



Exacta

ISSN: 1678-5428

exacta@uninove.br

Universidade Nove de Julho  
Brasil

Nguessan, Désiré; Westphall, Carlos Becker  
Computação de objetos distribuídos na era da internet  
Exacta, núm. 1, abril, 2003, pp. 77-88  
Universidade Nove de Julho  
São Paulo, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=81000107>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica  
Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal  
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

# COMPUTAÇÃO DE OBJETOS DISTRIBUÍDOS NA ERA DA INTERNET

DÉSIRÉ NGUESSAN

Mestre em Ciências da Computação – Universidade Federal de Santa Catarina e  
Professor do Curso de Ciências da Computação na UNINOVE

CARLOS BECKER WESTPHALL

Doutor em Ciências da Computação – Université Paul Sabatier – França; Professor  
e pesquisador do Curso de Pós-Graduação em Ciências da Computação e  
Laboratório de Redes e Gerência na Universidade Federal de Santa Catarina

## Resumo

Neste trabalho, discutem-se as recentes tendências em objetos distribuídos e as tecnologias da Internet. As duas convergem na criação de um paradigma para computação distribuída. Apresenta-se uma visão do CORBA (*Common Object Request Broker Architecture*), salientando sua arquitetura aberta e seu protocolo IIOP (*Internet Inter-ORB Protocol*), que proporcionam uma melhor integração de aplicações distribuídas em ambientes heterogêneos. O protocolo do CORBA está emergindo como padrão para a comunicação entre aplicações na Internet e merece atenção das organizações de Tecnologias de Informação (TI). Conclui-se que CORBA, em conjunto com a Internet, constitui uma simbiose perfeita para desenvolvimento e manutenção de aplicações cliente/servidor com missão crítica.

**Palavras-chave:** *objetos distribuídos; Corba; tecnologias da Internet.*

## Abstract

In this paper, we discuss recent trends in distributed objects and the Internet computing technologies. Both technologies converge to create a paradigm for distributed computing. We provide an overview of CORBA (*Common Object Request Broker Architecture*), emphasizing its open architecture for distributed application based on distributed objects and the IIOP (*Internet Inter-ORB Protocol*) that improve the integration of the applications in heterogeneous environments. The CORBA protocol is emerging as the business application messaging standard for the Internet and deserves attention from Information Technologies (IT) organizations. We conclude that CORBA, together with the Internet, constitute a perfect symbiotic relationship to build, maintain and extend client/server applications with critical mission.

**Key words:** *distributed objects; Corba; Internet computing technologies.*

## 1. Introdução

Nos quinze últimos anos, ocorreram muitas mudanças no desenvolvimento e manutenção dos sistemas de informação das empresas. Este período começou com os sistemas *mainframe* monolíticos, em que cada um tinha seu próprio gerenciamento de apresentação, sua própria lógica de aplicação e de acesso aos



dados. Cada interface proprietária de acesso a dados era dependente de um único sistema de dados, não sendo possível a um sistema compartilhá-los com outros em razão dos problemas de sincronização e integridade desses dados; portanto, cada sistema tinha de armazenar cópias privadas de seus próprios dados para *back-up*. Estes sistemas monolíticos eram ineficientes e apresentavam um alto custo; por isso foram substituídos pelos modelos cliente/servidor. Esta mudança foi possível pela convergência das tecnologias de redes, do baixo custo dos PCs, das GUIs (*Graphic User Interface*) e dos bancos de dados relacionais. A computação cliente/servidor simplificou relativamente o desenvolvimento e a manutenção das aplicações complexas, dividindo os sistemas monolíticos centralizados em componentes, clientes e servidores. (*Visibroker Distributed Object Computer*, 1998)

