Um framework baseado em componentes para o domínio de adaptação de conteúdo na Internet

Renato A. T. Claudino[1], Wanderley L. de Souza, Antonio F. do Prado

Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
Caixa Postal 676 – 13565-905 - São Carlos, SP

{renato, desouza, prado}@dc.ufscar.br

Abstract. *In the last few years it has become more common to have small mobile devices accessing the Internet by means of wireless access networks. In this new context one major challenge is the content adaptation to these devices. These adaptations aim to satisfy the devices' capabilities and users' preferences, and to optimize the use of the wireless access networks. In this sense this paper presents a framework, based on the reuse of components, for the development of applications that deal with Internet content adaptations.*

Keywords: content adaptation, components reuse; adaptation policy.

Resumo. *Nos últimos anos tornou-se mais comum o acesso de pequenos dispositivos móveis a Internet via redes de acesso sem fio. Neste novo contexto um grande desafio é a adaptação de conteúdo para tais dispositivos. Essas adaptações buscam satisfazer as capacidades dos dispositivos, as preferências dos usuários e otimizar o uso das redes de acesso sem fio. Neste sentido este artigo apresenta um framework, baseado no reuso de componentes, para o desenvolvimento de aplicações que lidam com adaptações de conteúdo na Internet.*

Palavras-chave: adaptação de conteúdo; reuso de componentes; política de adaptação.

1. Introdução

O termo adaptação de conteúdo refere-se à modificação da representação de objetos ou conteúdos da Internet, buscando encontrar uma versão ideal desses conteúdos [Buchholz 2003]. Os trabalhos nessa área de adaptação cresceram bastante nos últimos anos, visto que a quantidade de conteúdo disponível na Internet aumentou significativamente.

Essa adaptação pode ser realizada sobre diferentes tipos de conteúdos, assim como: **páginas de marcação**, e.g. HTML (*HyperText Markup Language*), XML (*eXtensible Markup Language*) ou WML (*Wireless Markup Language*); **imagens**, e.g. JPEG (*Joint Photographic Experts Group*), GIF (*Graphic Image Format*) ou WBMP

[1] Bolsista CAPES

(*Wireless BitMaP*); **vídeo sob-demanda e arquivos de vídeo**, e.g. MPEG (*Moving Picture Experts Group*), AVI (*Audio Video Interleave*) ou Quicktime; **aplicações** (arquivos executáveis); ou **e-mail**;

A preocupação com a adaptação de diferentes conteúdos foi reforçada devido à disseminação do acesso à Internet. Com a difusão dos computadores pessoais, os conteúdos da Internet passaram a ser acessados por milhares de usuários. Estes geralmente possuem diferenças em relação a: classe social e financeira, profissão, idade, localização geográfica, idioma e preferências pessoais, entre outros. Nesse ambiente heterogêneo, as adaptações visam obter um maior nível de satisfação de cada usuário, atendendo seus interesses pessoais e suas necessidades.

Outro fator que incentivou a realização dessas adaptações foi o desenvolvimento de novas tecnologias de redes de acesso à Internet. Algumas destas possuem grande largura de banda (e.g. conexão a cabo, *Asymmetric Digital Subscriber Line* – ADSL e conexão a rádio), outras possuem baixa largura de banda (e.g. via linha telefônica comum), além das tecnologias com largura de banda variável (e.g. rede local sem fio, chamada de *Wireless Local Access Network* ou *WLAN*).

O desenvolvimento de variados dispositivos com acesso à Internet, como PDAs (*Personal Digital Assistent*), telefones celulares, *palmtop* e *laptop*, consolidou a necessidade de adaptar os conteúdos da Internet. Esses dispositivos geralmente possuem características próprias como tamanho de tela, memória, processamento e aplicativos.

Partindo desse contexto, os trabalhos nessa área buscam desenvolver uma estrutura que viabilize a realização dos serviços de adaptação. Buscando uma solução, foram criadas as redes de serviços que constituem uma nova camada na infra-estrutura da Internet. Essas redes descrevem mecanismos para interceptar a entrega de conteúdo e executar sobre este os serviços de adaptação. As redes de serviços são principalmente representadas pelo modelo OPES (*Open Pluggable Edge Services*) [Tommlinson 2001].

Um aspecto essencial na realização desses serviços é a definição de uma política de adaptação, que decide qual a melhor adaptação a ser realizada sobre o conteúdo, quando e para quem solicitá-la. Caso seja necessário realizar mais de uma adaptação, essa política define ainda a ordem de execução das adaptações. Entretanto para desempenhar essa política, devem ser consideradas informações sobre o usuário, o dispositivo do usuário, o conteúdo, a rede de acesso e o contrato de serviços.

Assim, motivados em pesquisar soluções para os problemas de adaptação, este artigo apresenta um framework baseado em componentes que fornece uma infra-estrutura flexível para o desenvolvimento de aplicações que prestam serviços de adaptação de conteúdo. O framework define uma política geral de adaptação que pode ser empregada, através da reutilização de componentes de software, pelas diferentes aplicações do domínio de adaptação de conteúdo na Internet.

A seqüência deste artigo está estruturada como segue: a seção 2 discute a adaptação de conteúdo na Internet, suas principais aplicações e a política de adaptação; na seção 3 é apresentado o framework proposto; a seção 4 descreve uma experiência de reuso do framework, considerando uma política de adaptação, os protocolos abordados, além de mostrar uma avaliação de desempenho dessa reutilização; na seção 5 são discutidos os trabalhos correlatos; e finalmente na seção 6 apresentam-se as conclusões deste artigo.

2. Adaptação de Conteúdo na Internet

No final do século passado houve um grande crescimento no número de usuários da Internet, provocando o aumento do fluxo de dados transmitidos nessa rede. Esse cenário implicou no congestionamento da Internet e no aparecimento de seus quatro pontos críticos [Akamai 2000]. Para minimizar esse problema desenvolveram-se as **redes de conteúdo**, que aproximam o conteúdo ao usuário através dos dispositivos de borda, evitando assim os gargalos da rede entre o usuário e o servidor de conteúdo [Vakali 2003].

No entanto, com a grande diversidade dos quatro elementos da Internet (conteúdo, usuário, redes de acesso e dispositivo), gerou-se a necessidade de adaptar "*on the fly"* os diversos conteúdos da Internet, visando satisfazer aos usuários, atender às capacidades de seus dispositivos e aperfeiçoar o uso dos recursos das redes de acesso. Dessa maneira foram criadas as **redes de serviços**, nas quais a função de adaptar os conteúdos é distribuída a servidores remotos, os *módulos de serviços de conteúdo*. As redes de serviços oferecem um mecanismo de direcionamento do conteúdo a esses módulos, que modificam o conteúdo segundo as adaptações solicitadas.

