

# Um Meta Modelo para a Engenharia de Serviços de Telecomunicações\*

SÉRGIO COLCHER, ANTÔNIO TADEU A. GOMES, LUIZ FERNANDO G. SOARES  
{colcher, atagomes, lfgs}@inf.puc-rio.br

Departamento de Informática – PUC-Rio  
R. Marquês de São Vicente, 225, Rio de Janeiro – RJ, 22453-900, Brazil  
Tel.: +55 (21) 274-2731 r. 3504, Fax.: +55 (21) 259-2232

**Resumo.** A Engenharia de Serviços de Telecomunicações é uma nova disciplina que tem como principal objetivo tratar dos aspectos de especificação, projeto, implementação, gerenciamento e validação de serviços de telecomunicações e sua implantação sobre arquiteturas de rede existentes e futuras. Um dos seus maiores desafios tem sido o de orientar a construção de uma infra-estrutura de comunicação que exiba duas características básicas: (i) a integração da mais variada gama de diferentes serviços em um único e eficiente sistema de comunicação, e (ii) a criação de um ambiente no qual esses mesmos serviços possam ser rápida e facilmente criados, alterados e continuamente adaptados às novas necessidades e condições. Torna-se, porém, cada vez mais difícil encontrar uma arquitetura única, genérica o suficiente para acomodar os diversos serviços e adaptações através de simples parametrizações ou opções predefinidas. A alternativa, recentemente considerada, é a definição de modelos nos quais arquiteturas de protocolos e serviços podem ser dinamicamente programados ou adaptados ao longo de seu ciclo de vida. Este trabalho apresenta um modelo para a análise e comparação de mecanismos de adaptação e programação de serviços em redes de comunicação, fornecendo uma base para a representação de aspectos de QoS e comunicação de grupo em serviços de comunicação.

**Abstract.** The Telecommunication Services Engineering is a new subject which aims to deal with specification, design, implementation, management and validation aspects of telecommunication services, as well as their deployment over current and future network architectures. One of its main challenges has been to guide the design of a communication infrastructure that allows: (i) the integration of a multitude of services in a single and efficient communication system, and (ii) the creation of an environment upon which these services can be rapidly and easily created, modified and continually adapted to new demands and conditions. Nevertheless, it has become a difficult task to find a sufficiently generic architecture that accommodates the diversity of services and adaptations through simple parameterization or predefined options. A recently proposed alternative is to define models from which protocols and service architectures can be dynamically programmed or adapted along its life cycle. This work presents a model for the analysis and comparison of service adaptation and programming mechanisms on communication networks, providing a basis for the representation of QoS and multicasting aspects on communication services.

## 1 INTRODUÇÃO

No atual estágio de desenvolvimento e competitividade do setor de telecomunicações, a habilidade de se oferecer uma vasta gama de serviços com características diferenciadas e perfeitamente sintonizadas com os requisitos das aplicações tem se tornado imprescindível. Em um cenário como esse, diversas atividades como planejamento, projeto, implementação e implantação têm se demonstrado tão complexas a ponto de justificar a criação de uma “Engenharia de Serviços de Telecomunicações” [36].

Engenharia de Serviços de Telecomunicações é uma nova disciplina que tem como principal objetivo tratar dos aspectos de especificação, projeto, implementação, gerenciamento e validação de serviços de telecomunicações e sua implantação sobre arquiteturas de rede existentes

---

\*Trabalho parcialmente financiado pela Empresa Brasileira de Telecomunicações (EMBRATEL).

e futuras. Um dos seus maiores desafios tem sido o de lidar com os problemas de integração dos diferentes serviços e da necessidade da rápida criação e contínua adaptação desses serviços.

As principais características dessa nova disciplina podem ser consideradas como parte da intersecção de várias áreas, como as de telecomunicações, comunicação de dados, engenharia de software e sistemas multimídia. Os novos desafios que têm sido lançados à indústria de telecomunicações e às operadoras de serviços parecem justificar essa convergência, revelando problemas inerentes a essa combinação e, ao mesmo tempo, ausentes em cada uma daquelas áreas isoladas. Em última análise, justamente da complexidade proveniente dessa interdisciplinaridade é que surge a motivação para a realização do presente trabalho, cujo objetivo maior é fornecer um ponto de apoio e referência para um maior desenvolvimento de uma área que encontra-se, hoje, ainda em seus primórdios.

A sofisticação das novas aplicações multimídia distribuídas sugere o alto grau de complexidade exigido dos serviços de comunicação e as dificuldades associados ao projeto dessas aplicações. Grande parte dessas dificuldades consiste em oferecer serviços capazes de manter as características necessárias a toda a diversidade de tipos de fluxo. Torna-se cada vez mais difícil encontrar uma arquitetura única, genérica o suficiente para acomodar os diversos serviços e adaptações através de simples parametrizações ou opções predefinidas. A alternativa, recentemente considerada, é a definição de modelos nos quais arquiteturas de protocolos e serviços podem ser dinamicamente programados ou adaptados ao longo de seu ciclo de vida. Essa *adaptabilidade* é uma característica desejável tanto para acomodar e absorver os avanços tecnológicos que acontecem cada vez mais rapidamente, como para permitir a criação de novos serviços e aplicações que apresentam requisitos específicos de QoS.

O presente trabalho visa propor um modelo que forneça abstrações adequadas para a representação e programação de aspectos de QoS e comunicação de grupo em serviços de comunicação. Tal modelo deverá servir como base para a análise e comparação de mecanismos de adaptação e programação de serviços em redes de comunicação. Qualquer implementação de um ambiente de suporte a serviços deverá conter, porém, outras propriedades não capturadas pelo modelo aqui proposto. De fato, a intenção deste trabalho não será a de propor um modelo que cubra todos os aspectos necessários a uma implementação, mas sim a de estabelecer uma base conceitual que tornará possível o desenvolvimento de outros modelos mais específicos. Dentro dessa linha de raciocínio, é importante frisar que o presente modelo *não deve ser interpretado* como um “modelo definitivo” que, quando implementado, irá sobrepujar ou tornar quaisquer outros obsoletos. Ao contrário, o modelo será, propositadamente, incompleto demais para ser diretamente implementado, refletindo o fato de que o objetivo foi sempre o da generalidade [7].

Ao se considerar o aspecto da adaptabilidade, não se pode negligenciar o fato de que o impacto no processo de construção e operação dos serviços não se verifica apenas sob os aspectos técnicos, ou de suporte computacional, mas também organizacionais. Não se tem por objetivo, neste trabalho, propor uma nova metodologia de desenvolvimento, nem tão pouco um novo paradigma para operação e administração de serviços. Considerou-se fundamental, porém, salientar os pontos em que um ciclo de vida será influenciado pela possibilidade e dinamismo das adaptações que poderão ser efetuadas sobre os serviços.

## 2 TRABALHOS RELACIONADOS

Vários trabalhos têm focado a forma com que serviços de comunicação são desenvolvidos, operados e mantidos. Alguns deles tratam do problema sem propor mudanças efetivas no ciclo de vida dos serviços, preferindo focar somente o lado da rapidez no processo de criação dos

serviços. As técnicas que têm sido utilizadas para agilizar o processo de desenvolvimento de serviços incluem: (i) a utilização de *design patterns* [13], aplicados ao domínio específico da construção de serviços, protocolos e software de comunicação [31, 21]; (ii) a criação de linguagens específicas para construção de protocolos e serviços de comunicação [1, 25], e (iii) o desenvolvimento de bibliotecas específicas [30, 23], juntamente com a introdução de mecanismos de suporte nos sistemas operacionais [16, 11].