No modelo cliente/servidor, o cliente solicita um determinado serviço. Um ou vários processos chamados de servidores são responsáveis pelo fornecimento de serviços ou resposta às solicitações do cliente. Os serviços são acessados via interface. Ao receber uma solicitação válida, o servidor executa a operação apropriada e envia a resposta ao cliente; tipo de interação conhecida como *Request/Reply*. O modelo cliente/servidor aumentou a disponibilidade e otimizou o custo, mas não resolveu problemas, tais como: reusabilidade de códigos, extensibilidade, interoperabilidade, autonomia e mobilidade. As tecnologias de objetos distribuídos e da Internet convergem no mesmo caminho, para proporcionar melhores soluções a esses problemas. (*Visibroker Distributed Object Computer*, 1998)

Este trabalho está estruturado da seguinte forma: nas seções 2 e 3, apresentam-se as recentes tendências em objetos distribuídos, enfatizando a arquitetura CORBA; na seção 4, salientam-se as tecnologias da Internet, e na 5, mostra-se que CORBA, em conjunto com a Internet, constitui uma simbiose perfeita para desenvolvimento de aplicações distribuídas heterogêneas.

## 2. Revolução dos objetos distribuídos

A tecnologia de objetos distribuídos transformou as aplicações cliente/servidor monolíticas em componentes autogerenciáveis, ou objetos que podem interoperar em redes e sistemas operacionais diferentes. Os objetos são implementados segundo o conceito da programação orientada a objetos, sendo executados em ambientes que suportam serviços como localização transparente de objetos, invocação de métodos em objetos locais ou remotos e a migração de objetos, usando um ORB. (*Object Request Broker*) (OMG, 1998)

Essa tecnologia permite às organizações de Tecnologia de Informação construir infra-estruturas que se adaptam às mudanças e às oportunidades do

mercado. Desde 1997, a tecnologia de objeto vem-se tornando o paradigma mais usado para conceituar e desenvolver as aplicações. (*Visibroker Distributed Object Computer*, 1998) A tecnologia de objetos distribuídos apresenta várias vantagens, como, por exemplo, a reusabilidade e a extensibilidade. Associada ao paradigma cliente/servidor, ela permite criar aplicações distribuídas que proporcionam uma melhor forma de integração no processamento das informações distribuídas, ou aplicações que suportam interoperabilidade. Entretanto, as aplicações distribuídas trouxeram novas requisições: operar em ambientes de computação heterogênea e integrar as tecnologias antigas com as novas requer um alto grau de disponibilidade e performance, facilidade de administração e, sobretudo, integridade dos dados, ou seja, sua segurança.

Para atender a essas novas requisições, existem atualmente vários ambientes e ferramentas para desenvolvimento de aplicações distribuídas orientadas a objetos. Entre essas tecnologias que suportam o desenvolvimento de aplicações baseadas em objetos, as mais conhecidas no mercado são: CORBA, que é a proposta do OMG; o DCOM, a solução da *Microsoft* e o Java-RMI, do *JavaSoft*. Um ponto comum entre as diferentes abordagens para sistemas distribuídos orientados a objetos, apresentadas por essas tecnologias, é a existência de um elemento responsável por disponibilizar transparentemente os objetos das aplicações servidoras para aplicações clientes. (ORFALI, 1997) Neste trabalho, enfatiza-se CORBA, pois no ‘estado da arte’ ele se destaca melhor, apresentando resultado mais satisfatório. (NGUESSAN, 2000)

### 3. CORBA

O OMG é um consórcio internacional, criado em 1989, que agrupa atualmente mais de 850 membros do mundo da informática: os construtores, como IBM e Sun; os produtores de programas *Netscape*, *Inprise* ou *Ex-Borland/Visigenic*, *IONA Tech*; os usuários, como Boeing e Alcatel, por exemplo, e as instituições e universidades (NASA, INRIA, LIFT). O objetivo deste grupo é definir padrão para de aplicações distribuídas heterogêneas, utilizando tecnologias orientadas a objeto. Portanto, os conceitos-chave enfatizados são reusabilidade, interoperabilidade e portabilidade dos componentes das aplicações distribuídas. As especificações do OMG são reconhecidas pelo ISO (*International Standards Organization*). O elemento principal da visão do OMG é o CORBA. (GEIG, 1997) No fim de 1994, o OMG lançou o CORBA versão 2.0, que inclui o protocolo IIOP. Neste trabalho, foi usada a versão 2.3.