2.1. Aplicações de Adaptação de Conteúdo

Dentre as aplicações de adaptação de conteúdo na Internet [Beck 2000 e McHenry 2001], que determinaram os requisitos do domínio do framework proposto, destacam-se:

Escaneamento de vírus: capacidade de desempenhar dinamicamente a busca por vírus sobre o conteúdo requisitado antes de entregá-lo ao usuário. Com esse serviço o usuário acessa os conteúdos da Internet sem receber um conteúdo infectado.

Inserção de propaganda: adiciona informações publicitárias num conteúdo (e.g. páginas Web) baseado nos interesses do usuário ou na sua localização geográfica. Essa adaptação permite, por exemplo, que as propagandas originais das páginas Web sejam substituídas por outras regionais ou focadas no perfil do usuário.

Tradução de linguagens markup: permite aos dispositivos que não suportam páginas HTML, e.g. certos telefones celulares, receberem essas páginas em outro formato (e.g. WML). Esse serviço além de disponibilizar um número maior de conteúdo para dispositivos limitados, alivia o trabalho que o servidor de origem teria para desenvolver duas apresentações de um mesmo conteúdo.

Compressão de dados: permite ao servidor de origem enviar seus conteúdos compactados para que o dispositivo de borda o descompacte, economizando a largura de banda utilizada nessa comunicação.

Filtragem de conteúdo: capacidade de redirecionar uma requisição não autorizada ou bloquear uma resposta também não permitida. Esse serviço permite, por exemplo, aos pais controlarem o acesso de seus filhos a conteúdos impróprios.

Transcodificação de imagens: trata-se da capacidade de manipular os arquivos de imagem, assim como: transformar seu formato; reduzir seu tamanho, resolução e sua escala de cores; ou remover imagens de uma página Web.

Tradução de idiomas: traduz uma página Web de um idioma para outro, por exemplo, de português para chinês. Pode ser útil quando o servidor de origem não contiver todas as traduções necessárias para entregar um mesmo conteúdo em múltiplos idiomas.

2.2. Política de Adaptação

Outro requisito importante a ser atendido pelo framework relaciona-se com a política de adaptação. A política de adaptação define: os serviços de adaptação que serão executados; os adaptadores locais ou remotos que realizarão essas adaptações; quando as adaptações serão solicitadas; e a ordem de suas execuções. Para que essas decisões sejam tomadas com eficiência é necessário conhecer as informações relativas ao ambiente de adaptação, assim como: preferências e dados pessoais dos usuários; características e capacidades de seus dispositivos; características dos conteúdos requisitados; condições da rede de acesso; e contrato de serviços entre o usuário e o provedor de serviços [Souza 2002].

Essas informações podem ser classificadas como dinâmica ou estática. As informações dinâmicas são coletadas do ambiente a cada processo da política de adaptação, pois sofrem alterações periodicamente, assim como as informações da rede de acesso e do conteúdo. As informações estáticas são adicionadas no momento da conexão do usuário e não são obrigatoriamente atualizadas a toda requisição/reposta de conteúdo. As informações estáticas são relativas ao usuário, dispositivo e contrato de serviços, sendo que um meio de disponibilizá-las é através de perfis.

O perfil de **usuário** contém informações pessoais do usuário e suas preferências de adaptação. Com base nesse perfil, o processo de adaptação é direcionado para que seu resultado esteja em conformidade com as predileções do usuário. Uma vez que os usuários podem requerer diferentes serviços para um mesmo conteúdo, adaptações que não se baseiam em suas preferências podem ser inconvenientes ou indesejadas.

No perfil de **dispositivo** estão contidas características e capacidades de hardware (e.g., dimensões de tela e memória) e software (e.g., sistema operacional e *browser*) do dispositivo. O conhecimento das capacidades de um dispositivo é fundamental para o processo de adaptação, que considera as limitações de cada dispositivo.

No perfil de **contrato de serviços** constam os serviços, oferecidos pelo provedor de serviços, que o usuário pode utilizar. O provedor de serviços disponibiliza diversos planos de serviços, buscando atingir as necessidades do usuário. Um serviço pode ser vetado ao usuário, caso esse serviço não esteja contido no perfil de contrato de serviços.

Dentre as informações dinâmicas utilizadas, sobre **conteúdo** pode-se analisar o tipo desse conteúdo, formato e dimensão de uma imagem, idioma de uma página Web, tamanho em *bytes* e origem do conteúdo, entre outros. Essas informações podem ser encontradas no cabeçalho de resposta do protocolo ou nos *metadados* do conteúdo.

Finalmente, têm-se as informações relativas às características da **rede de acesso**, determinando a largura de banda, o tempo de resposta e o tráfego da rede, entre outros. Essas informações são analisadas para otimizar o uso dos recursos dessa rede, por exemplo, evitando que ocorra uma alta taxa de erros nas transmissões de vídeo.

Além disso, como nem todo conteúdo requisitado deve sofrer as adaptações de conteúdo oferecidas pelo provedor de serviços, é necessário um mecanismo que invoque serviços de adaptação somente se certas condições forem satisfeitas. Para guiar esse processo de tomada de decisão, utilizando os perfis e as informações citadas acima, a política de adaptação deve considerar as **regras de adaptação**.

Essas regras indicam as condições que devem ser atendidas para que determinada adaptação seja realizada. Sendo que essas condições devem ser

condizentes ao usuário, dispositivo, contrato de serviços, rede de acesso ou ao conteúdo. O grupo de trabalho OPES WG desenvolveu duas linguagens para especificar as regras de adaptação: IRML (*Intermediary Rule Markup Language*) e P.

IRML foi designada para oferecer uma simples e eficiente maneira de expressar uma política de execução de serviços. Essa linguagem, baseada em XML, foi criada para refletir os interesses dos dois pontos finais de uma transação de conteúdo: servidor de origem e usuário [Hoffman 2001]. Enquanto que a linguagem P é interpretada e suas características são parecidas com Smalltalk e C++. Essa linguagem oferece os seguintes fatores: exatidão, flexibilidade, eficiência, simplicidade, segurança e rigidez [Beck 2003].