Todas essas técnicas, porém, não lidam diretamente com a possibilidade de adaptações realizadas ao longo da operação de um serviço. No presente trabalho, procurou-se uma abordagem que permitisse aliar as técnicas acima mencionadas com outras que influenciassem no ciclo de vida dos serviços e permitissem formas de adaptação mais dinâmicas. Um dos pontos de partida foi o estudo da utilização de *frameworks* na modelagem de serviços de comunicação [14, 29] que, embora ainda não tenham sido, nessa área específica, tão explorados quanto os *design patterns*, já tiveram sua importância reconhecida em algumas instâncias [15].

Um *framework* [26] difere de uma aplicação pelo fato de que algumas de suas partes (denominadas *pontos de flexibilização* ou *hot-spots*) são explícita e propositadamente deixadas incompletas ou marcadas como “sujeitas a variações”. Através do respectivo preenchimento dessas “lacunas” ou de modificações nos pontos marcados, pode-se obter a adaptabilidade do serviço. O momento dentro do ciclo de vida de um serviço em que *hot-spots* são completados, que pode variar desde as fases de construção e implementação até as fases de operação, determina o seu grau de dinamismo. Os *hot-spots* complementados nas fases de construção (mais precisamente na fase de *instanciação* do *framework*) dão origem a serviços específicos que, por sua vez, ainda podem apresentar outros *hot-spots* a serem complementados, modificados ou estendidos em tempo de operação.

Modelos de objetos e arquiteturas tem sido, cada vez mais, tratados de forma integrada nos trabalhos relacionados a engenharia de serviços de telecomunicações. Os principais trabalhos que seguem essa estratégia baseiam-se no suporte oferecido por ambientes baseados em plataformas definidas pelo CORBA [24] ou pelo ODP [17]. O modelo proposto neste trabalho pode ser comparado às visões de *informação*, *computacional* e de *engenharia* do ODP. Porém, a divisão dessas visões não é feita de forma estática; a própria recursividade do modelo proposto faz com que a visão computacional se transforme na visão de engenharia de um outro nível.

A questão da *reflexividade* em plataformas de componentes baseadas no ODP é abordada nos trabalhos descritos em [5, 9]. No presente trabalho, porém, três características podem ser ressaltadas como diferenças básicas em relação a essas propostas: (i) a existência de uma organização em camadas, que regula a definição dos mecanismos de adaptabilidade, (ii) a definição de *frameworks*, que correspondem a estruturas ou modelos estabelecidos de forma a regular as adaptações a serem feitas no ambiente e (iii) a definição de uma modelagem que captura, especificamente, as abstrações comuns presentes em serviços de comunicação. Em relação a essa terceira característica, já existem modelos baseados no ODP especificamente voltados aos aspectos de serviços de comunicação como o TINA [3], que procura encarar sistemas de comunicação como plataformas de desenvolvimento e programação de serviços baseados nos conceitos de orientação a objetos.

Esforços semelhantes aos mencionados acima são encontrados também para plataformas baseadas no padrão CORBA, como os descritos em [10, 32]. A maioria dessas propostas, no entanto, não menciona explicitamente os mecanismos de adaptabilidade de serviços, estando mais voltadas ao aspecto das abstrações para programação com qualidade de serviço.

O foco nos mecanismos de adaptabilidade é encontrado com maior frequência nos trabalhos

relacionados a redes inteligentes (Intelligent Networks — IN) [18], redes programáveis [19, 35] e redes ativas [34, 27]. Em particular, o trabalho descrito em [6] procura fornecer um modelo que compare as estratégias utilizadas na adaptação de serviços nas diferentes propostas de arquiteturas para redes ativas e redes programáveis. O foco das comparações, porém, foi feito com base na definição de uma arquitetura genérica para os nós da rede que compõem os respectivos sistemas de comunicação. A classificação apresentada na Seção 5.2 do presente trabalho pode ser considerada mais genérica, no sentido de que ela está baseada somente nos elementos do modelo conceitual definido para os serviços, que é, na realidade, um meta modelo a ser utilizado para a construção de arquiteturas. Nesse meta modelo, a adaptabilidade de um sistema de comunicação é vista sob o prisma de uma *base de informações distribuída* que regula e auxilia no funcionamento de todo o sistema. Um exemplo clássico é a introdução das bases de informação de gerência.

A evolução dos serviços e, por conseguinte, o crescente volume e complexidade das informações manipuladas e contidas na base de informação criaram a necessidade de novos modelos e ferramentas para criação e manutenção dos serviços. Além do TINA, outros esforços surgiram com o propósito de construir um modelo de referência para a construção de serviços de telecomunicações, baseado em avanços oriundos da área de programação distribuída e orientada a objetos. O *Binding Model* [20], por exemplo, define a base conceitual para criação de serviços de comunicação utilizando-se de uma *base de informações distribuída* denominada *Binding Interface Base (BIB)* sobre a qual os mecanismos de associação deverão operar.

O conjunto de esforços para transformar e encarar sistemas de comunicação como plataformas de desenvolvimento e programação de serviços deu origem ao conceito de “redes programáveis”, cuja principal característica é a de propor o fornecimento de um modelo que permite o acesso aos recursos internos de cada equipamento e a modelagem das bases de informação necessárias para a construções dos serviços. Uma estratégia que pode ser considerada ainda mais agressiva foi adotada pelos grupos que propõem a construção de *redes ativas*. A idéia básica pode ser considerada como a mesma das redes programáveis, porém, nesses trabalhos, o próprio código da implementação dos protocolos internos à rede passa a se tornar uma informação disponível na base de informações, que pode ser alterada, inserida e manipulada tanto por usuários como pela operadora durante a fase de operação. Dessa forma, redes ativas permitirão que os equipamentos da rede, vistos como uma plataforma distribuída, executem procedimentos injetados em seus nós ao longo da operação do serviço. No casos mais extremos, pacotes convencionais serão substituídos por fragmentos de código (e informações), que serão executados a cada nó de rede do caminho em que passarem.

### 3 META MODELO PARA APLICAÇÕES E SERVIÇOS DE COMUNICAÇÃO ADAPTÁVEIS

Os elementos de modelagem básicos utilizados para compor serviços são: (i) os *componentes usuários* (ou simplesmente *componentes*), que correspondem a entidades que utilizam diretamente os serviços, e (ii) os *provedores*, que são os responsáveis pelo funcionamento dos serviços durante sua operação. Usuários e provedores são, em geral, equipamentos e produtos de hardware ou software. Denomina-se de um *ambiente de oferecimento de serviços* a um conjunto de usuários e os respectivos provedores. A Figura 1 ilustra um ambiente de oferecimento de serviços com um provedor e quatro usuários.

A definição de serviços de comunicação é abstrata no que concerne a dispersão dos usuários. Usuários podem, por exemplo, estar tão próximos quanto objetos implementados em uma mesma linguagem de programação e ativados em um mesmo processo de um sistema opera-

cional de uma máquina isolada, ou tão distantes quanto objetos localizados em máquinas de sub-redes distintas. Um sistema de comunicação exige mais que a definição de seus elementos. Um modelo deve também capturar as restrições básicas de estruturação desses elementos para a construção de serviços válidos.

Cada componente tem uma *interface*. Interfaces definem como e quais informações são trocadas entre componentes, estabelecendo o seu comportamento observável perante o provedor de serviços e os outros componentes, i.e., suas formas de cooperação. Para que haja cooperação, assume-se que alguma forma de *associação* existe entre os componentes que desejam se comunicar. Associações podem ser *explícitas* ou *implícitas*.