1 Camada de *software* residente acima do sistema operacional que oferece abstrações de alto nível, com objetivo de facilitar a programação distribuída.

O que faz do CORBA uma ferramenta importante é o potencial de assumir o lugar de um *middleware*<sup>1</sup> cliente/servidor, usando objetos como mecanismo de unificação de aplicações existentes. A utilização do CORBA faz com que o sistema inteiro seja autodescritivo, pois a especificação dos serviços é feita de forma separada da implementação, o que permite a incorporação de sistemas existentes, independência de plataforma e de linguagem de programação. Um objeto distribuído CORBA pode viver em qualquer lugar da rede, sendo acessado por clientes remotos por meio de métodos de invocação. Na Figura 1, um cliente implementado na forma de *applet* pode invocar um objeto no servidor remoto de uma rede de empresa, via Internet. Tanto a linguagem quanto o compilador utilizado para a geração do código do objeto servidor são totalmente transparentes para o cliente. Ele não precisa saber onde o objeto distribuído está localizado, qual sistema operacional está sendo utilizado para executá-lo ou em que linguagem foi escrito; necessita apenas saber a interface de acesso fornecida pelo objeto servidor. (OMA, 1998)

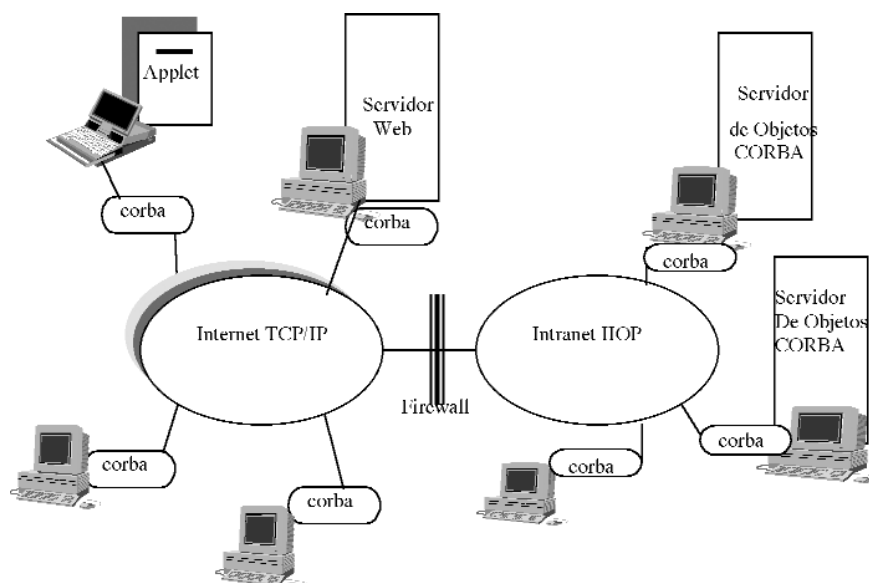


Figura 1 - Arquitetura de sistema distribuído usando CORBA e Internet. (Nguessan, 2000)

### 3.1 Arquitetura CORBA

Em 1990, o OMG criou o OMA (*Object Management Architecture*) com o objetivo de fornecer infra-estrutura conceitual para todas as especificações OMG. O OMA é a arquitetura geral dos componentes necessários ao desenvolvimento de um ambiente portátil e interoperável; isso inclui a necessidade de interfaces

orientadas a objetos, transparência de distribuição, uma forma comum de modelar objetos, uma base comum para todos os componentes, suporte total para todos os estágios do ciclo de vida do *software*, uma natureza flexível e dinâmica, alta performance, implementações robustas e compatibilidade com padrões existentes dentro da indústria de *software*. (OMA, 1998)

O OMA é composto de quatro elementos principais: ORB (*Object Request Broker*), Objetos de Aplicação (*Application Objects*), Serviços de Objetos Comuns (*Corba Services*) e Facilidades Comuns (*Corba Facilities*). A Figura 2 dá uma idéia geral da estrutura e dos elementos que compõem o OMA.