Conforme descrito nesta seção existem vários aspectos a serem considerados quando se pretende oferecer os serviços de adaptação de conteúdo. Com o surgimento das redes de serviços, um aspecto importante do problema é o que trata da distribuição dos serviços de adaptação para servidores dedicados, de forma a não sobrecarregar o dispositivo de borda. Com foco voltado para atender os requisitos apresentados foi construído um framework, conforme se apresenta a seguir.

3. Framework para Adaptação de Conteúdo na Internet

A Figura 1 apresenta uma visão geral do framework, denominado FACI (**F**ramework para **A**daptação de **C**onteúdo na **I**nternet), desenvolvido neste trabalho. O framework baseia-se nas idéias das redes de serviços, sendo constituído de dois pacotes principais: *Adaptation Proxy*, que desempenha o papel do dispositivo de borda; e *Adaptation Server*, que desempenha o papel do módulo de serviços de adaptação.

Os atores, usuário da Internet (*User*) e servidor de origem (*Origin Server*), interagem com o *Adaptation Proxy*, através das requisições de conteúdo e suas respostas. O *Adaptation Proxy* oferece recursos para analisar essas requisições e respostas, baseado em uma política de adaptação, e utiliza os serviços disponíveis no *Adaptation Server* para adaptar o conteúdo transmitido.

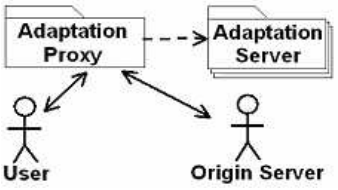

Figura 1. Visão geral do FACI

3.1. *Adaptation Proxy*

A Figura 2 apresenta os componentes do pacote *Adaptation Proxy*. O componente ***Content Transfer Protocol*** é responsável pela comunicação com o usuário e com o servidor de origem, sendo genérico para suportar os protocolos de aplicação utilizados na Internet para transmissão de conteúdo. Dentre os principais protocolos de aplicação citam-se: HTTP (*HyperText Transfer Protocol*), RTP (*Real-time Transport Protocol*), SOAP (*Simple Object Access Protocol*) e FTP (*File Transfer Protocol*).

O *Adaptation Proxy* solicita as adaptações a um *Adaptation Server* através de protocolos que fazem a comunicação entre o dispositivo de borda e o módulo de serviços. Assim, foi definido o componente ***Callout Protocol Client***, que realiza a chamada remota de adaptação. Esse componente suporta diferentes protocolos, assim como os protocolos criados especificamente para essa comunicação: ICAP (*Internet Content Adaptation Protocol*) [Elson 2003] e OCP (*OPES Callout Protocol*) [Rousskov 2005].

Outro componente é o ***Cache***, que foi construído visando melhorar o desempenho do FACI**.** Esse componente armazena temporariamente os conteúdos requisitados da Internet, assim como páginas Web, vídeos ou imagens. Antes de solicitar um conteúdo ao servidor de origem, o *Adaptation Proxy* verifica se o mesmo está em *Cache*. Caso positivo, o FACI evita uma requisição ao servidor de origem e o conteúdo é entregue mais rapidamente ao usuário.

Apesar de utilizar os módulos de serviços de adaptação, o FACI oferece recursos para realizar as adaptações de conteúdo no próprio dispositivo de borda, através do componente ***Local Adapter***. Esse componente utiliza os recursos do próprio dispositivo de borda, portanto a utilização em grande escala do mesmo pode afetar o desempenho do *Adaptation Proxy*, retardando a entrega do conteúdo.

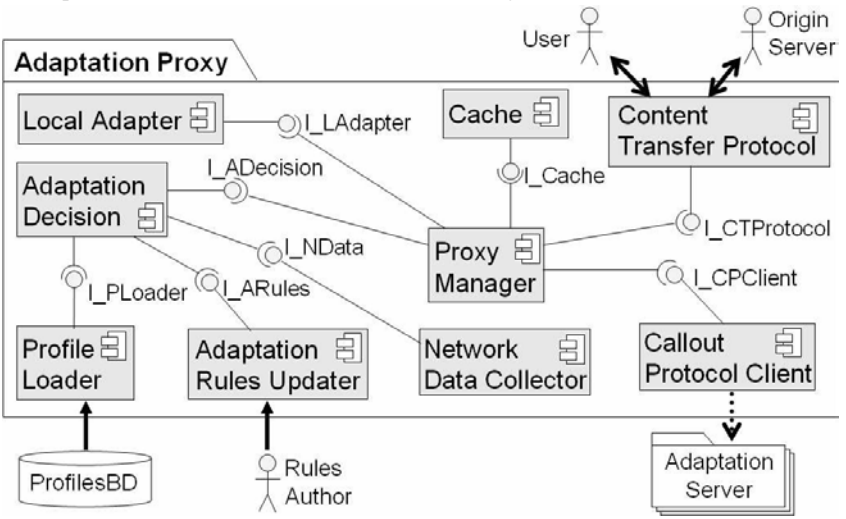

Figura 2. Componentes do *Adaptation Proxy*

O componente ***Proxy Manager*** gerencia o fluxo de dados do *Adaptation Proxy*. Através do *Content Transfer Protocol*, esse componente realiza a troca de mensagens com o usuário e com o servidor de origem. De posse do conteúdo, seja proveniente do usuário ou do servidor de origem, o *Proxy Manager* requisita uma análise de adaptação ao componente *Adaptation Decision*. E, caso alguma adaptação seja necessária, o *Proxy Manager* invoca esse serviço remotamente através do componente *Callout Protocol Client* ou localmente através do componente *Local Adapter*. Além disso, antes da requisição de conteúdo, o *Proxy Manager* verifica se esse conteúdo está presente no *Cache*, visando eliminar requisições desnecessárias.

Um importante requisito da adaptação de conteúdo é a definição da política de adaptação. No FACI, esta é orientada por regras de adaptação e baseada nas informações do usuário, dispositivo, rede de acesso, conteúdo e contrato de serviços. A política de adaptação do FACI é implementada através dos componentes *Adaptation Decision*, *Profile Loader*, *Adaptation Rules Updater* e *Network Data Coletor*.

O componente ***Profile Loader*** busca os perfis de usuário, de dispositivo e de contrato de serviços que estão armazenados na base de dados *ProfilesDB*. Esses perfis viabilizam ao framework satisfazer as preferências e os interesses dos usuários, além de atender às capacidades e limitações dos dispositivos e determinar quais os serviços que o usuário está habilitado a receber. Para inserir os perfis de usuário e de dispositivo na base *ProfilesDB*, deve ser disponibilizada uma interface ao usuário da Internet. Essa interface pode ser uma página Web que contenha um formulário, onde os perfis são preenchidos em campos pré-estabelecidos. Já o perfil de contrato de serviços é inserido pelo *administrador* do sistema, também através de uma interface.

