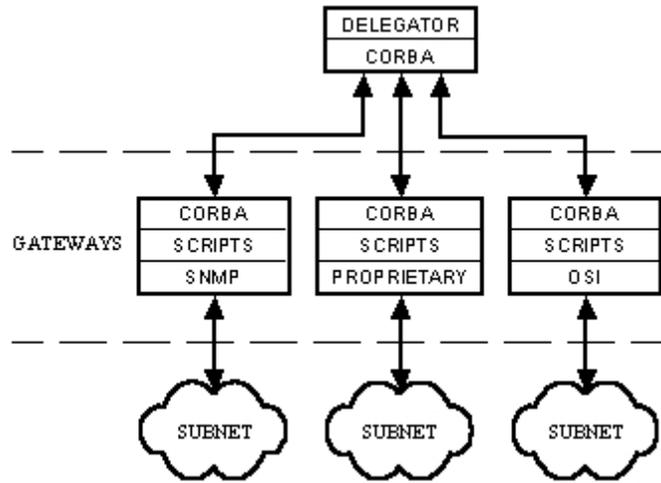


Figura 9: Arquitetura proposta.

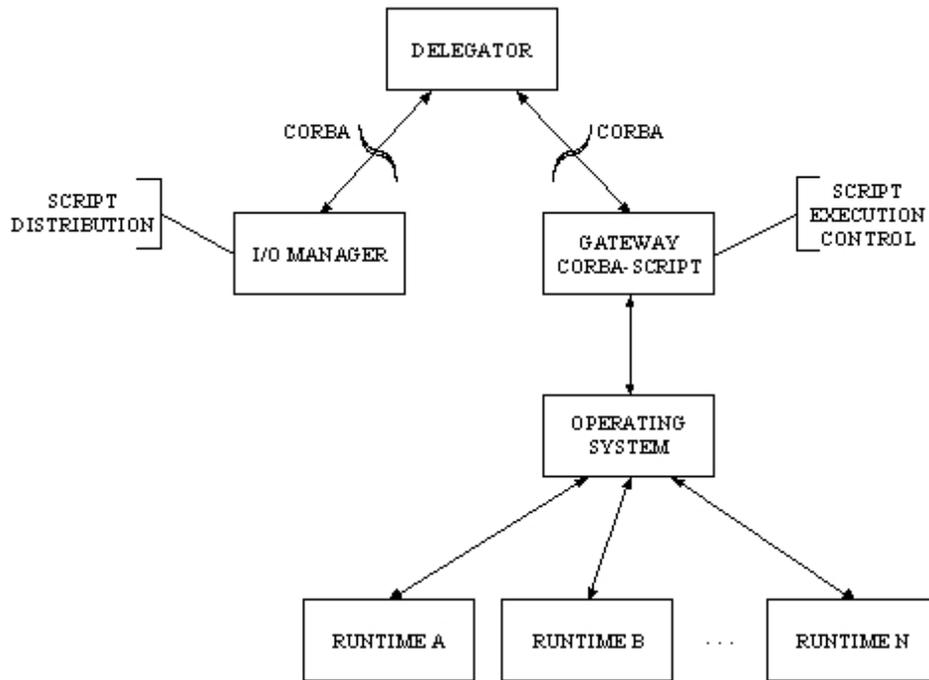


Figura 10: Arquitetura proposta.

O Ambiente de Execução por ser desacoplado do agente de delegação gateway (CORBA-SCRIPT) pode ser qualquer interpretador de script, máquina virtual ou até mesmo programas que executem diretamente sobre o Sistema Operacional.

A arquitetura não restringe a comunicação e troca de informações ao Delegador e o gateway. Na realidade os próprios scripts podem se comunicar entre si e com outros aplicativos através de protocolos como o ftp, http, snmp, tcp e udp. Para tanto, basta que linguagem script ofereça suporte a estes protocolos ou que faça uso de utilitários do próprio sistema

operacional.