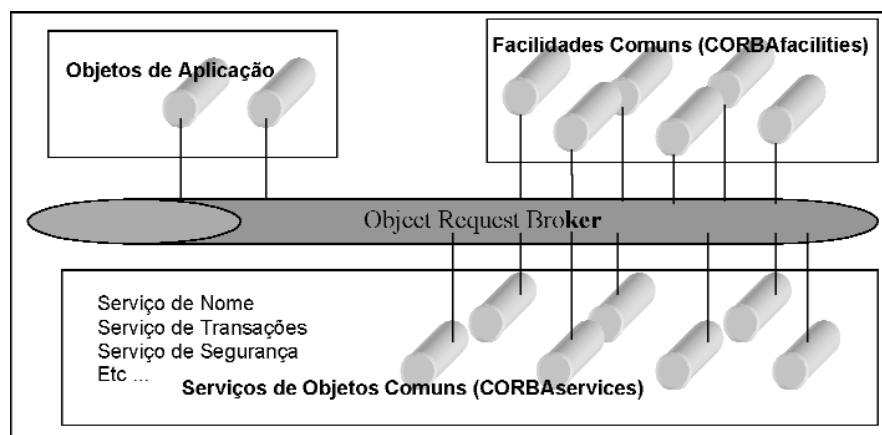


Figura 2 - Arquitetura de Gerenciamento de Objetos. (OMA, 1998)

Percebe-se que os Objetos de Aplicação estão relacionados às aplicações específicas do usuário, que serão integradas ao ambiente e não padronizadas pelo OMG. As Facilidades Comuns propiciam uma coleção de serviços que muitas aplicações podem compartilhar, mas que não são tão fundamentais como os Serviços de Objetos. Estes fornecem funções básicas para usar e implementar objetos, que se comunicam com outros, via ORB – elemento-chave de comunicação, que fornece mecanismos pelos quais objetos são ativados, enviam requisições e recebem respostas de forma transparente. (OMA, 1998)

### 3.1.2 *Object Request Broker*

Objetos Clientes requisitam serviços às implementações de objetos por meio de um ORB (Figura 3), que é responsável por todos os mecanismos requisitados para encontrar o objeto servidor e preparar a implementação de objeto para receber a requisição e executá-la. A Figura 3 ilustra a requisição de uma implementação de objeto.

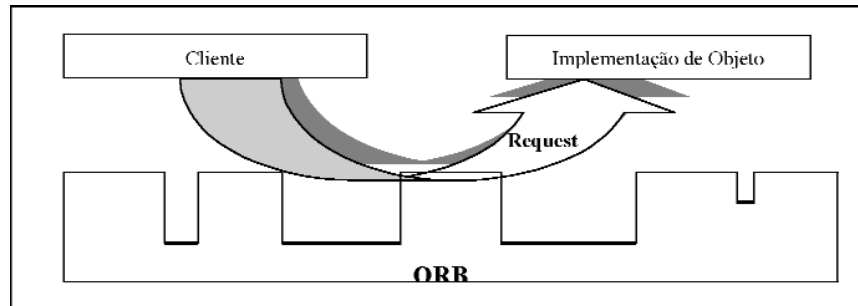


Figura 3 - Requisição de uma implementação de objeto.

O cliente não tem noção da localização do objeto servidor nem sabe que linguagem foi utilizada para sua implementação ou qualquer outro detalhe referente à implementação deste objeto, que é definido usando a OMG IDL (*Interface Definition Language*) como forma de descrever interfaces, isto é, de especificar um contrato entre os objetos. A OMG IDL é uma linguagem puramente declarativa. A Figura 4 ilustra a estrutura das interfaces de requisição de objetos.