O componente ***Network Data Coletor*** monitora as informações da rede de acesso. A cada alteração dessa rede, esse componente aciona o *Adaptation Decision*, indicando a característica que sofreu modificação e o seu valor atualizado.

O controle da política de adaptação do framework é realizado pelo componente ***Adaptation Decision***. Esse componente recebe as alterações da rede de acesso através do componente *Network Data Coletor* e o conteúdo a ser analisado através do componente *Proxy Manager*. Além disso, esse componente busca os perfis de usuário, dispositivo e de contrato de serviços na base de perfis através do componente *Profile Loader*. De posse dessas informações, o *Adaptation Decision* utiliza as regras de adaptação para decidir quais adaptações deverão ser executadas, qual servidor (seja local ou remoto) que realizará cada adaptação e, caso necessite-se desempenhar múltiplas adaptações, qual a ordem de execução dessas adaptações.

Para assegurar que regras desatualizadas ou inválidas não sejam processadas, o ***Adaptation Rules Updater*** insere, remove e modifica as regras de adaptação implementadas no *Adaptation Decision*. A atualização dessas regras é efetuada pelo ator *Rules Author* através de uma interface oferecida pelo *Adaptation Rules Updater*.

3.2. *Adaptation Server*

Esse pacote do FACI é responsável pelas adaptações de conteúdo solicitadas pelo *Adaptation Proxy*. Considerando que os servidores de adaptação podem realizar diferentes tipos de serviços, cada *Adaptation Server* poderá ter uma estrutura diferente. Assim, baseado nas idéias de componentes de software, foi definida uma estrutura genérica para os diversos *Adaptation Servers*, composta dos componentes apresentados na Figura 3.

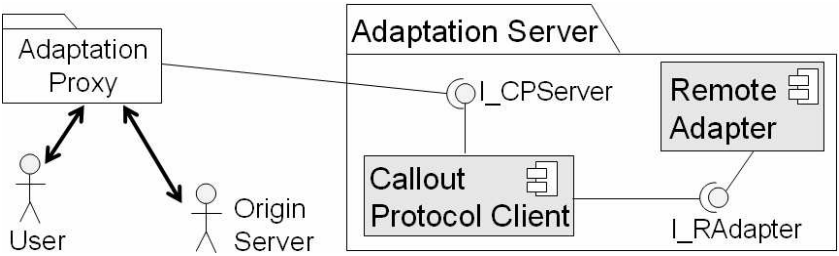

Figura 3. Componentes do *Adaptation Server*

O componente ***Callout Protocol Server*** trata da comunicação com o *Adaptation Proxy*. Esse componente também suporta diferentes protocolos como ICAP e OCP. Esse componente recebe e analisa a solicitação de adaptação do *Callout Protocol Client*,

obtendo a ação a ser realizada e os respectivos parâmetros. A partir desses dados o *Callout Protocol Server* invoca a adaptação solicitada ao componente *Remote Adapter*.

O ***Remote Adapter*** é responsável pela execução das adaptações de conteúdo. A estrutura interna desse componente suporta variações conforme o tipo de adaptação oferecido pelo *Adaptation Server*. Um exemplo de modelagem desse componente pode ser encontrado em [Forte 2004]. O *Remote Adapter* realiza as mesmas funções do *Local Adapter*, porém não utiliza os recursos de processamento do *Adaptation Proxy*. A decisão de adaptar o conteúdo localmente ou remotamente, deve considerar o tempo gasto com o *Callout Protocol* em relação ao tempo gasto com a adaptação do conteúdo. Quanto menor essa relação, mais favorável será a adaptação remota.

3.3. Funcionamento do FACI

Seguindo as definições de [Elson 2003], o FACI foi construído para suportar dois modos de operação: modificação da requisição (*reqmod*) e modificação da resposta (*respmod*). No *reqmod* a adaptação é executada sobre a requisição que o usuário envia ao servidor de origem. Já o *respmod* visa o processamento das respostas de conteúdos provenientes dos servidores espalhados pela Internet antes de entregá-las ao usuário.

Durante o fluxo de requisições e respostas dos conteúdos, existem quatro pontos onde a política de adaptação pode ser processada, conforme mostra a Figura 4 [Babir 2004]. Nos pontos 1 e 2 a política é processada sobre a requisição do usuário, respectivamente, antes ou após a busca do conteúdo em *Cache*. Nos pontos 3 e 4, executa-se a política sobre a resposta do servidor de origem, respectivamente, antes ou após seu armazenamento em *Cache*.

Figura 4. Pontos de processamento da política de adaptação

O componente *Adaptation Decision* exporta em sua interface o método *policy.request*, que tem um parâmetro que indica o ponto de processamento da análise de adaptação. A Figura 5 mostra um modelo de seqüência do caso de uso que realiza uma adaptação da resposta do servidor de origem (*respmod*).

Nesse modelo o *Proxy Manager* solicita ao *Content Transfer Protocol,* através do método *cTP.receiveRequest*, que aguarde uma requisição do usuário (*User*). Quando essa requisição é realizada, o *Proxy Manager* solicita um processamento da política de adaptação (*policy.request*) ao *Adaptation Decision*, enviando os identificadores do usuário, de seu dispositivo e de sua rede de acesso, além de fornecer o conteúdo, que nesse caso é a requisição do usuário. Baseado nesses dados o *Adaptation Decision* busca os perfis (*profile.find*) e determina se será realizado um serviço de adaptação. No caso mostrado na Figura 5, não existe essa necessidade. Dessa forma o *Proxy Manager* verifica, sem sucesso se o conteúdo original está armazenado em *Cache* (*cache.search*). Após essa busca em *Cache*, é solicitada outra análise de adaptação (ponto 2).

