

# Uma Implementação de um Gateway para as Redes Inteligentes Distribuídas

Renzo Quevedez Malini<sup>1</sup>, Rosane Bodart Soares<sup>2</sup>, Anilton Salles Garcia<sup>1</sup>

Universidade Federal do Espírito Santo (UFES) - <sup>1</sup>Depto. de Informática (DI), <sup>2</sup>Depto. de Elétrica (DEL) - Vitória, ES

*rmalini@inf.ufes.br, rosane@ele.ufes.br, anilton@inf.ufes.br*

**Resumo** — As telecomunicações vêm experimentando mudanças significativas, que apontam para a integração das redes de dados e voz. Neste artigo, tal tendência é considerada, especificamente para as Redes Inteligentes (RIs) de Telecomunicações. É proposto um primeiro passo em direção à migração das RIs tradicionais, centralizadas, para as RIs distribuídas, e apresentada a implementação de parte de um gateway de comunicação entre a RI tradicional e uma RI distribuída via CORBA, implementação esta baseada em uma especificação apresentada como resposta a uma RFP do OMG. São descritas as implementações realizadas e apresentadas algumas conclusões sobre os resultados obtidos nas mesmas.

**Palavras-Chave** — Redes Inteligentes, CORBA, Gateway RI/CORBA, Serviços CORBA.

**Abstract** — Telecommunications have experimented significant changes, which lead to an integration between data and voice nets. In this paper, such tendency is considered, specifically for Intelligent Networks (INs). A first step is proposed toward the migration from the traditional, centralized INs, to the distributed ones, and the implementation of part of a gateway that provide communication between the traditional IN and the one distributed via CORBA is presented, which was based in a specification presented to the OMG as a response of a RFP. The developed implementations are described and the main conclusions about them are presented.

**Keywords** — Intelligent Networks, CORBA, IN/CORBA Gateway, CORBA Services.

## I. INTRODUÇÃO

As operadoras de telecomunicações sempre foram requisitadas por seus clientes para a oferta de novos serviços e por mais qualidade naqueles já existentes. Entretanto, nos últimos anos, especialmente a partir de meados da década de 90, essa pressão tem sido exercida ainda com mais vigor, devido à popularização da Internet, à expansão da Telefonia Móvel e, especialmente no Brasil, à privatização da Telefonia Fixa, dentre outras.

A partir da análise do histórico mais recente sobre a evolução das telecomunicações, é razoável supor que as tecnologias caminham para a convergência das redes de dados e voz, como forma de oferecer serviços de maior qualidade aos usuários de telecomunicações. Este trabalho propõe um passo inicial nessa direção, com a

implementação de um *gateway* (GW) entre a Rede Inteligente (RI) tradicional e uma nova versão distribuída, para a oferta de novos serviços. Para tal, foi seguida uma especificação feita por pesquisadores da área de telecomunicações e apresentada como resposta a uma *Request for Proposal* (RFP) do Object Management Group (OMG) [1].

O propósito deste artigo é descrever a implementação proposta para uma parte do GW entre a RI tradicional e a RI distribuída, cuja funcionalidade pôde ser isolada e testada de forma independente. A próxima seção traça um panorama do estado da arte das RIs. Na seção 3 são discutidos o modelo de GW do grupo OMG e a proposta de implementação, enquanto a solução desenvolvida é apresentada na seção 4. Finalmente, a seção 5 traz as conclusões do trabalho.

## II. ESTADO DA ARTE DAS RIs

O conceito de RI surgiu no início da década de 80 e sua padronização internacional foi iniciada em 1989 pelo International Telecommunication Union (ITU) e o European Telecommunications Standardization Institute (ETSI).

Fisicamente, a RI é composta por alguns elementos que desempenham as funções da rede [2], como ilustrado na Figura 1.

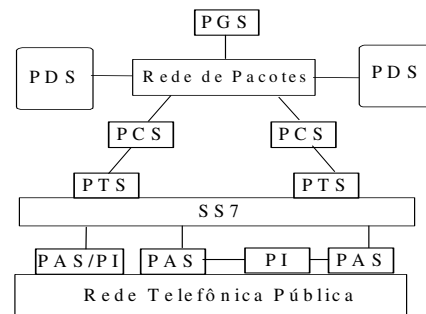


Fig. 1. Arquitetura da RI.