A principal característica da associação implícita é a ausência de uma entidade que represente a associação. Em associações implícitas, componentes usuários utilizam diretamente alguma identificação dos componentes com quem se comunicam. A forma com que essa identificação é fornecida pode variar desde a codificação de endereços pré-estabelecidos até a obtenção dinâmica de identificações através da comunicação com outros componentes. A partir dessa identificação, em uma comunicação com associação implícita, o provedor é responsável por encaminhar os fluxos de informação aos destinos.

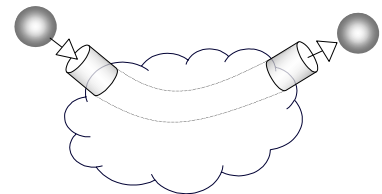
Em associações explícitas, por outro lado, entidades especiais são criadas de forma a conectar pontos terminais de fluxos de informação (componentes). Uma associação explícita tem um ponto terminal fonte e um ou mais pontos terminais de destino, correspondendo, portanto, a uma abstração ponto-a-multiponto unidirecional denominada *pipe*. Componentes que se comunicam através de associações explícitas são denominados *filtros*. Assim, filtros são componentes que interagem diretamente com o provedor, solicitando dele a capacidade de envio de informações a outros filtros por meio de pipes, conforme a representação da Figura 2. Filtros são componentes que possuem referências diretas a pipes e não a outros componentes, como na associação implícita. Pipes podem definir ou restringir os tipos de dados das mensagens que transportam. Aos pipes podem ser associadas especificações de QoS utilizadas para definir e regular a forma de comunicação entre os filtros que os utilizam. Pipes que são associados a especificações de QoS passam a ser denominados *MediaPipes* [8].

A implementação de um provedor é realizada com a utilização de elementos denominados *componentes de implementação do serviço* (ou, simplesmente, *componentes do serviço*). Para desempenhar suas funções, componentes de serviço comunicam-se utilizando-se dos serviços fornecidos por outros provedores mais primitivos, cada qual denominado *provedor de infra-estrutura* (ou, simplesmente, *infra-estrutura*) do serviço, conforme ilustrado na Figura 3(a).<sup>1</sup>

Além disso, todo provedor de serviços deverá fornecer uma forma de acesso para que usuários se comuniquem com componentes de implementação do serviço. Um tal serviço de comu-

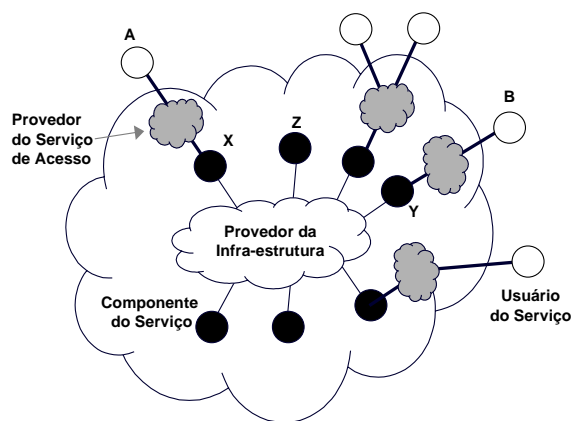


**Figura 1:** Um exemplo de ambientes de oferecimento de serviços.

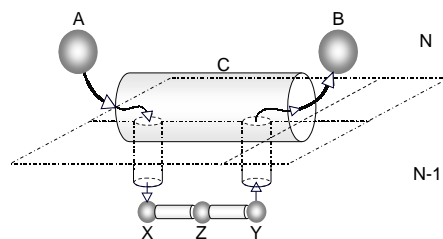


**Figura 2:** Representação gráfica de associações explícitas (pipes) entre os filtros.

<sup>1</sup>Por simplicidade, limitou-se a apresentação dos elementos do modelo utilizando apenas uma única infra-estrutura. Exemplos de provedores com mais de uma infra-estrutura podem ser encontrados em [7].



(a) Infra-estrutura e acesso.



(b) Estrutura em camadas de um Pipe.

**Figura 3:** Estruturação de provedores de serviços.

nicação é denominado *serviço de acesso*, conforme ilustrado nessa mesma figura.

O serviço oferecido por uma infra-estrutura fornece o suporte necessário aos componentes do serviço para que esses se comuniquem e, coletivamente, implementem as funções que realizarão o comportamento esperado do provedor. Em sendo um provedor, a infra-estrutura também pode ser recursivamente estruturada, fazendo com que os componentes de serviço que aparecem na Figura 3(a) passem a ser vistos como usuários e, internamente, novos componentes de implementação de serviço, novos provedores de infra-estrutura e de acesso apareçam. A mesma estruturação recursiva pode existir no provedor de serviço de acesso.

Esse aninhamento de provedores pode ser, em certos casos, utilizado como um conceito semelhante ao princípio de divisão em camadas definido por arquiteturas hierárquicas como o OSI-RM. Nesse caso, componentes de serviço poderiam ser associados a entidades de protocolo de uma camada  $N$ , com a infra-estrutura correspondendo à camada  $N-1$ , e usuários a entidades de protocolo da camada  $N+1$ .

A Figura 3(b) ilustra a abstração de um MediaPipe  $C$  (em um nível  $N$ ) entre os usuários  $A$  e  $B$  (filtros) do serviço de comunicação com sua estrutura (no nível  $N-1$ ) interna revelada. Ao revelar a estrutura de  $C$ , obtém-se a visão do provedor de nível inferior ( $N-1$ ) que, nesse exemplo, contém dois MediaPipes verticais e dois horizontais, além dos componentes  $X$ ,  $Y$  e  $Z$ . Todos esses elementos (filtros e MediaPipes) são conjuntamente responsáveis pelo comportamento da abstração do MediaPipe do nível  $N$ . Os MediaPipes verticais representam as associações explícitas estabelecidas nos provedores de acesso (entre  $A$  e  $X$ , e entre  $B$  e  $Y$ ), enquanto que os horizontais correspondem a associações estabelecidas através da infra-estrutura, de acordo com a representação da Figura 3(a). Cada um dos MediaPipes do nível  $N-1$  poderia ser então, recursivamente, estruturado de maneira similar ao MediaPipe  $C$ .

É importante salientar que as correspondências do modelo proposto com as camadas do OSI-RM são apenas aproximações, haja visto que o provedor de serviço ainda apresenta outras características ausentes nos modelos tradicionais. Apenas como um pequeno exemplo dessa diferença, com grandes consequências na provisão de QoS, pode-se observar que o serviço de acesso é o correspondente a um mecanismo de comunicação local entre entidades de duas camadas adjacentes. Esse tipo de comunicação é considerada fora do escopo do OSI-RM, que delega tal tratamento aos sistemas específicos e seus respectivos ambientes locais (Local System Envi-

ronments — LSEs). Ao se considerar sistemas onde garantias de QoS são necessárias, todas as partes integrantes da comunicação devem ser cuidadosamente projetadas para a construção de um sistema onde os requisitos fim-a-fim são realmente satisfeitos. Por essa razão, vários grupos de trabalho têm proposto abordagens que tratam da arquitetura do sistema como um todo na solução do problema da provisão de QoS fim-a-fim, sendo coletivamente denominadas de *arquiteturas de QoS* [2].

Uma das grandes dificuldades encontradas em arquiteturas de QoS surge da diversidade com que os aspectos de QoS são especificados e, posteriormente, a forma com que eles são tratados nos diferentes sistemas que compõem o ambiente de comunicação e processamento. A modelagem de serviços de acesso por meio da abstração de um provedor, abstração idêntica à utilizada na modelagem da comunicação entre quaisquer usuários independente de sua dispersão, propõe uma forma de tornar mais homogêneos o tratamento da QoS em ambientes de processamento e de comunicação. Os frameworks de QoS [14] e multicast [29] propostos para o modelo confirmam essa homogeneidade através de uma modelagem única, na qual parte dos hot-spots são responsáveis justamente pelo tratamento das questões específicas de cada ambiente.

