

Provisão de QoS Inter e Intra Objetos de Mídia em Formataadores Adaptativos¹

Rogério Ferreira Rodrigues, Luiz Fernando Gomes Soares

Laboratório TeleMídia – Departamento de Informática – PUC-Rio
R. Marquês de São Vicente, 225 – Gávea – 22453-900, Rio de Janeiro, RJ, Brasil

rogerio@telemidia.puc-rio.br, lfgs@inf.puc-rio.br

Resumo. *O desenvolvimento de sistemas multimídia/hipermídia requer a implementação de um elemento, denominado formador, que é responsável por receber a especificação de um documento e controlar sua apresentação. O controle da apresentação de um hiperdocumento com um resultado de qualidade aceitável é um problema que envolve a orquestração da QoS, devendo ser tratado em dois níveis relacionados: a orquestração inter objetos de mídia e a orquestração intra objetos de mídia. Este artigo objetiva discutir as questões associadas à provisão de QoS em sistemas hipermídia, focando no projeto e implementação de formadores. O trabalho discute a importância dos mecanismos de adaptação e pré-busca nos sistemas hipermídia e propõe um framework de QoS para formadores hipermídia. O artigo também comenta a experiência obtida com a instanciação do framework para a implementação do formador do sistema HyperProp.*

Abstract. *The development of multimedia/hypermedia systems requires the implementation of an element, named formatter, which is in charge of receiving the specification of a document and controlling its presentation. The process of controlling and maintaining the presentation of a hyperdocument with an output of acceptable quality is a QoS orchestration problem, which needs to be treated by formatters in two related levels: the inter media-object and the intra media-object orchestration. This paper aims to discuss the issues associated to QoS provisioning in hypermedia systems, focusing on the design and implementation of formatters. The paper discusses the importance of adaptation and prefetching mechanisms and proposes a QoS framework for hypermedia formatters. The article also comments the experience obtained in the framework instantiation for the HyperProp system formatter implementation.*

1. Introdução

Formatador é um dos nomes que tem sido utilizado na literatura para designar o elemento responsável por controlar a apresentação de documentos multimídia/hipermídia [BuZe93, SoRM00]. Além de coordenar a apresentação de conteúdos de diferentes tipos de mídia, formadores hipermídia devem procurar garantir que os relacionamentos entre os objetos de um documento sejam respeitados, assegurando uma apresentação de boa qualidade.

O processo de oferecimento de uma apresentação de hiperdocumento de qualidade aceitável pode ser considerado como um problema de provisão de qualidade de serviço

¹ Este trabalho foi realizado com apoio do Fundo Setorial para o Desenvolvimento Tecnológico das Telecomunicações (FUNTTEL), através do contrato 0594/02.

(QoS), que precisa ser tratado pelo formatador em dois níveis. Primeiro existe a questão da sincronização entre os objetos de mídia, às vezes designada simplesmente sincronização inter-objetos, ou inter-mídia. Esse tipo de sincronização está associado com as relações espaciais e temporais especificadas pelo autor (relações de restrição, relações de causalidade etc.), a duração de apresentação assumida por cada objeto de mídia e, evidentemente, os instantes em que as ações de exibição (início, término, pausa etc.) são executadas. O segundo nível é a sincronização interna ao objeto de mídia, também denominada sincronização intra-objeto, ou intra-mídia. Essa sincronização está vinculada ao momento em que o conteúdo de cada objeto de mídia é recuperado do seu local de armazenamento, à taxa e ao retardo de transferência e às políticas de escalonamento de tarefas/processos nos sistemas operacionais envolvidos na busca e apresentação dos conteúdos. De fato, a sincronização intra-objeto é um suporte importante para que a sincronização inter-objetos seja corretamente mantida.