3.3 Componentes da ferramenta

O delegador (figura 11) é o único componente que possui uma interface com usuário. A comunicação com o gateway CORBA-SCRIPT e com o gerenciador de Entrada/Saída é feita através do ORB via GIOP. O núcleo do delegador (*Delegator Kernel*) e a interface com o usuário são executados em uma Máquina Virtual Java (JVM - *Java Virtual Machine*) garantido a portabilidade da ferramenta. Quanto ao ORB, este pode ser executado em uma JVM ou diretamente sobre o Sistema Operacional pois a arquitetura CORBA garante a portabilidade dos objetos distribuídos. A geração de relatórios e de scripts na realidade é feita através de requisições feitas ao Sistema Operacional pela JVM. As informações dos relatórios são coletadas pelo Delegador.

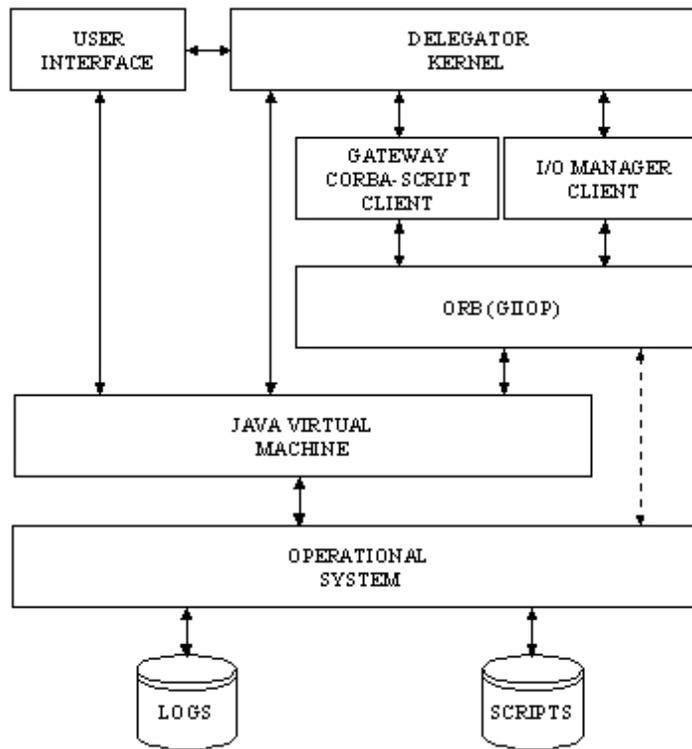


Figura 11: Arquitetura do Delegador.

Todas as funcionalidades do gateway CORBA-SCRIPT (figura 12) são disponibilizadas através de um ORB. O gateway CORBA-SCRIPT é executado em uma JVM e a geração de relatórios é feita através de requisições feitas ao Sistema Operacional pela JVM. As informações dos relatórios são coletadas pelo gateway CORBA-SCRIPT.

O gateway CORBA-SCRIPT oferece apenas uma interface comum para o controle de execução de scripts. Portanto, ele não adota nenhum Modelo de Gerenciamento específico (Internet, OSI ou proprietário) como os gateways CORBA-SNMP ou CORBA-OSI [9] propostos pela OMG. O gateway CORBA-SCRIPT poderá ser controlado por um ou mais