#### 4 ARQUITETURA DO NÍVEL META

Para que um ambiente de oferecimento de serviços possa sofrer adaptações, é necessária a existência de um sistema (ou sistemas) que permite adaptações a esse ambiente — uma espécie de “meta sistema”. Um *meta sistema* é um sistema que age ou dá significado a um outro sistema. No caso de serviços de comunicação, meta sistemas agem sobre serviços, sendo por essa razão denominados *meta serviços*. A arquitetura na qual os elementos de um serviço estão representados e disponíveis para a adaptação por meio de um meta serviço será denominada *arquitetura do nível meta*, como em [33]. A forma com que meta serviços são implementados pode variar desde procedimentos manuais ou ferramentas simples de instalação até ambientes complexos de processamento distribuído. Alguns exemplos de meta serviços comuns são os mecanismos de sinalização e gerência, e os protocolos de roteamento, como será explicado mais adiante.

Meta serviços podem atuar sobre serviços sem qualquer restrição de nível. Além disso, o mesmo provedor pode ser o alvo de vários meta serviços e um mesmo meta serviço pode agir sobre mais de um provedor, inclusive de níveis diferentes. O serviço sobre o qual um meta serviço atua é dito o *serviço alvo*. Alguns tipos de meta serviços permitem adaptações de forma dinâmica e distribuída, como no caso dos mecanismos de sinalização, redes programáveis e redes ativas. Em todos esses casos, o meta serviço é estruturado como um ambiente de oferecimento de serviços, no qual componentes se comunicam com o intuito de realizar modificações em pontos determinados do serviço alvo. Esse tipo de meta serviço e sua relação com serviço alvo é ilustrado na Figura 4.

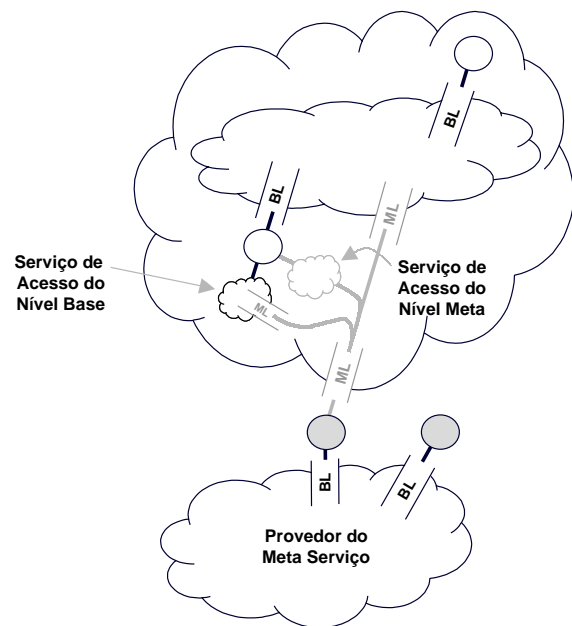


Figura 4: Meta serviço.

Para que um meta serviço possa atuar sobre um serviço alvo, esse alvo deve apresentar uma *implementação aberta*. Implementações abertas expõem detalhes internos do provedor de uma maneira organizada e controlada. O conceito de serviço com implementação aberta é uma adaptação dos conceitos apresentados em [4, 28]. Um serviço com uma implementação aberta fornece (pelo menos) duas interfaces a seus usuários: (i) uma interface do *nível base* (*Base Level Interface — BL Interface*), que permite o acesso às facilidades que, em última instância, constituem o próprio objetivo final do serviço alvo, e (ii) uma interface de *nível meta* (*Meta Level Interface — ML Interface*), que revela aspectos da implementação do serviço alvo e permite a adaptação do comportamento do serviço obtido na interface do nível base por meio de modificações nesses aspectos. O acesso aos componentes de implementação é realizado por intermédio de um provedor de acesso especial denominado *provedor de serviço de acesso do nível meta* (*Meta-Level Access Service provider — MLAS provider*).

Ao se comunicarem através do provedor do meta serviço, usuários se utilizam da interface do nível base desse provedor, de forma que o meta serviço e seus usuários formam um ambiente de oferecimento de serviços como outro qualquer. Dessa forma, meta serviços podem ser o serviço alvo de outros meta serviços, constituindo o que se poderia chamar de “meta serviços de meta serviços” ou *torres de serviços e meta serviços*. Em arquiteturas com essa característica, o “topo da torre” será denominado o *serviço principal*. Como exemplo de uma torre de meta serviços, considere o serviço principal (plano do usuário), a sinalização e a meta sinalização em uma rede ATM.

#### 4.1 TOPOLOGIAS VIRTUAIS

Os componentes de um serviço e suas associações (pipes) definem um grafo, denominado *topologia virtual*. Baseando-se nesse conceito, define-se a principal noção sobre a adaptação e criação de serviços:

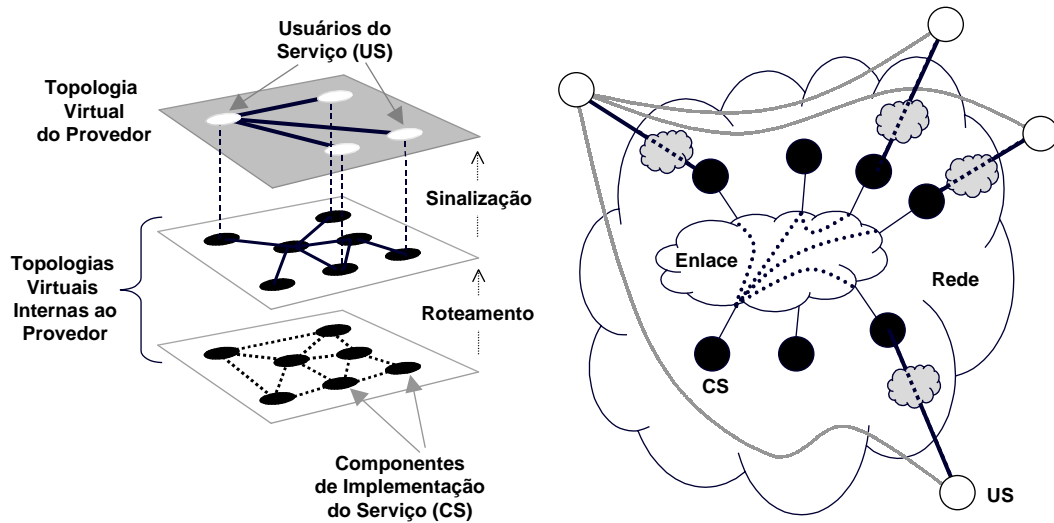
*A adaptação ou criação de serviços é, na realidade, uma questão de adaptação ou criação de topologias virtuais, que envolve a criação (ou modificação ou destruição) de componentes e de associações entre eles na base de informações distribuída.*

Topologias virtuais podem ser criadas “do nada” ou a partir de outras já existentes. Dependendo do grau de adaptabilidade do ambiente, a criação de topologias sem a utilização de outra como base pode ser limitada às fases de construção e implantação, que acontecem antes da operação dos serviços. Nesses casos a prestadora de serviços é em geral a responsável por essa criação, empregando procedimentos que podem variar da instalação manual de componentes de hardware e software até a utilização de ferramentas de gerenciamento e configuração.