Com o intuito de controlar a exibição de um documento, o formatador deverá construir um plano de apresentação, que irá conter as características de exibição de cada objeto de mídia, a programação das tarefas a serem escalonadas e as informações dos relacionamentos de sincronização entre os objetos. Essa estrutura irá auxiliar justamente na manutenção da sincronização inter-objetos. Por outro lado, a sincronização intra-objetos será desempenhada com o auxílio de mecanismos de reserva de recursos e também, se possível, de mecanismos de pré-busca (um plano de pré-busca), que permitam ao formatador antecipar-se às ações vinculadas à exibição do documento.

Para oferecer uma QoS inter e intra objetos será necessário dividir, dentro do sistema hipermídia, as responsabilidades de provisão entre os elementos envolvidos no processo de apresentação dos documentos: formatador, servidores de conteúdo, sistemas operacionais, rede etc. Esse procedimento de divisão é conhecido como orquestração da QoS e ocorre tanto no momento de negociação do serviço de exibição (*fase de negociação*) como durante a própria exibição (*fase de sintonização*) [GoCS01].

O objetivo deste trabalho é discutir os aspectos relacionados à provisão de QoS em sistemas hipermídia, focando no projeto e implementação dos formatadores de documentos. Como uma especialização dos frameworks de QoS genéricos descritos em [GoCS01], este artigo propõe um framework de QoS para formatadores hipermídia. No entanto, a proposta pode ser estendida para outros domínios de aplicação que também necessitem de sincronização em tempo real.

O trabalho também aborda os aspectos de adaptação de documentos, principalmente no que se refere ao ajuste da qualidade da informação apresentada. A existência de recursos de adaptabilidade no sistema hipermídia irá apoiar tanto a sintonização das apresentações, fazendo com que o sistema tenha capacidade de reagir diante de acontecimentos imprevisíveis (violações na QoS negociada, ações do usuário etc.) como também a própria etapa de negociação, permitindo que autores tenham uma maior flexibilidade para criação dos documentos.

Apesar de ser desejável dispor o sistema de recursos de adaptação, adaptar documentos, principalmente na fase de sintonização, é algo que consome recursos computacionais e, sempre que possível, deve ser evitado durante a execução dos documentos, reforçando a importância dos mecanismos de pré-busca na provisão da QoS intra-mídia, conforme mencionado anteriormente.

Este artigo está estruturado da seguinte forma. A Seção 2 aborda as questões relacionadas à adaptação em formatadores hiper-mídia. A Seção 3 apresenta, com base em um modelo genérico de orquestração de QoS, a especialização desse modelo para o domínio específico de formatadores hiper-mídia. A Seção 4 discorre sobre a instanciação realizada do framework proposto para o formatador HyperProp [SoRM00, Rodr03]. A Seção 5 comenta alguns trabalhos relacionados e, por fim, a Seção 6 apresenta as conclusões e os trabalhos futuros.

2. Adaptação de Apresentações Hiper-mídia

A existência de mecanismos de adaptação nos sistemas hiper-mídia torna a apresentação dos documentos uma aplicação orientada a contexto [ScAW94]. O objetivo é dispor de algoritmos capazes de transformar os hiperdocumentos a fim de torná-los mais apropriados às informações que os circundam, tais como parâmetros de desempenho dos sistemas de comunicação, recursos disponíveis nas plataformas de exibição, características e preferências do usuário, localização etc.