Uma vez que não foi necessário adaptar a requisição do usuário, o conteúdo é solicitado do servidor de origem através de *cTP.sendRequest* e sua resposta é coletada através de *cTP.receiveResponse*. Em seguida o *Proxy Manager* solicita uma análise de adaptação referente ao ponto 3 de processamento. Nesse caso *Adaptation Decision* retorna

a necessidade de executar uma adaptação remota. Então o *Proxy Manager* solicita essa adaptação ao componente *Callout Protocol Client* (*cPC.invoke*), indicando o endereço (*url*) do servidor de adaptação, a ação a ser executada (*action*) e os parâmetros dessa ação (*parameters*), além do próprio conteúdo (*content*). O conteúdo adaptado é armazenado pelo *Proxy Manager* em *Cache* (*cache.write*). Em seguida o mesmo solicita outra análise de adaptação (*policy.request* - ponto 4), que nesse caso não retorna uma adaptação a ser realizada. Finalmente o conteúdo adaptado é retornado ao usuário (*cTP.sendResponse)*.

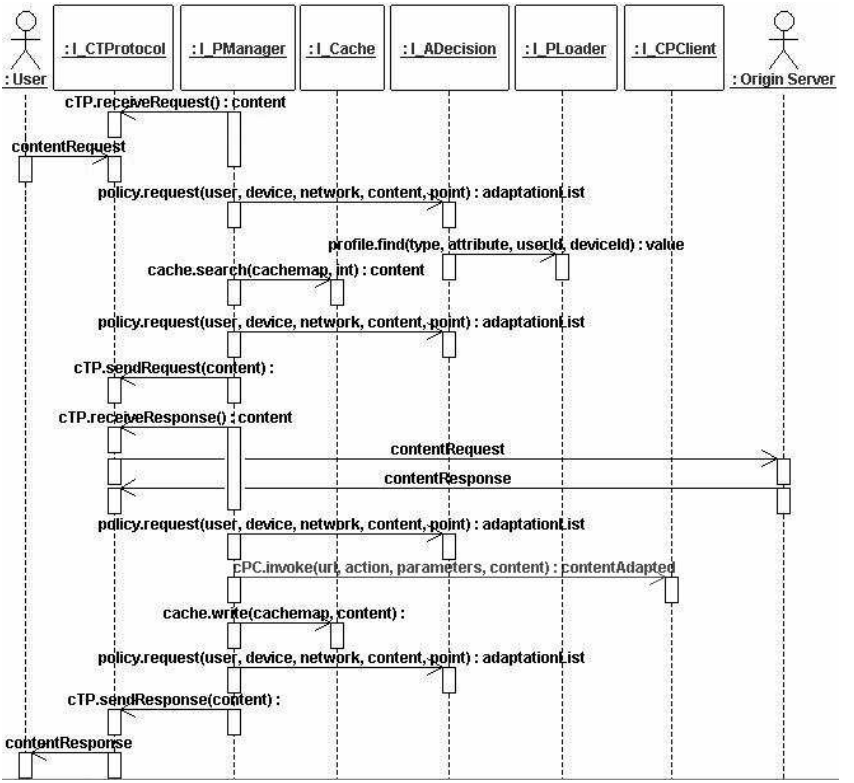

Figura 5. Modelo de seqüência do *Adaptation Proxy* na adaptação *respmod*.

No modelo de seqüência da Figura 6 o *Adaptation Proxy* solicita um serviço (*serviceRequest*) ao componente *Callout Protocol Server,* que recebe a solicitação através do método *calloutProtocol.receive*, juntamente com o cabeçalho desse protocolo e o conteúdo a ser adaptado. Esse componente analisa o cabeçalho do protocolo, extraindo deste a ação a ser executada e seus parâmetros. Assim o *Callout Protocol Server* invoca, através do método *service.invoke*, a adaptação ao componente *Remote Adapter*, que por sua vez realiza a adaptação de conteúdo solicitada (*adaptContent*). Em seguida o conteúdo adaptado é encapsulado numa mensagem de resposta e retornado ao *Adaptation Proxy* através do método *calloutProtocol.send*.

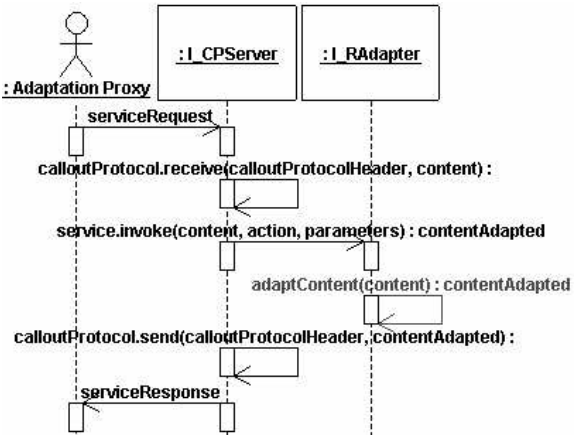

Figura 6. Modelo de seqüência de uma adaptação no *Adaptation Server*

4. Reutilização do FACI

O FACI foi desenvolvido com o objetivo de fornecer uma estrutura básica para o desenvolvimento das aplicações de adaptação de conteúdo na Internet através do reuso de seus componentes. Esta seção apresenta uma experiência de reutilização do framework proposto, representada no modelo de componentes da Figura 7. Nesse modelo têm-se os componentes que foram reutilizados do FACI através da instanciação direta: *Local* Adapter, *Remote Adapter*, *Proxy Manager*, *Cache, Adaptation Rules Updater*, *Network Data Coletor* e *Profile Loader*.

Uma vez que a implementação da política de adaptação é a principal dificuldade encontrada nos sistemas de adaptação, o FACI define que essa política deve considerar as regras de adaptação e as informações relativas ao ambiente de adaptação. Contudo, existem diversas formas de realizar esse processo de decisão, inclusive através da implementação de um algoritmo em linguagem procedimental. Assim, para atender este requisito específico da aplicação alguns componentes foram adicionados e outros modificados, conforme descrito a seguir.

4.1. Inserção de um mecanismo de inferência

A decisão da política de adaptação do FACI é de responsabilidade do componente *Adaptation Decision*. Nessa experiência este foi adaptado para trabalhar integrado com um mecanismo de inferência que, baseado nas regras de adaptação e nas informações do ambiente, define quais os serviços de adaptação devem ser realizados. O *Adaptation Decision* define apenas a ordem de execução desses serviços, caso existam múltiplas adaptações, e determina o adaptador (local ou remoto) que realizará essas adaptações.

Nessa reutilização do FACI, o mecanismo de inferência foi introduzido através de uma base de conhecimento (***Knowledge Base – KB***), implementada em linguagem Prolog. Esse mecanismo fornece uma maior flexibilidade à política de adaptação do framework e uma inteligência no processo de tomada de decisão.