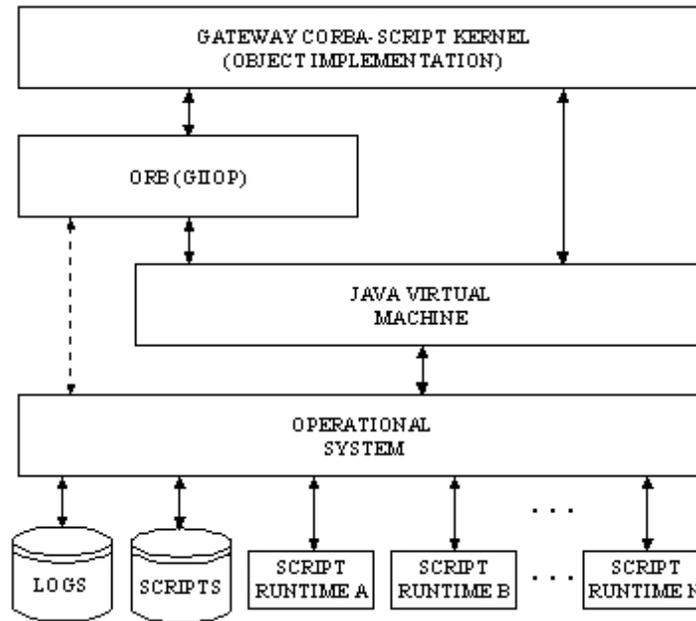


Figura 12: Arquitetura do *Gateway* CORBA-SCRIPT.

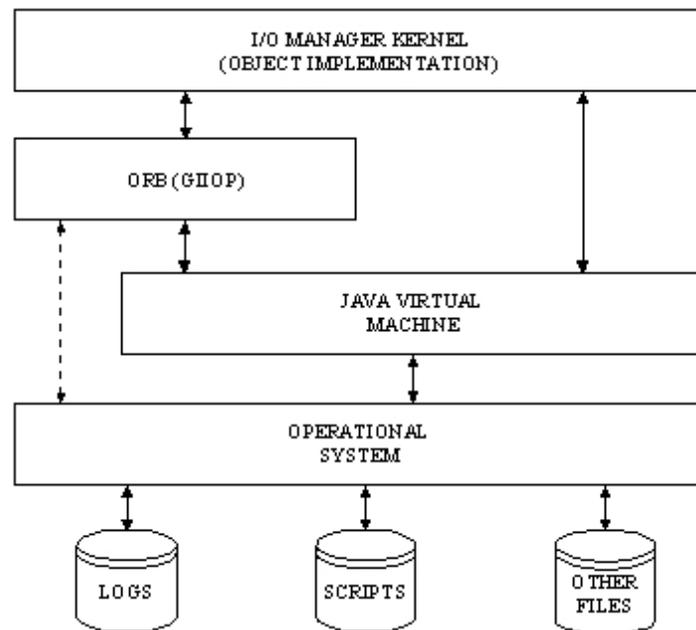


Figura 13: Arquitetura do Gerenciador de Entrada/Saída.

Delegadores mediante a autenticação destes usuários. Todas as operações executadas pelo gateway CORBA-SCRIPT serão registradas em relatórios.

O gerenciador de Entrada/Saída (figura 13) utiliza o ORB para disponibilizar suas funcionalidades. O ambiente de execução do gerenciador de Entrada/Saída é uma JVM e a geração de relatórios também é feita através de requisições feitas ao sistema operacional pela JVM.

Este é o componente responsável pela transferência de arquivos entre a entidade delegadora e o gateway. Outras funcionalidades como a listagem de arquivos/diretórios, a exclusão de arquivos/diretórios e a criação de diretórios também são suportadas pelo Gerenciador de Entrada/Saída.

3.4 Testes do módulo de geração de scripts

A linguagem utilizada para os scripts foi Tcl utilizando a extensão de gerenciamento de redes scotty [16]. O ambiente de execução dos scripts foi o interpretador Scotty2.1.10. Os gateways CORBA-SCRIPT e os gerenciadores de Entrada/Saída foram executados em um PC Celeron 466 MHz com 64 MB de memória utilizando o sistema operacional Linux Red Hat 6.1. O delegador foi executado, em um PC K6-II 450 MHz com 160 MB de memória e sistema operacional Windows 98. As aplicações delegador, gateway CORBA-SCRIPT e gerenciador de Entrada/Saída foram executadas em uma máquina virtual Java baseada no JDK1.3.