Na realização de um serviço inteligente, o Ponto de Acesso aos Serviços (PAS) recebe uma requisição através da Rede Telefônica Pública (RTP) e encaminha essa requisição ao Ponto de Controle dos Serviços (PCS) através dos Pontos de Transferência de Sinalização (PTSs), que são os roteadores do Sistema de Sinalização por Canal Comum

No. 7 (SS7). O PCS acessa os dados necessários no Ponto de Dados de Serviço (PDS). Um Periférico Inteligente (PI) é acessado quando a RI manda uma mensagem gravada para o usuário. A introdução de novos serviços e a gerência daqueles existentes é feita através do Ponto de Gerência de Serviços (PGS).

Apesar das muitas melhorias introduzidas na oferta de serviços, as RIs ainda têm algumas limitações e desvantagens [3], a saber: isolamento e pouca escalabilidade dos PCSs, que são normalmente sistemas de grande porte e caros; vinculação da criação de novos serviços ao conhecimento detalhado da rede; isolamento da RI em relação a outros tipos de redes; dificuldade de personalização de serviços e criação de serviços particulares, entre outras. Com o intuito de superar esses aspectos negativos, o *Telecommunications Information Network Architecture Consortium (TINA-C)*, um consórcio de empresas que trabalha na especificação de uma arquitetura de serviços padronizada para os sistemas de telecomunicações e informação, criou o grupo de trabalho TINA-IN [4]. Ele desenvolve uma nova arquitetura de implementação de serviços para o domínio das telecomunicações, através da aplicação de sistemas de objetos distribuídos. Essa nova arquitetura de RI é conhecida como *RI Distribuída*.

#### A. A RI Distribuída e o Gateway RI/CORBA

A idéia da RI distribuída apóia-se nos ambientes definidos pelo padrão CORBA [5] para suportar as aplicações referentes aos PASs, PCSs, PDSs e PGSs. Uma vez que essa arquitetura esteja funcionando plenamente, tais entidades físicas poderão não existir mais como são conhecidas hoje, já que as funções que atualmente lhes são atribuídas estarão espalhadas pela rede, comunicando-se através do General Inter-ORB Protocol (GIOP) sobre a SS7 [3]. Esse novo conceito de RI traz vários benefícios, que culminam numa maior facilidade e rapidez para inserir e gerenciar serviços, a um custo bem menor do que o atual.

É importante destacar que outras plataformas também poderiam ser adotadas para a distribuição dos serviços RI. Como exemplo, pode-se citar o Java Remote Method Invocation (RMI), que provê um modelo para computação distribuída com objetos Java. Naturalmente, cada tecnologia tem suas características voltadas para situações específicas.

CORBA não é diferente. Entre seus pontos fortes, pode-se citar a transparência de localização dos objetos na rede, a oferta de diversos serviços que também são utilizados de forma transparente no desenvolvimento de uma aplicação, a escalabilidade de recursos proporcionada e a possibilidade de execução em ambientes heterogêneos, seja no aspecto de plataforma de hardware, sistema operacional ou rede. Tudo isso se traduz em robustez e confiabilidade na implementação de sistemas de objetos distribuídos. Diante disso, e considerando ainda que o ambiente de processamento distribuído sugerido pelo TINA-C é baseado em CORBA, tal padrão é a escolha

natural para este trabalho. Além do mais, a referência principal para o trabalho [1] foi desenvolvida para CORBA.

As restrições financeiras e tecnológicas impulsionam a implantação gradativa dessa arquitetura, estando ainda em princípio de especificação, o que tem levado os fabricantes a desenvolverem soluções proprietárias. Além disso, as operadoras de telecomunicações ainda estão amortizando investimentos feitos na RI tradicional. Tecnicamente, entretanto, a RI distribuída é um campo de pesquisa amplo e necessário devido às características das novas aplicações e dos novos serviços de telecomunicações multimídia.

Neste trabalho, supõe-se que as funcionalidades suportadas por PCS+PDS+PGS são executadas em um ambiente distribuído baseado em CORBA, enquanto os PASs permanecem centralizados (o que pode ser o passo inicial na implantação da arquitetura TINA para a RI). Especificamente, são tratadas as comunicações entre os PASs convencionais e os PCSs distribuídos.