Além da base de conhecimento, para o funcionamento desse mecanismo, foram adicionados dois novos componentes: *KB Manager* e *Sql2Prolog*. O ***KB Manager*** gerencia o funcionamento da base de conhecimento, realizando as seguintes funções: recebe do componente *Adaptation Rules Updater* as atualizações das regras de adaptação, convertendo-as para a linguagem Prolog e inserindo-as na *KB*; recebe do componente *Network Coletor Data* as modificações do estado corrente da rede de acesso, inserindo-as na *KB*; e quando solicitado pelo *Adaptation Decision* faz uma consulta de adaptação à *KB*, informando o usuário, dispositivo, conteúdo e rede de acesso que devem ser considerados. Para responder a esta consulta, a *KB* utiliza os perfis de usuário, dispositivo e de contrato de serviços, além das informações sobre a rede de acesso e o conteúdo. Considerando que a *KB* está implementada em Prolog e que os perfis estão na base *ProfilesDB*, o qual utiliza a linguagem de consulta SQL, existe uma incompatibilidade entre essas bases. Dessa forma foi desenvolvido o componente ***Sql2Prolog***, que viabiliza a troca de informações entre essas bases através de uma interface, permitindo à *KB* receber e analisar os perfis armazenados na base de dados.

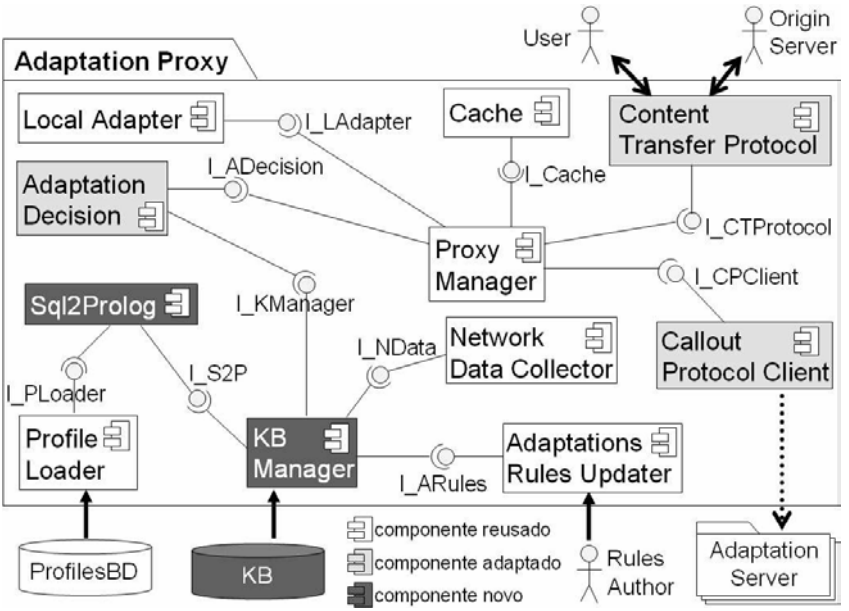

Figura 7. Reutilização do FACI

4.2. Reuso adaptado dos componentes *Content Transfer Protocol*, *Callout Protocol Client* e *Callout Protocol Server*

Nessa experiência de reutilização do FACI, três componentes foram adaptados com implementações específicas de suas interfaces, caracterizando o reuso através de especialização. Assim o componente *Content Transfer Protocol* foi especializado para efetuar a transmissão das requisições e respostas de conteúdo através do HTTP. Esse protocolo é bastante utilizado na Internet para transmissão de páginas Web, imagens, arquivos de vídeos, páginas dinâmicas ou aplicações. O HTTP é constituído por dois

campos, o cabeçalho e a carga útil (conteúdo). O componente modificado contém um *parser* HTTP, que identifica as ações semânticas desse protocolo.

Os componentes *Callout Protocol Client* e *Callout Protocol Server* foram especializados para viabilizar a comunicação entre o dispositivo de borda e o servidor de adaptação através do ICAP. Este é um protocolo "leve" com cabeçalhos de requisição e de resposta similares aos cabeçalhos HTTP. Numa mensagem ICAP são encapsuladas o conteúdo e as mensagens HTTP trocadas entre o usuário e o servidor de origem. Ao receber uma solicitação de serviço, o componente *Callout Protocol Client* encapsula as ações, seus parâmetros e o conteúdo numa mensagem ICAP de requisição e a envia ao *Callout Protocol Server*. Este faz uma análise semântica dessa mensagem, retirando as informações necessárias e invoca o serviço solicitado ao *Remote Adapter*. O conteúdo, após ser adaptado, é encapsulado pelo *Callout Protocol Server* numa mensagem ICAP de resposta e retornado ao *Adaptation Proxy.*

4.3. Funcionamento da política de adaptação

Com a inserção de novos componentes nessa experiência de reuso do FACI, o funcionamento da sua política de adaptação sofreu alterações, uma vez que em seu processo de análise foi inserido um mecanismo de inferência. Assim o *Adaptation Decision* foi alterado e a análise de adaptação foi atribuída à base de conhecimento. A nova interação entre os componentes dessa política é descrita no modelo de seqüência ilustrado na Figura 8.

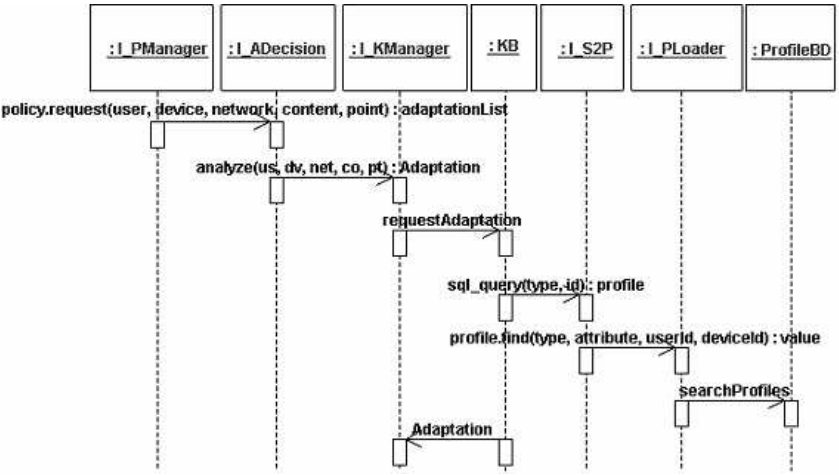

Figura 8. Requisição da análise de adaptação

Nesse modelo o *Proxy Manager* continua solicitando a análise de adaptação através do método *policy.request*, pertencente ao componente *Adaptation Decision*. No entanto, esse componente solicita ao *KB Manager, através do método analyse,* a decisão sobre quais as adaptações que deverão ser efetuadas. Em seguida, o *KB Manager* busca na base de conhecimento (*KB*) quais as adaptações são necessárias. Nessa consulta, o *KB Manager* envia, por parâmetro, informações sobre a rede de acesso e sobre o conteúdo, além dos identificadores de usuário e dispositivo. A partir destes, a *KB*