O teste descrito a seguir faz uma consulta a MIB para obtenção de informações de tráfego. Neste teste são consultadas as entradas da MIB ifInOctets contendo o valor do contador de tráfego de entrada e ifOutOctets contendo o valor do contador de tráfego de saída. São feitas apenas duas amostras intercaladas por cinco minutos e sendo obtidos os valores médios em seguida. Os valores são armazenados em arquivo. As entradas da MIB a serem consultadas são exibidas na tabela 1. A figura 14 apresenta uma possível implementação em Tcl. A figura 14 apresenta uma possível implementação em Tcl.

Identificador (OID)	Descritor	Tipo de Dado (ASN.1)
1.3.6.1.2.1.2.2.1.10	ifInOctets	Counter32
1.3.6.1.2.1.2.2.1.16	ifOutOctets	Counter32

Tabela 1: Entradas da MIB consultadas no teste 1.

4 Conclusões

Esse trabalho contribuiu com uma ferramenta para a construção e evolução de um ambiente de gerência baseado em CORBA, e especializada na construção de gateways e scripts de gerenciamento para integrar a esse ambiente os mais diversos elementos de rede, inclusive com o acréscimo de funcionalidades.

A utilização da ferramenta tem facilitado a geração de gateways CORBA, principalmente a partir do momento que foram incluídas mais classes auxiliares desenvolvidas para fins específicos e, pelo menos, tem demonstrado a eliminação de trabalhos repetitivos e que exigem menor especialização do desenvolvedor para a geração de gateways CORBA.

A potencialidade do módulo script vai além da delegação e controle de execução de tarefas de gerenciamento. A utilização de linguagens script permite o aumento da velocidade de desenvolvimento de aplicações de gerenciamento distribuídas. Os scripts podem se comunicar

<pre> GetTraffic; begin SnmpSession s; Counter32 InValue, OutValue, InOctets[2], OutOctets[2]; // creation of snmp session // with router 150.164.200.1 s = newSnmpSession(150.164.200.1); // get counters InOctets[0] = s.get(ifInOctets.1); OutOctets[0] = s.get(ifOutOctets.1); // wait 5 minutes wait(5min); // get counters again InOctets[1] = s.get(ifInOctets.1); OutOctets[1] = s.get(ifOutOctets.1); // compute traffic data InValue = InOctets[1] - InOctets[0]; InValue = InValue/300; OutValue = OutOctets[1] - OutOctets[0]; OutValue = OutValue/300; // save to file appendToFile(traffic.dat, "InValue:OutValue"); end; </pre>	<pre> # # GetTraffic # Possible Tcl Implementation # # creation of snmp session # with router 150.164.200.1 set s [snmp session -address 150.164.200.1] # get counters set InOctets0 [\$s get ifInOctets.1] set InOctets0 [split [lindex InOctets0 0]] set InOctets0 [lindex InOctets0 2] set OutOctets0 [\$s get ifOutOctets.1] set OutOctets0 [split [lindex OutOctets0 0]] set OutOctets0 [lindex OutOctets0 2] # wait 5 minutes after 300000 # get counters again set InOctets1 [\$s get ifInOctets.1] set InOctets1 [split [lindex InOctets1 0]] set InOctets1 [lindex InOctets1 2] set OutOctets1 [\$s get ifOutOctets.1] set OutOctets1 [split [lindex OutOctets1 0]] set OutOctets1 [lindex OutOctets1 2] # compute traffic data set inTraffic [expr \${InOctets1} - \${InOctets0}] set inTraffic [expr \${inTraffic} / 300] set outTraffic [expr \${OutOctets1} - \${OutOctets0}] set outTraffic [expr \${outTraffic} / 300] # save to file set fileID [open traffic.dat a+] puts \$fileID "\${InValue};\${OutValue}" close \$fileID exit </pre>
---	--

Figura 14: Pseudo-código do teste 1 e uma possível implementação Tcl.

e trocar informações com outros scripts e com outros aplicativos através de protocolos como o TCP, UDP, HTTP e SNMP.