A criação de novas topologias a partir de outras existentes pode envolver a criação, modificação ou remoção de componentes de serviço no provedor, e a criação, modificação ou remoção de associações entre os componentes na infra-estrutura. A natureza do serviço e dos mecanismos de adaptação e criação pode limitar essas operações.

Vários mecanismos de criação de topologias podem existir simultaneamente para um mesmo provedor. A forma com que eles influenciam uns nos outros pode ser a mais variada, desde a total dependência, quando um mecanismo age diretamente sobre a topologia resultante do outro, até a total independência, quando topologias completamente separadas e não relacionadas são geradas. Como exemplo, a Figura 5 poderia simbolizar a construção da topologia relacionada ao serviço de uma camada de comunicação do nível de rede, representada pelo provedor de serviços, utilizando-se da infra-estrutura do serviço do nível de enlace, representado pela infra-estrutura do serviço.





**Figura 5:** Exemplo da criação de topologias virtuais.

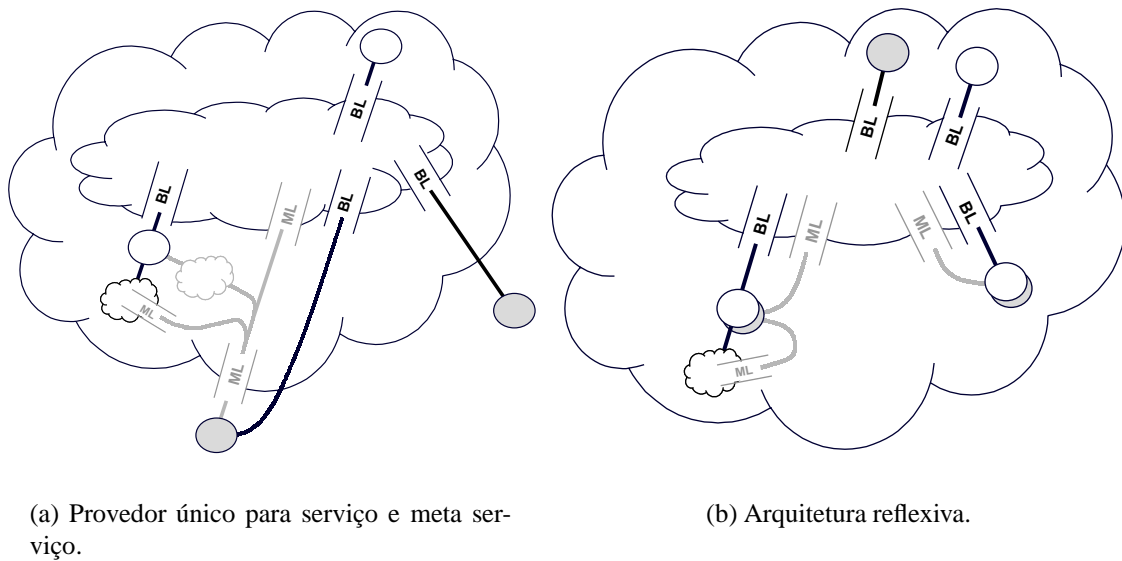
No plano mais abaixo da figura, encontram-se representados os componentes de serviço, que, no exemplo, correspondem a entidades de protocolo do nível de rede, configurados previamente pelo operador em cada nó de rede e ligados pela topologia dada pela camada de enlace. O segundo plano, logo acima, pode representar, por exemplo, o resultado da utilização de um mecanismo qualquer de provisão de informações sobre as rotas utilizáveis (que podem ser fornecidas através de qualquer mecanismo, desde configuração manual até a utilização de protocolos de roteamento). Dessa forma, protocolos de roteamento também podem ser vistos como mecanismos de adaptação de serviços. O terceiro plano corresponde à uma topologia virtual diretamente utilizável pelos usuários, podendo ser, por exemplo, o resultado do mecanismo de sinalização para o estabelecimento de conexões.

## 4.2 INTRODUZINDO REFLEXIVIDADE

Na Figura 4, o meta serviço representado atuava sobre um outro serviço que, de certa maneira, pode ser considerado independente do primeiro. No caso de se estar representando um mecanismo de sinalização, por exemplo, essa poderia ser a modelagem adequada de um sistema de sinalização *out-of-band*.

Um outro caso possível de modelagem surge quando os componentes do meta serviço se comunicam através do mesmo provedor dos usuários do serviço alvo, como ilustrado na Figura 6(a). Esse tipo de arquitetura poderia ser a escolha para a modelagem de mecanismos de sinalização que se utilizam de um mesmo sistema de transporte para informações do usuário e de controle (sinalização). É importante notar, porém, que dependendo da forma de atuação do mecanismo de sinalização, tal modelagem levará a uma situação equivalente à anterior (com os provedores separados), encarada sob um ponto de vista diferente. No caso, por exemplo, em que a topologia virtual utilizada para sinalização for completamente independente daquela utilizada pelos usuários do serviço principal, pode-se considerar que, logicamente, existem dois provedores separados: um para sinalização e outro para o serviço principal.

A configuração apresentada na Figura 6(a) ainda permite uma outra forma de utilização, quando se considera o fato que, ao unificar os provedores do meta serviço e do serviço alvo, a comunicação entre os usuários desses dois níveis se torna possível. Se essa comunicação se efetivar, a arquitetura exibirá uma capacidade denominada de *reflexão* ou *reflexividade*. A



**Figura 6:** Arquiteturas com reflexividade.

capacidade de reflexão originada pelo compartilhamento dos provedores do serviço alvo e do meta serviço, aliada a um outro tipo de unificação, dará origem a uma arquitetura denominada arquitetura *reflexiva*. Em uma arquitetura reflexiva, além dos provedores do serviço alvo e do meta serviço serem um único provedor comum, os próprios usuários desses serviços também podem se fundir, conforme indicado na Figura 6(b). Como exemplo, no caso da sinalização, os mesmos componentes seriam os responsáveis por tratar da comunicação de informações de usuário e de controle. Esse tipo de sistema de sinalização é frequentemente denominado de *in-band*, e é utilizado em protocolos como X.25.

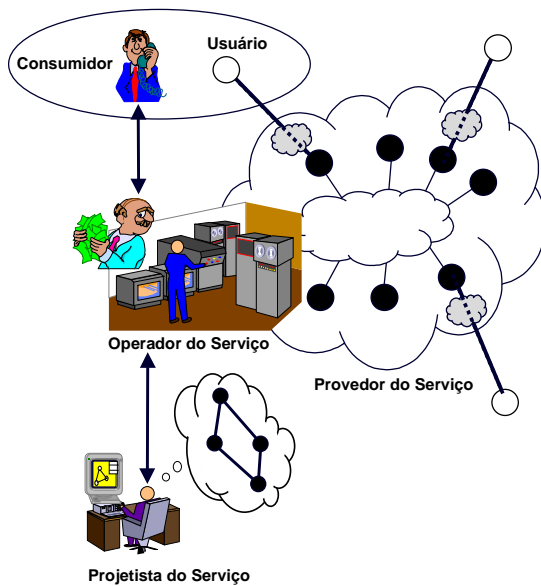
## 5 CLASSIFICAÇÃO DAS FORMAS DE ADAPTABILIDADE SEGUNDO O MODELO

A partir do modelo proposto e as unidades de informação que o compõem, torna-se possível comparar as formas de adaptação com relação aos tipos de elementos manipulados, o momento ou fase em que tais manipulações podem ocorrer, e os responsáveis por tais manipulações. Para tanto, é necessário distinguir que fases estão, em geral, presentes no ciclo de vida de um serviço e o impacto da adaptabilidade sobre esse ciclo. Dessa análise, será possível obter subsídios para classificar alguns mecanismos de adaptação e programação presentes em trabalhos encontrados na literatura.