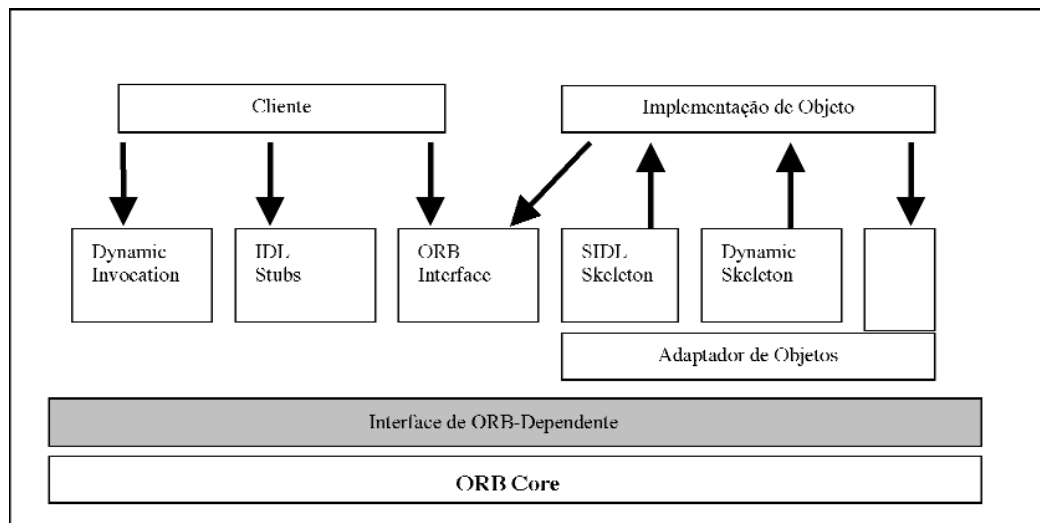


Figura 4 - Estrutura das Interfaces de Requisição de Objeto (OMG, 1998)

### 3.1.3 Arquitetura do ORB

A arquitetura do ORB é composta de três componentes específicos: interface de cliente, interface de servidor e ORB core ou núcleo do ORB. A interface de cliente é responsável pelo mapeamento entre a linguagem cliente e a

implementação ORB. O cliente interage com o servidor remoto invocando suas operações, exatamente como se invocasse operações em objetos locais. A interface servidor, pelo *Object Adapter*, permite associar uma interface de objeto com sua implementação; gerar e interpretar as referências de objetos e mapeá-las com as implementações desses objetos; ativar e desativar objetos e respectivas implementações; registrar implementações e invocar implicitamente métodos por meio do *skeleton*. O ORB *Core* é responsável pela manipulação da comunicação básica das requisições dos vários componentes, podendo ser visto como uma camada básica de transporte. (Figura 4)

### 3.2 Interoperabilidade entre ORBs

A especificação de interfaces de objetos obrigatoriamente em OMG IDL garante-lhes a portabilidade, utilizando diferentes linguagens, ferramentas, sistemas operacionais e redes. Entretanto, a característica de interoperabilidade entre objetos só foi possível no CORBA versão 2. Isso se deu pela especificação de uma arquitetura de interoperabilidade, um suporte para pontes entre ORBs, um protocolo para comunicar entre ORBs genéricos e um para comunicação entre ORBs para Internet. A arquitetura de interoperabilidade do CORBA identifica claramente a regra de diferentes tipos de domínios para informações específicas de ORBs. Tais domínios podem incluir domínios de referências de objeto, domínios de tipos, domínio de segurança, entre outros. Quando dois ORBs estão no mesmo domínio, podem comunicar-se diretamente. Entretanto, quando a informação em uma invocação deixa seu domínio, deve atravessar uma ponte.

A regra de ponte deve assegurar que o conteúdo e a semântica sejam mapeados adequadamente de um ORB para outro. O suporte para ponte entre ORBs pode também ser usado para prover interoperabilidade com outros sistemas não CORBA. Uma ponte que provê uma conversão direta é chamada de *immediate bridging*; a que requer um formato de interoperabilidade para ser usado externamente na comunicação entre dois ambientes de ORB denomina-se *mediated bridging*. Para a comunicação entre ORBs, o OMG definiu os protocolos GIOP e IIOP.