consulta os perfis de usuário, dispositivo e contrato de serviços usando o método *sql_query* do componente *Sql2Prolog*. Este por sua vez solicita a consulta ao *Profile Loader* através de *profile.find*, transformando os dados da consulta *SQL* na linguagem *Prolog* de forma que a base de conhecimento possa manipulá-los. Após realizar a análise de adaptação, a *KB* envia o seu resultado ao *KB Manager*, que retorna as adaptações necessárias ao *Adaptation Decision*. Este por sua vez finaliza a análise de adaptação retornando as adaptações necessárias ao *Proxy Manager*.

4.4. Avaliação de desempenho do reuso proposto

A avaliação de desempenho é uma questão importante a ser considerada quando são oferecidos serviços de adaptação, pois a execução destes tende a elevar o tempo de entrega do conteúdo. Como o FACI visa, dentre outros propósitos, a satisfação do usuário, essa entrega não deve ser demorada. Dessa forma esse framework se preocupa em oferecer seus serviços sem degradar o desempenho da infra-estrutura da Internet. Além disso, visto que o FACI define uma política de adaptação, esta deve ocupar pouco tempo de sua execução. Assim a avaliação desse reuso visa aferir o tempo consumido na adaptação de conteúdo e na política de adaptação, compreendendo o mecanismo de inferência adicionado.

A avaliação de desempenho desse reuso foi efetuada numa rede formada por cinco computadores: o *Adaptation Proxy*, três *Adaptation Servers* e o *User*. Os *Origin Servers* foram acessados diretamente da Internet. A implementação do *Adaptation Proxy* baseou-se em [Proxy Shweby]. O servidores de adaptação implementados nessa avaliação foram os seguintes: *Virus Scan* (VS), *Content Filter* (CF) e *Image Adapter* (IA). Os servidores VS e IA basearam-se em [ICAP Server], enquanto o servidor CF foi desenvolvido em outro trabalho deste grupo de pesquisa [Forte 2004].

Para avaliar o desempenho dos componentes desse reuso, definiram-se certos pontos que medem os tempos gastos num fluxo de requisição e resposta de conteúdo. A Figura 9 apresenta esses pontos que fornecem os seguintes tempos: *T(Origin Server)*, que representa o tempo da resposta do servidor de origem; *T(Analysis)*, referente ao tempo gasto no processo da política de adaptação; *T(Adaptation)*, que mede o tempo gasto pelo protocolo ICAP e pelas adaptações do conteúdo; e *T(Delivery)*, que representa o tempo de entrega do conteúdo ao usuário após todas as adaptações.

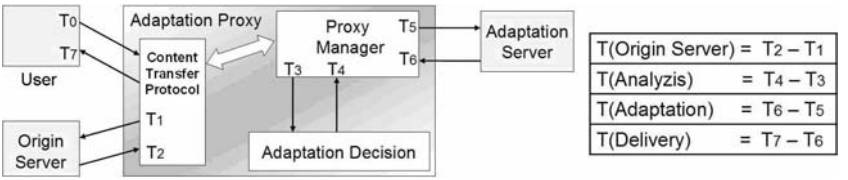

Figura 9. Tempos definidos para a avaliação do reuso do FACI

A partir das definições desses tempos, é possível examinar o *overhead* ocasionado pela política de adaptação e pelas adaptações de conteúdo, respectivamente, através dos tempos *T(Analysis)* e *T(Adaptation)*. Para efetuar essa avaliação, foram executadas 1000 requisições da página *www.folha.com.br*. Através dos *Adaptation Servers* implementados, essa página Web sofreu 5 testes diferentes de adaptações: sem adaptação (NA); utilizando separadamente os três *Adaptation Servers* (CF, VS e IA); e combinando esses três servidores (CF+VS+IA). A Figura 10 apresenta a média dos

tempos *T(Origin Server)*, *T(Analysis)*, *T(Adaptation)* e *T(Delivery)* coletados nos testes realizados.

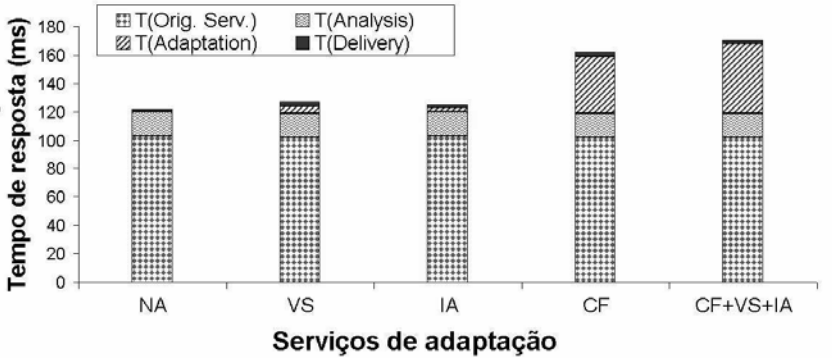

Figura 10. *Overheads* do framework

Nesses testes pode-se verificar que o maior atraso ocorreu na espera da resposta do servidor de origem e o tempo gasto no processo da política de adaptação foi semelhante em todas as adaptações testadas. Nos testes realizados sem a adaptação (NA), o tempo total da entrega do conteúdo teve média de 121 ms (milissegundos), sendo 103 ms referentes à resposta do servidor de origem, 17 ms referentes à política de adaptação e apenas 1 ms referente à entrega do conteúdo ao usuário. Dessa maneira, percebe-se que o atraso ocasionado pelo mecanismo de inferência (17 ms) é relativamente pequeno e satisfatório para a aplicação.

Utilizando os servidores VS e IA, o tempo médio gasto nas adaptações de conteúdo foi de 5,3 ms e 2,5 ms respectivamente. Logo esses servidores se mostraram eficazes, provocando um atraso pouco significativo na entrega do conteúdo. Somente o servidor CF ocasionou um atraso relevante de 40 ms.