Para a realização do cenário de integração proposto, é necessária uma entidade de ligação entre a RI distribuída e a RI convencional, com a função básica de converter as mensagens da SS7 em invocações de métodos no ambiente CORBA. Esse GW, cuja especificação já é de domínio do OMG [1], é o foco deste trabalho cujo objetivo é desenvolver a implementação de algumas das funcionalidades do GW, de acordo com a especificação proposta e ainda não realizada segundo a pesquisa bibliográfica efetuada. Sua inserção na comunicação entre o ambiente CORBA e a RI convencional pode ser vista na Figura 2.

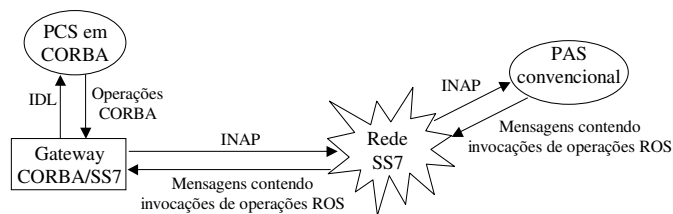


Fig. 2. Interação entre PAS tradicional e PCS CORBA através do Gateway.

Um estudo completo sobre a inserção do GW RI/CORBA na proposta de evolução das RIs pode ser encontrado em [3].

### III. O MODELO DE GW E A IMPLEMENTAÇÃO

A partir da especificação proposta em [1] foi desenvolvida a implementação apresentada neste trabalho. O GW proposto deverá realizar as conversões necessárias de sintaxe e semântica entre os domínios SS7 e CORBA. Na implementação proposta essas conversões se dividem em conversão de especificações e conversão de interações.

A parte de conversão de especificações definiu a maneira como os modelos de informação do protocolo *Intelligent Network Application Part (INAP)* da SS7 [6], a saber, contratos de associação, operações e erros, poderiam ser mapeados da sua notação *Abstract Syntax Notation One*

(ASN.1) para a *Interface Definition Language* (IDL) e vice-versa. Cada contrato de associação entre dois elementos da RI foi mapeado em duas interfaces, uma para um objeto iniciador e outra para o respondedor no diálogo de execução do serviço, como ilustra a Figura 3. Esses objetos são criados a partir de instâncias de fábricas de objetos. Cada interface define um conjunto de assinaturas de operações e exceções em IDL, baseadas nas operações e nos erros definidos dentro do contrato em questão. No cenário para o qual a arquitetura do GW foi definida, o objeto iniciador do diálogo é um *PAS proxy*, o qual representa o PAS convencional no GW. O objeto respondedor é o PCS-CORBA.

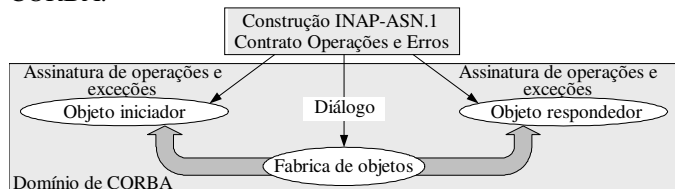


Fig. 3. Mapeamento de Contrato no Ambiente CORBA.

A parte de conversão de interações foi definida por meio de um conjunto de interfaces e objetos, que possuem recursos para localizar, iniciar e manter as associações entre os objetos que executam as aplicações no domínio de CORBA (iniciador e respondedor). Para realizar as conversões de interações, dois conjuntos de interfaces foram definidos: interfaces orientadas a *Protocol Data Units* (PDUs) e interfaces de interação. As interfaces orientadas a PDUs padronizaram o acesso dos objetos PAS proxy à pilha de protocolos da SS7. Trata-se de grupo compreendendo três interfaces: *TcPduUser*, *TcPduProviderFactory* e *TcPduProvider*. As interfaces de interação, por sua vez, possibilitam entre outras coisas a localização de objetos no GW através do Serviço de Nomes, o controle de associações entre os domínios CORBA e a RI convencional, e o acesso a referências para todos os serviços CORBA. São seis interfaces: *TcUser*, *TcFactoryFinder*, *TcUserGenericFactory*, *TcRepository*, *GwAdmin* e *TcServiceFinder*.