#### 3.2.1 GIOP e IIOP

O *GIOP* (*Generic Inter ORB Protocol*) especifica uma sintaxe de transferência padrão e um conjunto de formatos de mensagens para comunicação entre ORBs. O *IIOP* (*Internet Inter ORB Protocol*) especifica como mensagens GIOP são trocadas usando conexões TCP/IP.





O relacionamento entre GIOP e IIOP é similar ao mapeamento existente entre o OMG IDL e uma linguagem específica: o GIOP pode ser mapeado em diferentes protocolos de transporte e determina os elementos de protocolo comuns a todos os mapeamentos. No entanto, não fornece uma completa interoperabilidade, da mesma forma que IDL não pode ser usado para construir programas completos. IIOP e outros mapeamentos similares para diferentes protocolos de transporte são realizações concretas das definições abstratas de GIOP. A Figura 5 mostra o relacionamento de protocolos entre ORBs.

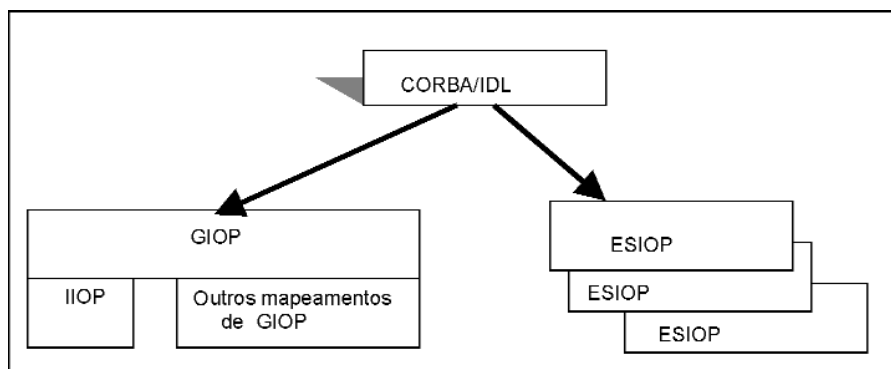


Figura 5 - Relacionamento de protocolos entre ORBs.

O ESIOP (*Environment Specific-Inter ORB Protocol*), protocolo entre ORBs para ambientes específicos, deve ser usado para interoperar em locais onde uma rede ou infra-estrutura de computação distribuída já esteja em uso. Apesar de cada ESIOP poder ser otimizado para uma arquitetura em particular, toda especificação ESIOP deve estar conforme as convenções da arquitetura de interoperabilidade para facilitar o uso de pontes, se necessário. O suporte de ponte entre ORBs habilita a construção de pontes entre domínios de ORBs que usam IIOP e os que utilizam um ESIOP particular.

### 3.3 Serviços CORBA

O ORB, por si só, não executa todas as tarefas necessárias para que os objetos interoperem; ele só fornece os mecanismos básicos. Outros serviços necessários são oferecidos por objetos com interface IDL, que o OMG vem padronizando. Esses serviços são utilizados para aumentar e complementar a funcionalidade do ORB. Até o momento da elaboração deste trabalho haviam sido publicados quinze serviços, entre os quais *Naming Service*, *Event Service*, *Security Service*.

### 3.4 Conclusão

CORBA é uma arquitetura aberta que associa o paradigma cliente/servidor com a tecnologia de orientação a objetos. Ele permite a integração de aplicações distribuídas heterogêneas, fazendo com que estas façam solicitações a objetos de forma transparente e independente, indiferente às plataformas de *hardware*, sistemas operacionais, linguagens e considerações de localização.