Documentos hiper-mídia podem ser adaptados principalmente com relação à qualidade, à forma e à semântica da informação a ser exibida. A adaptação da qualidade está associada com variação de resolução, variação de taxa (para mídias contínuas), mudança na duração dos objetos e, em alguns casos, até substituição de um tipo de mídia por outro, também chamada de adaptação *cross-media* [BoKW99]. A adaptação da forma de exibição diz respeito ao estilo de apresentação de um conteúdo, o que pode afetar ou não a qualidade da informação. Isso faz com que as adaptações da qualidade e da forma de exibição da informação estejam intimamente relacionadas com a adaptação dos conteúdos dos objetos (os *nós*) e de suas características de apresentação (os *descritores*). Por outro lado, a adaptação da semântica (ou da quantidade) da informação de um documento pode estar relacionada não apenas com o ajuste nos conteúdos e das características de apresentação, mas também com modificações nos próprios relacionamentos (os *elos* ou *composições*) [Brus96]. Obviamente, pode ser interessante combinar adaptações tanto da quantidade como da qualidade da informação para um mesmo objeto hiper-mídia. Por exemplo, o vídeo de uma aula a ser exibida para um aluno pode ser escolhido dependendo do seu histórico e, posteriormente, a sua taxa de exibição definida em função da banda passante da conexão de rede utilizada pelo aluno. Apesar da proposta neste artigo não restringir o objetivo da adaptação, o trabalho concentra-se nos ajustes referentes à qualidade das apresentações.

Um caso particular de adaptação de objetos de mídia pela variação da qualidade que merece destaque é o ajuste das durações, também denominado *ajuste elástico dos tempos* [BMRS00, BuZe93, KiSo95, LaSR02]. A idéia é aumentar (*stretch*) ou diminuir (*shrink*) a duração dos objetos em relação aos valores nominais, com o objetivo de garantir, principalmente, a QoS inter-mídia do documento, ou seja, respeitar as restrições temporais entre os objetos impostas pelo autor. Nada impede, porém, que esse tipo de adaptação ocorra também em pequenos trechos internos aos objetos, para compensar atrasos e manter a QoS intra-mídia da exibição de um determinado nó.

Para compreender o processo de adaptação, é preciso examinar não apenas o objeto da adaptação (os documentos hiper-mídia), mas também os principais elementos que influenciam os ajustes. Existem dois importantes conjuntos de parâmetros externos aos documentos que podem exercer influência sobre suas adaptações: os parâmetros do

usuário e os parâmetros da infra-estrutura [BoKW99]. Os valores desses conjuntos de parâmetros são justamente as informações que reunidas compõem o *contexto de exibição* de uma aplicação hipermídia, mencionado no princípio desta seção. A adaptação às características do usuário pode levar em consideração, por exemplo, o seu conhecimento a respeito dos assuntos tratados no conteúdo dos documentos, os seus objetivos, as suas preferências e o comportamento do usuário durante a sua navegação. Já a adaptação à infra-estrutura concentra-se nos aspectos relacionados à plataforma de exibição, tais como banda passante do sistema de comunicação, poder computacional e memória disponíveis nos equipamentos participantes da transmissão e da exibição do documento, dispositivos de E/S presentes na máquina onde o documento é exibido etc.

Manter os valores desses vários parâmetros exige a introdução do conceito de *gerência de contexto* nos sistemas hipermídia adaptativos. Devido ao dinamismo de algumas das informações, tais como banda passante e recursos de processamento e armazenamento disponíveis, é útil que haja agentes monitores distribuídos pelo sistema a fim de coletar os dados pertinentes para constituição do contexto. Este artigo não trata dos aspectos ligados à gerência de contexto, mas assume a sua existência para apoiar a formatação/controlar das apresentações dos hiperdocumentos.

De um modo geral, as características da infra-estrutura de exibição permitem guiar a adaptação dos documentos em termos de ajuste da qualidade dos conteúdos dos objetos. Em algumas situações, principalmente com relação às características dos dispositivos de E/S na plataforma de apresentação, as informações referentes à infra-estrutura podem também direcionar a escolha das características de apresentação dos nós e a escolha entre alternativas de tipos de mídia diferentes. Em contrapartida, os parâmetros relacionados ao usuário são mais propícios à adaptação da semântica e estilo das informações contidas nos documentos, assim como à adaptação dos relacionamentos entre os nós hipermídia. Como este trabalho se propõe a focar nas questões de ajuste de qualidade, o seu desenvolvimento é centrado na adaptação dos documentos em função das características da infra-estrutura de exibição. Todavia, é importante ressaltar que nem toda adaptação depende exclusivamente de informações externas aos documentos. O ajuste elástico temporal é um exemplo de adaptação que é feito também com base nas especificações dos relacionamentos entre os nós e de suas durações.