Em resumo, o GW RI/CORBA proposto compreende um conjunto de interfaces e objetos, e conta também com os serviços independentes do ambiente CORBA, tais como serviço de localização de objetos (*Naming Service*), serviço para criação e destruição de objetos (*Life Cycle Service*) e serviço para as comunicações assíncronas em tempo real (*Messaging Service*). O Serviço de Mensagens, por ser um padrão recente do OMG à época da proposta do GW, não foi definido dentro da arquitetura especificada.

A Figura 4 ilustra as interfaces que compõem o GW e a seqüência de invocação de métodos entre elas para a completa realização de um serviço RI distribuído. Na região destacada estão as interfaces implementadas neste trabalho. Informações adicionais sobre a forma como as interfaces do GW interagem entre si e um exemplo completo de comunicação entre um PAS convencional e um PCS distribuído através do GW podem ser obtidos em [3]. A

explicação da seqüência de invocação de métodos entre as interfaces implementadas neste trabalho, Figura 4, será feita na sub-seção B.

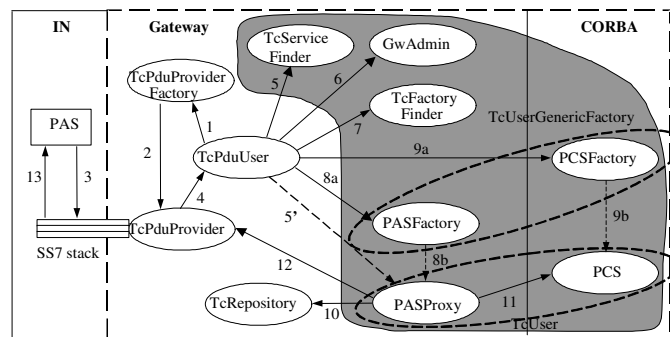


Fig. 4. Interação entre as Interfaces do Gateway.

A proposta apresentada neste trabalho consiste na implementação de duas funções do GW: localização das fábricas que instanciam os objetos de execução dos serviços (iniciador e respondedor) e iniciação do diálogo entre eles. É considerado apenas o núcleo do GW RI/CORBA (região destacada na Figura 4), onde todas as operações são definidas em IDL, sem se preocupar com a interação do GW com a SS7. Assim, as interfaces orientadas a PDUs não foram abordadas neste trabalho, bem como a interface de interação *TcRepository*, que provê acesso aos mapeamentos de identificadores ASN.1 em seus equivalentes em IDL. Isso não é necessário na implementação proposta, já que se supõe que tal mapeamento foi feito em um momento anterior.

#### A. Localização da Aplicação

Para facilitar a localização das aplicações no lado GW/CORBA, é criado um contexto de nomes *GT* (*Global Title*), para ser a raiz da árvore de títulos globais (endereços). Esse contexto de nomes é amarrado diretamente à raiz do *CORBA Naming Service* e uma referência a ele pode ser obtida na interface *TcServiceFinder*.

O ITU definiu as *Transaction Capabilities* (TC) [7] como o meio de se prover sinalização de controle e dados entre os nós da SS7. Na implementação proposta, todas as Entidades de Aplicação (EA) com essas capacidades TC, tais como os PASs e os PCSs, podem ser localizadas por seu título global, o GT. Um GT é adicionado à árvore de nomes como um novo contexto de nomes, abaixo do contexto raiz.

Abaixo de um GT particular existe um contexto de nomes chamado **DefAc**, referente ao contexto de aplicação suportado pelos objetos criados. A ele estão ligados três objetos bem definidos:

- Um objeto chamado "**TcUserFactory**", para a fábrica padrão associada àquela Entidade de Aplicação TC. Através desse objeto são criados o *Iniciador* e o *Respondedor* do diálogo;

- Um contexto “**INIT:<initiator\_interface\_name>**”, onde *initiator\_interface\_name* é o nome do objeto iniciador no domínio CORBA;
- Um contexto “**RESP:<responder\_interface\_name>**”, onde *responder\_interface\_name* é o nome do objeto respondedor no domínio CORBA.

A Figura 5 ilustra a estrutura da árvore de nomes dos títulos globais no CORBA *Naming Service*. A primeira parte deste trabalho foi exatamente definir objetos fábrica e entidades de aplicação que permitissem implementar no CORBA uma estrutura como essa.