## 4. Tecnologias-chave da Internet

Enquanto CORBA2.0 estava sendo especificado, a Internet e a *World Wide Web* constituíam um fenômeno em rápida expansão, que até hoje continua crescendo. A Internet ultrapassou as raízes de órgãos do governo e instituições de educação que lhe foram impostas no início da sua criação, para ser hoje o meio mais significativo de comunicação entre empresas, organizações governamentais educacionais e pessoas físicas. Por meio da Internet pode-se acessar, fornecer e trocar um conjunto quase ilimitado de informações de uso empresarial, acadêmico e pessoal. O crescimento da Internet e das Intranets continuará a grande velocidade até pelo menos por mais uma década. A Internet é uma interconexão global sem precedente na história da computação (*Visibroker Distributed Object Computer*, 1998). Desenvolvimentos relacionados à Internet – Java, *Applet Java*, *Web Browser* e *Firewall* – desempenham funções-chave que permitem às organizações de Tecnologia da Informação criar e utilizar melhor os objetos distribuídos.

### 4.1 Java

Trata-se de uma poderosa linguagem de programação orientada a objetos, desenvolvida pela *Sun Microsystems*. Embora seja conhecida como uma linguagem para o *World Wide Web*, Java pode ser utilizado para desenvolver aplicações simples e complexas. Programas Java são compilados e geram o *byte-code*, que é um código executado na Máquina Virtual Java, permitindo que todo programa Java seja executado independentemente do processador e do sistema operacional. Isso significa um avanço sobre as outras linguagens de programação. Em aplicações distribuídas é muito importante, na medida em que uma aplicação cliente ou um objeto servidor podem ser executados em qualquer plataforma. Conseqüentemente, o custo de desenvolvimento e de manutenção pode ser reduzido de maneira significativa.



#### 4.2 Applets Java

São pequenos programas escritos em Java, móveis do servidor Web até o cliente em que eles são executados no contexto de um *browser Web*. Uma das vantagens dos *applets* é que eles permitem evitar o problema de incompatibilidade de versão que existe com os *softwares* aplicativos, muitas vezes carregados e atualizados em cada máquina cliente.

#### 4.3 Web Browsers

São navegadores de páginas de hipermídia – *Netscape* e *Internet Explorer*, entre outros. Esses navegadores são fornecidos gratuitamente e vêm em milhões de computadores. Permitem visualizar as informações apresentadas na *Web* e incluem suporte para os serviços da *Internet*; além disso, possibilitam executar os *Applets Java* que podem comunicar-se e interagir com aplicações executadas em sistemas remotos via rede *Internet*. Dessa forma, as empresas de Tecnologia de Informação estão usando a *Internet* de fato como um *WAN* que interliga agências do governo, empresas de negócios, instituições de educação e indivíduos. A Figura 1 mostra como se podem combinar *CORBA* e *Internet*.

#### 4.4 Segurança dos Firewalls

Os *firewalls* fornecem uma segurança para os sistemas de informação das empresas, permitindo que seus aplicativos internos interajam de maneira segura com os aplicativos em sistemas fora do *firewall*.

#### 4.5 Conclusão

Na era da *Internet* não se pode mais pensar na criação de ambientes de computação homogêneos. Os sistemas de informação das empresas devem ser capazes de comunicar e interoperar com aplicações e sistemas fora do *firewall* da empresa e em ambientes heterogêneos. A *Internet* é uma interconexão de redes mundial em rápida expansão que oferece inúmeros serviços, com mais de dois milhões de organizações acadêmicas, militares, científicas e comerciais.

Nosso objetivo neste trabalho é focalizar os aspectos que, com a arquitetura *CORBA*, permitem desenvolver aplicações heterogêneas, proporcionando uma melhor forma de integrar o processamento das informações distribuídas, ou seja, aplicações que suportam interoperabilidade. Para isso, as tecnologias-chave focalizadas foram *Java*, *Applet Java*, *Web Browser* e *Firewall*.

## 5. Considerações sobre o CORBA e a Internet

CORBA é uma arquitetura aberta que associa o paradigma cliente/servidor com a tecnologia de orientação a objetos. Ele permite a integração de aplicações distribuídas heterogêneas, fazendo com que essas aplicações façam solicitações a objetos, de uma forma transparente e independente, indiferente às plataformas de *hardware*, sistemas operacionais, linguagens e considerações de localização.