### 5.1 CICLO DE VIDA

O *ciclo de vida de um serviço* (*Service Life Cycle — SLC*) é um modelo que inclui todas as fases de um serviço, desde de sua concepção inicial, passando por todo o processo de implantação e utilização até o momento em que ele é retirado do sistema. Um cenário genérico de oferecimento de serviços, baseado em uma simplificação do modelo apresentado em [12], encontra-se ilustrado na Figura 7(a).

*Usuários* e *provedor* de serviço são entidades relacionadas aos aspectos técnicos do serviço, correspondendo aos conceitos usuais utilizados em arquiteturas como o OSI-RM. Mais voltados a aspectos administrativos, *consumidores* representam as entidades ou organizações que são responsáveis pela *contratação* (ou *assinatura*) do serviço junto às *operadoras* (também



(a) Personagens.

Fases		Responsáveis
Construção	Análise	Projetista
	Projeto	
	Implementação	
	Testes	
Implantação	Instalação	Operador
	Ativação	
Operação	Assinatura	Consumidor e Operador
	Acesso	Usuário e Provedor
	Interação	
	Saída	
	Cancelamento	Consumidor e Operador
Retirada	Desativação	Operador
	Remoção	

(b) Exemplo das fases de um SLC.

**Figura 7:** Cenário de oferecimento de serviços.

conhecidas como *prestadoras*) de serviços. Essas, por sua vez, são as responsáveis por toda a administração do serviço, incluindo sua instalação, operação e manutenção, sendo, portanto, as representantes do serviço perante consumidores. *Projetista* é uma designação genérica atribuída à entidade ou organização responsável pela construção do serviço. As fases de um SLC tradicional e as responsabilidades de projetistas, operadores, consumidores, usuários e provedores podem ser visualizadas na Figura 7(b).

Com a introdução de programabilidade e adaptabilidade em redes, as responsabilidades começam a se alterar. A separação das fases do SLC torna-se menos óbvia dado que a construção e alteração dos serviços pode ser feita de forma dinâmica. Uma das características desses ambientes é justamente o fato de que consumidores assumem, ainda que parcialmente, as atividades de operadores e projetistas. É claro que não serão os consumidores finais das aplicações que deverão assumir essas responsabilidades, mas sim desenvolvedores, que serão os responsáveis por construir os aplicações usuárias do sistema de comunicação. Nesse sentido, esses desenvolvedores são também parte da entidade a que convencionou-se chamar de consumidor.

Os consumidores dos serviços passam então, de certa forma, a participantes do processo de evolução da infra-estrutura dos serviços de comunicação. A complexidade desse processo poderá se tornar um empecilho se ele não estiver apoiado em um modelo adequado, que integre as duas visões (programação de aplicações e implementação de serviços) [8].

## 5.2 UNIFICANDO AS ESTRATÉGIAS DE ADAPTAÇÃO DE SERVIÇOS

A partir do ciclo de vida definido na Seção 5.1, e dos elementos do modelo proposto na Seção 3, escolheu-se oito exemplos de estratégia para serem classificadas segundo três parâmetros: o momento em que adaptações são realizadas, os atores dessas adaptações e os seus alvos. Essas estratégias, representadas na Figura 8, correspondem a mecanismos presentes nos mais diferentes tipos de redes e serviços. Nos parágrafos que se seguem, encontram-se resumidas

algumas características e interpretações de cada estratégia.

	Momento da Adaptação						Atores da Adaptação					Alvos da Adaptação			
	Construção	Implantação ou Retirada	Subscrição ou Cancelamento	Antes ou Após Comunicação	Durante a Comunicação	Projetista	Operador	Consumidor	Usuário	Terceiros	Meta Serviços	Associações	Informações de Estado	Instâncias de Componentes	Classes de Componentes
Estratégia 1	✓					✓					✓	✓	✓	✓	
Estratégia 2		✓					✓				✓	✓	✓	✓	
Estratégia 3			✓				✓	✓			✓			✓	
Estratégia 4				✓					✓		✓	✓	✓	✓	
Estratégia 5				✓	✓					✓	✓	✓	✓	✓	
Estratégia 6				✓	✓					✓		✓			
Estratégia 7				✓		✓			✓		✓	✓	✓	✓	✓
Estratégia 8					✓				✓		✓	✓	✓	✓	✓

**Figura 8:** Estratégias de adaptação de serviços.

A estratégia 1 corresponde aos mecanismos normalmente utilizados na construção tradicional de serviços. Projetistas estabelecem equipamentos, enlaces físicos, protocolos básicos, etc., tudo feito em tempo de construção do serviço. A estratégia 2 pode ser interpretada como o tipo de mecanismo utilizado pelos operadores para configurar a rede em termos dos protocolos e outras informações como conexões permanentes (associações). A estratégia 3 corresponde as ações de adaptação efetuadas pelos operadores em concordância com o acordado ou solicitado por consumidores. A criação de enlaces e serviços específicos, por exemplo, são criados por este tipo estratégia. A estratégia 4 é simplesmente a representação de um mecanismo de sinalização, no qual os próprio usuários adaptam os serviços (no caso de arquiteturas reflexivas), ou isto é feito através de um meta serviço separado. Na estratégia 5, outros usuários (terceiros) podem adaptar no estado e na configuração de associações e entidades de protocolos, sendo essas adaptações possíveis tanto durante os períodos de comunicação como fora deles. Mecanismos de Gerência agem, basicamente dessa forma. Mecanismos utilizados por protocolos para estabelecimento de rotas podem ser vistos como instâncias da estratégia 6. Um meta serviço pode ser o responsável por tais adaptações. A estratégia 7 representa a capacidade de projetistas, operadores, usuários ou meta serviços podem ter de adaptar os serviços, incluindo a capacidade de adicionar novos protocolos à rede (representada pela última coluna da tabela). A possibilidade de que essas adaptações possam ser feitas durante a operação do serviço (apesar de ser antes da comunicação propriamente dita) revela tratar-se de mecanismos relacionados a redes programáveis. Por fim, a estratégia 8 é representativa dos mecanismos presentes em redes ativas, nos quais as adaptações podem ser feitas (de forma reflexiva) pelos usuários, durante a própria comunicação e incluem a adição de novos componentes de protocolo.

Essa lista de estratégias não abrange todas as possibilidades e é bastante genérica em relação as especificidades de cada mecanismo. Ela serve, porém como ponto de partida para a avaliação da função que cada mecanismo pode ter em uma rede e auxiliar no levantamento de que mecanismos podem ser necessários em cada caso ou necessidade.

## 6 UTILIZAÇÃO DE FRAMEWORKS

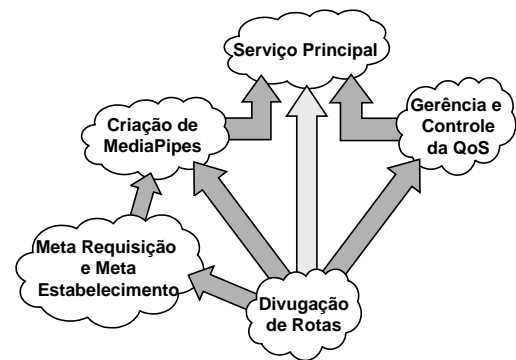
A idéia de se utilizar frameworks para a modelagem de certos aspectos de serviços de comunicação nasce da percepção de que as estruturas e mecanismos relativos à provisão de determina-

das funções são recorrentes nas várias escalas de distribuição e níveis de abstração presentes em um ambiente de oferecimento de serviços qualquer, e independentes da diversidade de serviços passíveis de serem oferecidos pelo seu provedor de serviços. A utilização de frameworks não só permite aumentar o reuso dos produtos gerados nas fases que precedem a operação, como também se prestam à definição dos pontos que podem ser dinamicamente adaptados dentro de um serviço. A experiência com a modelagem de dois frameworks, um que trata dos mecanismos de provisão de QoS [14] e outro que trata dos mecanismos de multicast [29], revelou alguns aspectos peculiares, tanto da modelagem de frameworks relacionados à comunicação, como da forma de implementação dos hot-spots nesse tipo de serviço. Nessa seção, apresenta-se, como exemplo, um resumo das principais características do framework de QoS.