Dispondo a árvore de nomes dessa maneira, qualquer objeto CORBA (incluindo objetos proxy em um GW) pode localizar uma fábrica associada a um GT específico, bem como os nomes das interfaces criadas por ela. Para isso, basta buscar o endereço (GT + Contexto de Aplicação) passado como parâmetro na operação invocada.

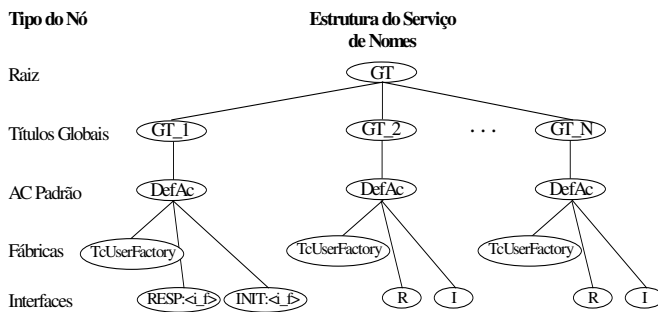


Fig. 5. Árvore de Nomes dos Títulos Globais no CORBA *Naming Service*.

### B. Iniciação do Diálogo

A iniciação de um diálogo no ambiente GW/CORBA pode ser feita basicamente de duas maneiras: invocando-se uma operação de criação de objeto respondedor na interface *TcUserFactory* correspondente; ou invocando-se uma operação de nova associação em um objeto respondedor já existente. Neste trabalho escolheu-se a primeira opção, pois a intenção era criar os objetos a partir da interface-fábrica, para que fosse possível utilizar a árvore de nomes dos títulos globais descrita anteriormente.

O processo de localização de aplicação e iniciação de um diálogo é mostrado no diagrama da Figura 6. Ele considera o caso da iniciação de um diálogo e invocação de operação no domínio CORBA e resume a segunda parte deste trabalho, que utiliza a árvore de nomes dos títulos globais para recuperar a referência a uma fábrica procurada. Os passos envolvidos, resumidamente, são:

a) O *TcUser* (PAS proxy) usa a interface *GwAdmin* para obter a referência ao objeto *TcFactoryFinder* do GW;

b) No objeto *TcFactoryFinder*, o *TcUser* invoca a operação **resolve** para obter a referência ao objeto fábrica associado ao GT passado como parâmetro na invocação. O objeto *TcFactoryFinder* faz uso do CORBA *Naming Service* (não mostrado no diagrama) para realizar a recuperação do

objeto (assume-se que as ligações de nomes foram devidamente realizadas em algum momento anterior);

c) A referência ao objeto *TcUserFactory* gerada pela resolução de endereço é usada para criar uma instância do objeto alvo em CORBA, o PCS;

d) O PAS proxy invoca no PCS a operação INAP descrita em IDL;

e) O resultado da operação invocada é retornado ao PAS, o qual então termina a associação.

Na próxima seção são apresentadas as implementações desenvolvidas neste trabalho, baseadas nas descrições anteriores, assim como algumas opções de projeto e os resultados obtidos.

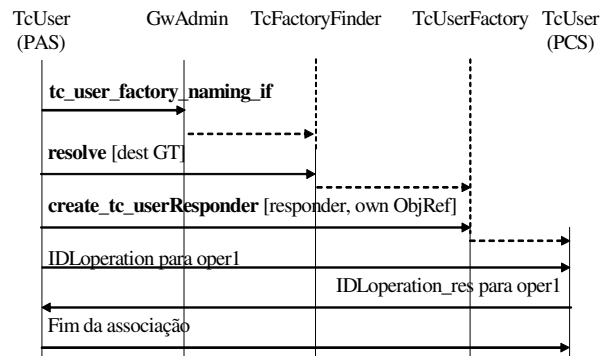


Fig. 6. Localização de Aplicação e Iniciação de Diálogo no GW.