O modelo CORBA oferece uma série de benefícios que facilitam a integração de aplicações em um ambiente distribuído. Este modelo permite que os projetistas de sistemas distribuídos tirem proveito de técnicas orientadas a objeto como encapsulamento e herança. Estas técnicas fazem com que os objetos sejam definidos como ‘caixas pretas’ capazes de executar determinadas tarefas, sem que o sistema necessite conhecer o funcionamento interno deles. Outra vantagem é a definição mais clara das interfaces entre as partes do sistema, as quais podem ser alteradas sem afetá-lo como um todo. Assim, consegue-se uma melhor modularidade. Os sistemas tornam-se mais extensíveis, pois novas características podem ser adicionadas com a definição de um novo conjunto de operações. CORBA pode coexistir e interoperar perfeitamente com outros sistemas que não foram desenvolvidos usando essa tecnologia, permitindo assim preservar o investimento do sistema existente. Por exemplo, um objeto CORBA pode comunicar-se com um objeto desenvolvido com tecnologia *ActiveX/DCOM* da Microsoft, por sua arquitetura aberta, flexibilidade, interoperabilidade e portabilidade, e pode ser visto como a solução dos problemas encontrados em sistemas distribuídos.

As atuais revoluções na Internet e na computação de objetos distribuídos convergem no mesmo caminho: a Internet, vista como um *framework* de comunicações, fornece uma plataforma ideal para aplicação de objetos distribuídos. Ao mesmo tempo, a tecnologia de objetos distribuídos está melhorando a qualidade das aplicações baseadas em *Web*, dando mais valor à Internet e Intranet nas empresas de Tecnologia de Informação. Esse relacionamento simbiótico entre as duas tecnologias gera um novo modo de conceituar, projetar, desenvolver, executar e manter as aplicações de negócios. Isso faz surgir um paradigma de aplicação, construído a partir de códigos-fonte encapsulados na forma de objeto executado em sistema protegido, ou não, por *firewall* e interagindo um com o outro, por meio de um ORB. Este novo paradigma de aplicação tem muitos benefícios a oferecer às empresas de Tecnologia de Informação, tais como: flexibilidade para integrar aplicações de diferentes fabricantes, administração e gerenciamento centralizados dos objetos



aplicativos, drástica redução no custo da configuração e manutenção dos sistemas clientes, entre outros.

As empresas que empregam ou se adaptam a estes novos conceitos, sem dúvida, terão uma grande vantagem sobre seus competidores.

#### Referências bibliográficas

GEIG, Jean-Marc; GRANSART, Christophe; Phillipe, M. *CORBA: des concepts à la pratique*. Collection Inter Éditions, Paris: Maison, 1997.

HAGGERTY, P.; SEETHARAMAN, K. The Benefits of CORBA-Based Network Management. *IEEE Communications of the ACM*. v. 41, 1998, p. 73-79.

NGUESSAN, D. Dissertação de Mestrado. *Um Modelo de Gerência de Segurança Baseado em Obejetos Distribuídos*. Curso de Pós-Graduação em Ciências da Computação da UFSC, Abril 2000.

OMA - Document Object Management Architecture. *Executive Overview*. Needham: OMA, 1998.

OMG - Document number. *CORBA/IIOP 2.3 Specification*. Disponível em <<http://www.omg.org/corba/corbiip.htm>>. Acessado em 24/03/2003.

OMG - Document number. *Security Service Specification*. In: CORBA services: Common Object Services Specification. Needham: OMG, 1998.

ORFALI, R.; HARKEY, D. *Client/Server programming with Java and CORBA*. New York: John & Wiley, 1997.

Visibroker Distributed Object Computer. Disponível em <[http://www.borland.com/products/downloads/download\\_besvisibroker.html](http://www.borland.com/products/downloads/download_besvisibroker.html)>. Acessado em 24/03/03.