## 6.1 FRAMEWORK DE QoS

No que se refere à provisão de QoS em um ambiente de oferecimento de serviços, duas fases principais podem ser identificadas: (i) *criação de MediaPipes* e (ii) *gerência da QoS*. Os mecanismos presentes nessas fases são os alvos da modelagem do framework de QoS.

A criação de MediaPipes, a gerência da QoS e o serviço principal podem ser vistos como uma torre de meta serviços, estabelecendo e controlando o funcionamento de topologias virtuais, conforme ilustrado na Figura 9. A divulgação de rotas é responsável por tornar disponíveis as informações que permitirão a escolha dos provedores que participarão na composição dos contratos requisitados. Esses protocolos servem de meta serviços para os demais serviços. Em particular, em redes não configuráveis com relação à QoS, esse meta serviço atuará diretamente sobre o serviço principal, seguindo a seta mais clara da Figura 9. Caso contrário, o meta serviço de distribuição de informação de roteamento atuará apenas sobre os demais serviços.



**Figura 9:** Torre de meta serviços para o tratamento da QoS.

A criação de um MediaPipe envolve a criação de MediaPipes nos provedores internos (de acesso e de infra-estrutura) adequados. Esse processo, denominado de *negociação*, começa quando o mecanismo de *controle de admissão*, ao ser informado sobre uma nova requisição de criação<sup>2</sup>, aciona um conjunto de mecanismos cuja função é identificar os provedores internos envolvidos e atribuir a cada um deles uma parcela da responsabilidade pela provisão da QoS (através de um novo controle de admissão do provedor interno). Dentre esses mecanismos, os de *roteamento*, *mapeamento* e *sintonização* desempenham papéis centrais.

O mecanismo de *roteamento* é o responsável pela escolha dos provedores internos adequados utilizando como base as informações de alcançabilidade dos componentes envolvidos. Tais informações serão fornecidas por mecanismos relacionados aos frameworks definidos em [29].

A *sintonização* é responsável por definir, quantitativamente, as parcelas de responsabilidade de cada provedor interno escolhido. Como diferentes provedores participam desse processo, cada qual possivelmente associado a uma visão de QoS distinta (conforme mencionado em [14]), torna-se necessário o *mapeamento* das requisições para a criação de MediaPipes nesses

<sup>2</sup>Uma requisição de criação de um MediaPipe inclui, entre outras informações, parâmetros que caracterizam a carga a ser gerada no MediaPipe e que especificam a QoS desejada pelos usuários, como descrito em [14].

diferentes provedores. Todo o processo se repete até chegar-se ao nível em que o mecanismo de controle de admissão pode efetuar testes diretamente sobre um recurso sem repassá-los a nenhum provedor interno.

Após o MediaPipe ser criado, o provedor deve garantir, durante todo o tempo de uso do MediaPipe, a manutenção da QoS negociada. Dentre os mecanismos responsáveis pela gerência da QoS, estão incluídos os de *monitorização* e *ressintonização* da QoS.

Os mecanismos de *monitorização* permitem o registro da carga efetivamente gerada pelos usuários e da QoS realmente oferecida pelo provedor. Funcionalmente, a *ressintonização* é similar à *sintonização*, porém atua de forma tal que reajustes no serviço ocorram sem que o fornecimento do mesmo seja interrompido. Quando um dos MediaPipes internos não puder honrar com sua parcela de responsabilidade pela provisão da QoS fim-a-fim (devido a uma sobrecarga, p. ex.), caberá ao mecanismo de *ressintonização* efetuar operações de ajuste.

A adaptação dos mecanismos de provisão de QoS a serviços e escalas de distribuição específicos é essencialmente guiada segundo a visão de QoS dos usuários e a infra-estrutura existente no ambiente. Todas essas informações podem ser estruturadas através de *parâmetros de caracterização do serviço*. Para tornar a parametrização de serviços em um provedor qualquer suficientemente flexível, definiu-se um esquema de parametrização, independente dos possíveis serviços a serem oferecidos pelo provedor. Este esquema é também modelado pelo framework de QoS. O resultado de sua utilização é, em última análise, a definição de uma base de informações de QoS sobre a qual os diversos mecanismos de provisão de QoS podem atuar.

Em [14], foi apresentado um caso de uso do framework, onde é modelada uma arquitetura para protocolos de configuração de MediaPipes em redes ATM. O esquema de configuração adotado baseou-se na abordagem de intermediação apresentada em [22]. Assim, os negociadores da QoS são responsáveis pela gerência de recursos das estações finais (basicamente, pelo controle de escalonamento de processos), bem como pela comunicação com a interface de gerência de recursos provida pela SAAL. Um protótipo dessa arquitetura foi implementado para a configuração específica de fluxos de quadros de vídeo MPEG-2.

## 7 CONCLUSÕES

Este trabalho apresentou um modelo genérico para a construção de serviços e arquiteturas de comunicação. O modelo proposto foi construído tendo em mente que o funcionamento de um sistema de comunicação pode ser totalmente regulado sobre uma *base de informações distribuída* que controla e auxilia o funcionamento de todo o sistema. Seus principais aspectos estão relacionados à modelagem dos elementos básicos que podem ser encarados como unidades de informação e configurados para a construção dinâmica de serviços. Um dos pontos levantados foi a utilização de *frameworks* na modelagem de serviços de comunicação. Procurou-se formar, por meio da definição dos elementos básicos e das restrições para sua configuração, uma espécie de “linguagem básica” para construção de serviços, a partir da qual estruturas pré-montadas poderiam ser definidas para modelar necessidades comuns em serviços de comunicação (como a provisão da QoS e a comunicação de grupo).

Uma das contribuições que pode ser ressaltada é o estabelecimento de uma relação mais estreita entre vários assuntos hoje tratados na área engenharia de software com os da nova disciplina de engenharia de serviços de telecomunicações. Em especial, enfatizou-se o aspecto de como se pode fazer uso de técnicas de orientação a objetos e frameworks e estilos arquiteturais para a construção de serviços adaptáveis. A importância e o significado do tratamento de informações em arquiteturas de comunicação e o impacto causado nos modelos de serviço foram

discutidos e, ao longo dessa discussão, alguns trabalhos relacionados à modelagem e à adaptabilidade desses serviços foram comentados. As diversas fases que, em geral, estão presentes no ciclo de vida de um serviço foram apresentadas e o impacto da adaptabilidade sobre esse ciclo foi estudado. Dessa análise, foi possível obter subsídios para classificar alguns mecanismos de adaptação e programação presentes em trabalhos encontrados na literatura. A partir da definição das unidades de informação, procurou-se comparar essas formas de adaptação com relação aos tipos de elementos manipulados, o momento ou fase em que tais manipulações podem ocorrer, e os responsáveis por tais manipulações. Os conceitos estabelecidos permitiram analisar alguns mecanismos comuns, presentes em redes tradicionais, bem como diferenciar de que forma as novas arquiteturas que tratam do problema da adaptabilidade se relacionam com esses mecanismos.