## IV. IMPLEMENTAÇÕES E TESTES COMPUTACIONAIS

As interfaces pertencentes ao GW implementadas neste trabalho foram: *TcUser*; *TcUserGenericFactory*; *TcFactoryFinder*; *GwAdmin* e *TcServiceFinder*. Também foram implementadas interfaces geradas a partir de Conversão de Especificação, que representam justamente os objetos reais do processo de comunicação (por exemplo, o PAS proxy e o PCS). Elas têm a particularidade de ocorrerem sempre três a três, sendo que duas descrevem as Entidades de Aplicação consideradas (Iniciador e Respondedor) e a outra descreve a fábrica daquelas. Neste trabalho foram implementados dois conjuntos dessas interfaces, para permitir testes mais completos. Os relacionamentos entre as interfaces implementadas são ilustrados na Figura 7.

Toda aplicação CORBA deve contar com um *servidor* e pelo menos um *cliente*. Neste projeto, implementou-se uma classe *Server*, que faz a criação dos objetos do GW e ligação destes no serviço de nomes, e três *Clients*, que desempenham o papel dos iniciadores dos diálogos, recuperando uma fábrica a partir do serviço de nomes (via interface *TcFactoryFinder*), criando objetos respondedores e invocando operações sobre eles. Foram definidas as classes *Client*, *Client2* e *Client3*, para os testes da implementação.

Nesta implementação foi utilizado um ORB comercial compatível com CORBA e distribuído como software livre,

conhecido como JacORB. Ele foi escolhido por sua segurança e estabilidade e por prover uma implementação confiável do Serviço de Nomes (*Naming Service*) do CORBA, ao qual foi amplamente utilizado [8].

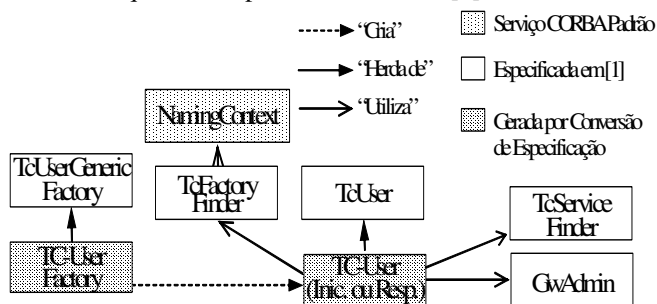


Fig. 7. Relacionamentos entre as Interfaces Implementadas.

Com relação ao desempenho, a implementação do GW desenvolvida não objetiva gerar resultados que possam ser comparados a dados reais, como o tempo de processamento de um serviço RI, por exemplo. A parte do problema que foi executada não comporta esses testes. Entretanto, houve uma preocupação constante com o desenvolvimento de um sistema que fosse eficiente e robusto. Um exemplo disso é a forma como os objetos da aplicação foram implementados, tirando máximo proveito da escalabilidade de recursos provida por CORBA.

Nas sub-seções seguintes são descritas todas as classes implementadas.

#### A. As Interfaces das Entidades de Aplicação

Neste trabalho foram implementados dois pares de EAs, de maneira a permitir testes mais completos sobre a arquitetura do GW, gerando as interfaces *SSP\_SCP\_Initiator*, *SSP\_SCP\_Responder*, *IP\_SCP\_Initiator* e *IP\_SC\_Responder*. As duas primeiras representam a comunicação entre um PAS e um PCS, e as outras duas, entre um PI e um PCS.

Em cada uma dessas interfaces foram definidas duas operações para a realização dos testes, chamadas *oper1* e *oper2*, que representam operações reais numa RI.

#### B. As Interfaces Fábrica

São definidas duas interfaces fábrica: *TcUserFactory*, que cria os objetos *SSP\_SCP\_Initiator* e *SSP\_SCP\_Responder*, e *TcUserFactory2*, para criação dos objetos *IP\_SCP\_Initiator* e *IP\_SCP\_Responder*. Assim como as interfaces das EAs, as interfaces fábrica também são originadas da Conversão de Especificação, porém não diretamente como aquelas. De fato, estas são definidas a partir dos objetos que elas devem criar, e contam sempre com pelo menos três métodos: um para criação do objeto iniciador e dois para criação do objeto respondedor. Tais métodos retornam objetos dos tipos definidos pelas interfaces das Entidades de Aplicação.