Este módulo, com a utilização de linguagens scripts, o padrão CORBA e a linguagem Java, permite um alto grau de portabilidade. As linguagens scripts possuem ambientes de execução para diferentes plataformas de hardware e de software como x86/Windows/Linux, Solaris/Unix UNIX e Vax/VMS. Além disso, as linguagens scripts representam uma forma rápida de integrar diferentes componentes de software como, por exemplo, os utilitários dos sistemas operacionais. Java, através do uso de máquinas virtuais, é capaz de gerar aplicações multiplataformas e, portanto, a ferramenta MSDT herda esta característica. O padrão

CORBA foi definido com o propósito de integrar ambientes heterogêneos de forma que outras aplicações podem ser definidas para utilizar os serviços do gerenciador de Entrada/Saída e do gateway CORBA-SCRIPT.

Referências

- [1] Borland. *VisiBroker for Java - Implementação de um ORB Java*. www.borland.com, 2001.
- [2] J.D. Case, M. Fedor, M. Schoffstall, and J. Davin. *A Simple Network Management Protocol (SNMP)*. Technical Reports, SNMP Research Inc., 1990.
- [3] P. Haggerty and K. Seetharaman. *The Benefits of CORBA-based Network Management*, 1998.
- [4] DOG Distributed Object Groups. *JavaORBv2.2.7 - Implementação de um ORB Java*. www.multimania.com/dogweb/, 2001.
- [5] Reaz Hoque. *CORBA 3*. IDG Books World Wide, 1998.
- [6] ISO/IEC ITU-T. *ITU-T Recommendation X.711 | ISO/IEC 9596-1*. Information Technology - Open Systems Interconnection – CMIP, Part 1: Specification, 1991.
- [7] JIDM. *JIDM - Joint Inter Domain Management*. www.jidm.org, 2001.
- [8] X/Open / NMF. *SNMP SMI to CORBA IDL, ASN.1/GDMO to CORBA IDL and IDL to GDMO/ASN.1 translations*. Joint Inter-Domain Management (JIDM) Specifications, 1994.
- [9] OMG. Corba/TMN Interworking, Version 1.0, August 2000.
- [10] OMG. <http://www.omg.org>, Acesso em Agosto de 2000.
- [11] OMG. *Event Service Specification, Version 1.0*. www.omg.org/technology/documents/formal/event_service.htm, 2001.
- [12] OMG. *Object Management Group - OMG*. <http://www.omg.org>, 2001.
- [13] OMG. *Persistent State Service Specification*. www.omg.org/technology/documents/formal/persistent_object_service.htm, 2001.
- [14] Juan Pavlon. *Building Telecommunications Management Applications with CORBA*. IEEE Communication Survey, 1999.
- [15] H. Schildt and P. Naughton. *Java 2-The Complete Reference*. Osborne, 4th ed, 2000.
- [16] J. Schönwälder and H. Langendorfer. Tcl Extensions for Network Management Applications. In *Proceedings of the 3rd Tcl/Tk Workshop*, Toronto, Canada, July 1995.
- [17] Jon Siegel. *CORBA 3 Fundamentals and Programming*. OMG Press, 2o edition, 2000.
- [18] Sun. *JDK 1.3 - Java 2 Plataform*. java.sun.com/j2se/1.3/, 2001.
- [19] Steve Vinoski. *CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments*, volume 35, No. 2. IEEE Communications Magazine, 1997.
- [20] Steve Vinoski. *New Features for CORBA 3.0*. ACM Communications Magazine, 1998.
- [21] Andreas Vogel and Kaith Duddy. *Java Programming with CORBA*. John Wiley, 2000.
- [22] Ron Zahavi and David Linthiam. *Enterprise Application Integration with CORBA Component and Web-Based Solutions*. OMG Press, 1999.