## REFERÊNCIAS

- [1] M. B. Abbott e L. L. Peterson. A language-based approach to protocol implementation. *IEEE/ACM Transactions on Networking*, 1(1):4–19, fevereiro 1993.
- [2] C. Aurrecoechea, A. Campbell e L. Hauw. A survey of QoS architectures. *ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture*, 6(3):138–151, maio 1998.
- [3] H. Berndt, P. Graubmann e M. Wakano. Service specification concepts in TINA-C. Em *International Conference on Intelligence in Broadband Services and Networks*, 1994.
- [4] G. S. Blair e G. Coulson. The case for reflective middleware. Relatório Técnico MPG-98-38, Distributed Multimedia Research Group, 1998.
- [5] G. S. Blair e J. B. Stefani. *Open Distributed Processing and Multimedia*. Addison Wesley Longman Ltd., 1997.
- [6] A. T. Campbell, M. E. Kounavis, H. D. Meer, K. Miki e J. Vicente. A survey of programmable networks. *ACM Computer Communications Review*, abril 1999.
- [7] S. Colcher. *Um Meta Modelo Para Aplicações e Serviços de Comunicação Adaptáveis Com Qualidade de Serviço*. Tese de Doutorado, Departamento de Informática — PUC-Rio, novembro 1999.
- [8] S. Colcher e L. F. G. Soares. Modelo de referência unificado para arquitetura de protocolos e programação de aplicações multimídia. Em *XVI Simpósio Brasileiro de Redes de Computadores (SBRC'98)*, editado por J. Leite e E. S. Silva, págs. 631–650, Rio de Janeiro, RJ, maio 1998.
- [9] F. M. Costa, G. S. Blair e G. Coulson. Experiments with reflective middleware. Em *Proceedings of the ECOOP'98 Workshop on Reflective Object Oriented Programming and Systems*. Springer Verlag, 1998.
- [10] G. Coulson e D. G. Waddington. A CORBA compliant real-time multimedia platform for broadband networks. Em *Proceedings of TRENDS'96*, Germany, setembro 1996.
- [11] P. Druschel. *Operating System Support for High Speed Networking*. Tese de Doutorado, Department of Computer Science — The University of Arizona, Tucson, agosto 1994.
- [12] EUROSCOM Project P103 — Evolution of the Intelligent Network. *Deliverable No 6a: Framework for Service Description with Supporting Architecture*, fevereiro 1995.
- [13] E. Gamma, R. Helm, R. Johnson e J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [14] A. T. A. Gomes, S. Colcher e L. F. G. Soares. Um framework para provisão de QoS em ambientes genéricos de processamento e comunicação. Em *XVII Simpósio Brasileiro de Redes de Computadores (SBRC'99)*, págs. 307–322, Salvador, BA, Brasil, junho 1999.
- [15] J. Harju, B. Silverajan e I. Toivanen. Experiences in telecommunications protocols with an OO based implementation framework. Em *Proceedings of the ECOOP'97 Workshops*, Jyväskylä, Finland, junho 1997. Simon Znaty, Jean-Pierre Hubaux, Springer, Lecture Notes In Computer Science, LNCS 1357.

- [16] N. C. Hutchinson e L. L. Peterson. The *x*-Kernel: An architecture for implementing network protocols. *IEEE Transactions on Software Engineering*, 17(1):64–76, janeiro 1991.
- [17] ISO/IEC JTC1/SC21/WG7. *ISO/IEC DIS 10746-1/ITU-T X.901 — Reference Model of Open Distributed Processing, Part 1: Overview*, maio 1995. output from the editing meeting in Helsink (Finland).
- [18] ITU-T Recommendation I.312/Q.1201. *Principles of Intelligent Network Architecture*, outubro 1992.
- [19] A. A. Lazar. Programming telecommunication networks. *IEEE Network*, págs. 8–18, setembro 1997.
- [20] A. A. Lazar, K. S. Lim e F. Marconcini. Binding model: Motivation and description. Relatório Técnico CTR 411-95-17, Comet Group, Department of Electrical Engineering and Center for Telecommunications Research, Columbia University, New York, setembro 1995. <http://www.ctr.columbia.edu/comet/xbind/xbind.html>.
- [21] G. Meszaros. Design patterns in telecommunications systems architecture. *IEEE Communications Magazine*, 37(4):40–45, abril 1999.
- [22] K. Nahrstedt e J. M. Smith. End-to-end QoS guarantees: Lessons learned from OMEGA. Relatório Técnico UIUCDCS-R-97-1990, UILU-ENG-97-1704, MONET (Multimedia Operating Systems and Networking Group) – University of Illinois, maio 1996.
- [23] P. Nikander, J. Parssinen, B. Sahlin e K. Hoglund. A java based framework for cryptographic protocols. Relatório técnico, Telecommunications Software and Multimedia, Helsinki University of Technology, maio 1997. Draft Research Report.
- [24] Object Management Group (OMG). *The Common Object Request Broker: Architecture and Specification*, fevereiro 1998. Revision 2.2.
- [25] K. S. Perumalla, R. M. Fujimoto e A. T. Ogielski. MetaTeD — a meta language for modeling telecommunications networks. Relatório Técnico GIT-CC-96-32, College of Computing — Georgia Institute of Technology, Atlanta, GA 30332, 1996.
- [26] W. Pree. *Framework Patterns*. SIGS Management Briefings. SIGS Books & Multimedia, 1996.
- [27] K. Psounis. Active networks: Applications, security, safety, and architectures. *IEEE Communications Surveys*, (First Quarter):2–16, 1999.
- [28] R. Rao. Implementation reflection in silica. Em *Proceedings of ECOOP'91*, editado por P. America, págs. 251–267. Lecture Notes in Computer Science, Springer-Verlag, 1991.
- [29] M. A. A. Rodrigues, S. Colcher e L. F. G. Soares. Um framework para a provisão de serviço de multicast em ambientes genéricos de comunicação de dados. Em *XVII Simpósio Brasileiro de Redes de Computadores (SBRC'99)*, págs. 206–221, Salvador, BA, Brasil, junho 1999.
- [30] D. C. Schmidt. The ADAPTIVE communication environment: An Object-Oriented network programming toolkit for developing communication software. Em *Proceedings of the 12th Annual Sun Users Group Conference*, págs. 214–225, San Jose, CA, dezembro 1993. SUG.
- [31] D. C. Schmidt. Applying design patterns and frameworks to develop Object-Oriented communication software. Em *Handbook of Programming Languages*, editado por P. Salus, volume I. MacMillan Computer Publishing, 1997.
- [32] D. C. Schmidt, A. Gokhale, T. H. Harrison e G. Parulkar. A High-Performance endsystem architecture for Real-Time CORBA. *IEEE Communications Magazine*, 14(2), fevereiro 1997.
- [33] P. Steyaert. *Open Design of Object-Oriented Languages, A Foundation for Specialisable Reflective Language Frameworks*. Tese de Doutorado, Vrije Universiteit Brussel, 1994.
- [34] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall e G. Minden. A survey of active network research. *IEEE Communications*, janeiro 1997.
- [35] Y. Yemini e S. D. Silva. Towards programmable networks. Em *Proceedings of the IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, L'Aquila, Italy, outubro 1996.
- [36] S. Znaty e J. Hubaux. Telecommunications services engineering: Principles, architectures and tools. Em *Proceedings of the ECOOP'97 Workshops*, Jyväskylä, Finland, junho 1997. Simon Znaty, Jean-Pierre Hubaux, Springer, Lecture Notes In Computer Science, LNCS 1357.