De acordo com a especificação do GW, toda interface fábrica deve herdar os atributos e métodos de uma superclasse que especifica métodos genéricos para criação

dos objetos iniciadores e respondedores. *TcUserFactory* e *TcUserFactory2* herdam da superclasse *TcUserGenericFactory*. A herança dessa superclasse por parte das fábricas é um dos grandes méritos da especificação do GW utilizada, pois permite um isolamento da aplicação CORBA em relação aos detalhes de implementação das EAs e suas fábricas. Por exemplo, da forma implementada, a aplicação só precisa invocar o método *create\_tc\_userInitiator* na fábrica correspondente, independente de qual objeto se deseja criar. Os métodos da superclasse, implementados em cada fábrica de objetos, invocam os métodos adequados destas para a criação do objeto requisitado.

#### C. As Interfaces de Definição de Superclasses

O projeto do GW especificou duas interfaces que são usadas exclusivamente como superclasses para outras classes. São as classes *TcUser*, superclasse de todas as classes que descrevem EAs, e *TcUserGenericFactory*, superclasse de todas as classes fábrica.

Como dito anteriormente, o uso dessas superclasses permite o acesso genérico aos objetos iniciadores, respondedores e fábricas nas aplicações do servidor e do cliente de CORBA. Isso torna a programação da aplicação distribuída muito mais simples de ser feita, proporcionando o acesso a diferentes objetos de maneira uniforme.

#### D. A Interface TcFactoryFinder

A interface *TcFactoryFinder* está ligada ao processo de localização de uma aplicação RI no domínio CORBA. É através dela que se constrói a árvore de nomes dos GTs da Figura 5, e também se recuperam as informações ali armazenadas - as referências às interfaces fábrica e os respectivos nomes de escopo dos objetos criados por elas.

O *TcFactoryFinder* define quatro métodos para manipulação da árvore de nomes: *bind*, *unbind*, *rebind* e *resolve*. Todos eles utilizam o CORBA *Naming Service*, que torna suas programações consideravelmente mais simples do que se fosse necessário definir uma estrutura de armazenamento em árvore, e os métodos de acesso a ela.

#### E. Outras Interfaces de Serviço do Gateway

O GW RI/CORBA conta com dois objetos que provêm operações auxiliares para administração e localização de serviços: *GwAdmin* e *TcServiceFinder*. O primeiro define os métodos *tc\_user\_factory\_naming\_if* e *tc\_pdu\_handler\_factory\_if*, que retornam referências para as interfaces *TcFactoryFinder* e *TcPduProviderFactory* do GW. O segundo contém os métodos *\_gt\_root*, *gw\_admin\_if* e *tc\_repository\_if*, que devolvem referências para o contexto de nomes raiz do GW e para os objetos *GwAdmin* e *TcRepository*, respectivamente. As referências aos objetos retornados por todos esses métodos são passadas como parâmetro através dos métodos construtores dessas interfaces.

### F. A Implementação do Servidor CORBA

O objeto servidor da aplicação foi implementado na classe *Server*. Essa classe é responsável pela definição e criação dos objetos do GW, ligação destes ao serviço de nomes do CORBA (quando necessário), e ligação das fábricas de objetos ao domínio de nomes particular do GW (através da interface *TcFactoryFinder*). Naturalmente, nessa classe também são iniciados o ORB, o *Portable Object Adapter* (POA) e o Serviço de Nomes do CORBA.

Cabe ressaltar que, graças à forma como a especificação do GW utiliza a herança de classes, foi necessário definir um único objeto-fábrica para referenciar genericamente todas as fábricas utilizadas no GW. Assim, o objeto *TcUserGenericFactory*, superclasse de qualquer fábrica do domínio do GW, pode criar qualquer objeto TC através dos métodos definidos em sua interface.

No objeto servidor são realizadas a criação das fábricas dos objetos TC e a ligação destas ao domínio de nomes particular do GW. Nesta implementação, foram definidos três títulos globais na árvore de nomes: dois deles para a criação de iniciadores e respondedores em um diálogo entre um PAS e um PCS, e o terceiro para a criação de iniciadores e respondedores no diálogo entre um PI e um PCS. Ao primeiro e segundo títulos globais (diálogo PAS / PCS) foi adicionada a mesma fábrica, porém com nomes de contexto distintos para iniciador e respondedor. Isso comprova a escalabilidade e a robustez providas pelo GW em situações em que é necessária a criação de uma nova instância do mesmo objeto com um nome de escopo diferente (para que não haja conflito). Para o terceiro e último título global (diálogo PI / PCS) é criada uma nova fábrica de objetos.