5. Trabalhos Correlatos

Em meados da década de 90, com a proliferação do uso da Internet, os trabalhos dessa área se preocupavam em entregar conteúdos adaptados a seus usuários. Inicialmente as adaptações de conteúdo foram realizadas no próprio dispositivo de borda [Smith 1998 e Bharadvaj 1998]. A desvantagem agravante dessa técnica vem do acúmulo das adaptações de conteúdo que podem sobrecarregar o dispositivo de borda. A partir da elaboração do modelo de redes OPES, onde as adaptações são distribuídas a servidores dedicados, tornou-se viável a construção de uma única arquitetura que oferecesse vários tipos de adaptação. No entanto, para que essa arquitetura não realize todas as adaptações oferecidas, deve-se oferecer a política de adaptação [Babir 2004].

Em [Beck 2001] é apresentado uma arquitetura para realização dos serviços de adaptação que contém mecanismo de tomada de decisões baseado num conjunto de condições. Essas condições e as ações relacionadas constituem as regras de adaptação, que nesse trabalho foram especificadas através de IRML. Porém, essas condições das regras eram limitadas, pois não utilizavam as informações do ambiente de adaptação.

Já em [Ravindran 2002] é apresentado um framework para gerenciamento de personalização de serviços, sendo sua política de adaptação baseada numa combinação das preferências do usuário, das restrições do dispositivo e das características do conteúdo.

Em [Marques 2004] foi apresentada uma arquitetura de adaptações direcionadas apenas a dispositivos móveis e que oferece adaptações de imagem, vídeo e compressão de textos. Nesse trabalho também foram utilizadas informações sobre a rede de acesso para tomar as decisões quanto a melhor adaptação.

Diferentemente dos trabalhos relacionados, este artigo apresenta um framework baseado em componentes que oferece uma flexibilidade através da reutilização de seus componentes. Além disso, no framework é definida uma política de adaptação que se baseia nos perfis de usuário, de dispositivo e de contrato de serviços, além das informações sobre a rede de acesso e o conteúdo.

6. Considerações Finais e Trabalhos Futuros

Este trabalho apresentou um framework baseado em componentes para o domínio de adaptação de conteúdo na Internet e uma experiência do seu reuso. Nessa experiência o framework foi instanciado para a utilização dos protocolos ICAP e HTTP, além de inserir um mecanismo de inferência na implementação da política de adaptação, baseado na linguagem Prolog.

O uso desse mecanismo de inferência facilita a manutenção das regras de adaptação e oferece uma flexibilidade ao processo de decisão do framework. Neste trabalho também foi demonstrado que a inserção desse mecanismo satisfaz, em termos de desempenho, a entrega do conteúdo ao usuário.

Espera-se com o reuso do framework em novas experiências, que seus componentes sejam refinados para atender um maior número de aplicações. Acredita-se que com uma maior generalização do framework seja possível definir padrões de software para o domínio de adaptação de conteúdo na Internet.

Referências

Akamai White Paper (2000). “Internet Bottlenecks: the case for Edge Delivery Services”. Disponível em: <www1.cs.columbia.edu/~danr/6762/week1/akamai-bottlenecks.pdf>.

Babir, A. et al. (2004) “Policy, Authorization, and Enforcement Requirements of the OPES”. IETF Request for Comments 3507. Disponível em <www.ietf.org/rfc.html>

Beck, A.; Hofmann, M. (2000) “Example Services for Network Edge Proxies”. Internet Draft. <standards.nortelnetworks.com/opes/non-wg-doc/draft-beck-opes-esfnep-01.txt>

Beck, A.; Hofmann, M.; (2001). “Enabling the Internet to deliver content-oriented services”. Sixth Int. Workshop on Web Caching and Content Distribution, EUA.

Beck, A.; Rousskov, A. (2003) “P: Message Processing Language”. IETF Internet Draft. Disponível em: <www.measurement-factory.com/tmp/opes/>.

Bharadvaj, H.; Joshi, A.; Auephanwiriyakul, S. (1998). "An Active Transcoding Proxy to Support Mobile Web Access". IEEE Symposium on Reliable Distributed Systems, p. 118-123.

Buchholz, S.; Schill, A (2003). "Adaptation-aware web caching: Caching in the future pervasive web". <citeseer.ist.psu.edu/buchholz03adaptationaware.html>.

Elson, J.; Cerpa, A. (2003) "Internet Content Adaptation Protocol". IETF Request for Comments 3507. Disponível em: <www.ietf.org/rfc.html>.

Forte, M.; Souza, W.; Prado, A. (2004). "Servidor de classificação e filtragem de conteúdo com suporte ao protocolo ICAP". Workshop Coop. UFSCar - FSA em Ciência da Computação.

Hofmann, M.; Beck, A. (2001) "IRML: A Rule Specification Language for Intermediary Services". IETF Internet Draft. Disponível em: <standards.nortelnetworks.com/opes/ non-wg-doc/draft-beck-opes-irml-02.txt>.

ICAP Server by Network Appliance. http://www.i-cap.org/docs/.

McHenry, S. et. al. (2001) "OPES Use Cases and Deployment Scenarios". IETF Internet Draft. Disponível em: <standards.nortelnetworks.com/opes/non-wg.htm>

Marques, M.; Loureiro, A.; (2004). "Adaptation in mobile computing". 23o. Simpósio Brasileiro de Redes de Computadores.

Proxy Shweby. http://shweby.sourceforge.net/.

Ravindran, G.; Jaseemudin, M.; Rayhan, A.; (2002). "A management framework for service personalization". IEEE Conf. on Management of Multimedia Networks and Services, EUA.

Rousskov, A. (2005). "OPES Callout Protocol (OCP) Core". IETF Request For Comments 4037. Disponível em: <www.ietf.org/rfc/rfc4037.txt>

Smith, J.; Mohan, R.; Li, C. (1998). "Content-based Transcoding of Images in the Internet". Anais da IEEE Int. Conference on Image Processing (ICIP), p. 7-11.

Souza, W. et. al.; (2002). "Adaptação de conteúdo baseada em perfis de dispositivo, conteúdo, usuário e serviço de rede". 20º Simpósio Brasileiro de Redes de Computadores. vol. 2, p. 554-568.

Tommlinson, G. et. al. (2001). "A Model for Open Pluggable Edge Services". IETF Internet Draft, Work In Progress. http://www.ietf.org/internet-drafts/drafttomlinson-opes-model-01.txt.

Vakali, A.; Pallis, G. (2003). "Content Delivery Networks: Status and Trends". IEEE Internet Computing, Vol. 7, No. 6, pp. 68-74.