Com os objetos do GW criados e as fábricas de objetos TC devidamente adicionadas à árvore de nomes, o servidor CORBA está apto a dar suporte às requisições dos clientes.

### G. A Implementação dos Clientes CORBA

Como explicado na seção 3, foi implementado neste trabalho o processo de iniciação de um diálogo entre um objeto iniciador e um objeto respondedor, de acordo com os passos indicados na Figura 6. Mais precisamente, foi simulado um objeto iniciador (PAS ou PI no caso) invocando operações sobre outros objetos do GW (*GwAdmin*, *TcFactoryFinder*, etc), de maneira a executar uma tarefa da RI.

Após analisar algumas alternativas, concluiu-se que aquela que faria mais sentido para uma aplicação RI seria a implementação do iniciador como o *cliente* do ambiente CORBA, sendo executado conforme a necessidade da situação. Em um problema real, esse cenário implica na execução de um objeto cliente de CORBA quando uma PDU de início de diálogo é recebida.

É importante destacar que, ao testar a simulação, executava-se o servidor e em seguida os três clientes CORBA, o que assegura que eles pesquisavam na mesma

árvore de nomes do GW. Os valores impressos na execução dos métodos *oper1* e *oper2* em cada cliente, bem como o nome do respondedor criado, atestaram o sucesso da implementação.

### V. CONCLUSÕES

Neste trabalho foi apresentada uma proposta de implementação de parte de um GW de comunicação entre uma Rede Inteligente centralizada tradicional e a Rede Inteligente distribuída através de CORBA. A idéia do trabalho acompanha a tendência cada vez mais forte de integração entre as redes de dados e voz, e pode ser considerada um primeiro passo para a oferta futura de serviços RI do tipo banda larga com mais qualidade e facilidade de implantação e manutenção, através dos objetos distribuídos via CORBA.

Foram implementados os mecanismos do GW responsáveis pela *localização da aplicação e iniciação do diálogo entre EAs*. Os testes realizados permitiram certificar o funcionamento correto da implementação proposta. Confirmou-se que a árvore de nomes dos títulos globais do GW foi corretamente construída, as fábricas foram corretamente adicionadas e recuperadas pelos métodos do localizador de fábricas, e os clientes acessaram perfeitamente os objetos do GW, realizando com sucesso a simulação de serviço RI distribuído, desde a recuperação do administrador do GW a partir do Serviço de Nomes do CORBA até a invocação das operações RI no objeto respondedor criado pela fábrica correspondente.

Um aspecto a se destacar é a forma como foi utilizada a herança de classes na implementação. Também se nota como ponto positivo a estrutura adotada para a árvore de nomes do GW, e ainda a escolha de se implementar o objeto iniciador do diálogo TC como um cliente CORBA. A proposta apresentada também possui a vantagem de ser baseada totalmente em padrões abertos – IDL para CORBA e ASN.1 para a SS7, o que diminui os custos de implementação.

Finalmente, a sugestão natural para trabalhos futuros é a implementação das interfaces orientadas a PDUs, o que tornará o GW completo.

### REFERÊNCIAS

- [1] OMG, “Interworking Between CORBA and TC Systems”, *Document Number: telecom/98-10-0319*, 1998.
- [2] ITU-T, “Intelligent Network Physical Plane Architecture”, *Recommendation Q.1205*, 1993.
- [3] R. B. Soares, “Estratégias de Introdução de Sistemas Distribuídos nas Redes Inteligentes”, *Tese de Doutorado*, Universidade Federal do Espírito Santo, 2003.
- [4] TINA-C, “TINA-IN Workgroup”, *Disponível em <http://www.tinac.com/>*, 1998.
- [5] CORBA, Site Oficial, <http://www.corba.org/>, 2004.
- [6] ITU-T, “Interface Recommendation for Intelligent Network CS-1”, *Recommendation Q.1218*, 1995.
- [7] ITU-T, “Functional Description of Transaction Capabilities”, *Recommendation Q.771*, 1993.
- [8] JacORB, Site Oficial, <http://www.jacorb.org/>, 2004.