Evidentemente, a existência de mecanismos de adaptação em sistemas hipermídia requer uma flexibilidade na forma como os documentos são especificados, trazendo também como consequência uma necessidade de flexibilidade para os modelos conceituais hipermídia. Um recurso desejável para os modelos é a possibilidade do autor especificar partes do documento como conjuntos de alternativas, tendo a cada opção de alternativa associada uma regra que, com base nas informações do contexto de exibição, possa ser avaliada e assim determinar a escolha mais adequada. É interessante que o modelo permita definir não apenas alternativas de objetos de mídia mas também alternativas de características de apresentação, alternativas de relacionamentos e até mesmo alternativas entre estruturas compostas (conjuntos de nós, conjuntos de elos etc.). Melhor ainda se alguns dos conjuntos de alternativas puderem ser gerados automaticamente pelo sistema, simplificando a tarefa de autoria. Evidentemente, essa geração automática é mais propícia a adaptações que contemplem a qualidade das informações, sendo complexa quando o objetivo é ajustar a semântica do documento.

No caso do ajuste elástico temporal, as durações dos objetos devem ser especificadas

não mais de uma maneira rígida, com um único valor, mas sim com uma faixa de valores aceitáveis. Mais interessante ainda se esses valores forem descritos através de funções de custo que ofereçam métricas, permitindo à estratégia de adaptação decidir qual a melhor escolha para uma dada apresentação e contexto de exibição [BMRS00, BuZe93, KiSo95].

Para que um documento seja adaptado, é necessário que exista no sistema hipermídia um componente capaz de analisar a sua descrição, quando necessário em conjunto com as informações contextuais, e com esses dados aplicar as estratégias de ajuste. Em sistemas hipermídia distribuídos, as estratégias podem ser executadas tanto no ambiente de armazenamento (servidor) como no ambiente de execução (formatador no cliente), podendo estar inclusive presentes nos dois ambientes paralelamente. Pode ser também adequada, para reduzir o processamento em clientes e servidores e ao mesmo tempo reutilizar os resultados para mais de um cliente, a inclusão dessas funcionalidades em agentes intermediários no sistema (*proxies*).

Com relação ao momento em que a adaptação ocorre, é possível separar em duas opções: antes e durante a apresentação do documento. As adaptações realizadas antes da exibição estão relacionadas às informações contextuais estáticas, sendo por isso essa opção também classificada como adaptação estática, enquanto o ajuste em tempo de execução é também classificado como dinâmico, pois depende de parâmetros cujos valores se modificam ao longo da exibição do documento [BoKW99].

Os mecanismos de adaptação dinâmica relacionados à correção de perdas de sincronização inter-mídia e intra-mídia devem atuar cooperativamente com o formatador do sistema hipermídia. A razão está no fato do formatador ser o elemento mais próximo do usuário, que é quem efetivamente perceberá tais inconsistências, que terminam por degradar a qualidade da exibição. Além disso, o usuário, através de suas ações de navegação, será uma importante fonte de geração de eventos imprevisíveis na execução de um documento hipermídia, fazendo com que a adaptação no formatador tenha condições de corrigir o curso da exibição com um menor tempo de resposta. Evidentemente, existem justificativas para a existência de mecanismos de adaptação em proxies e servidores (redução da carga na rede, por exemplo), mas que não eliminam a importância da adaptação no cliente hipermídia.

A próxima seção propõe uma estruturação que modela o funcionamento de formatadores hipermídia, focando na questão da orquestração da QoS inter-objetos e intra-objetos e mostrando mais claramente o papel da adaptação na provisão da QoS.

3. Orquestração de QoS em Formatadores Hipermídia

Componentes do usuário e provedores de serviços delineiam o que é chamado de um *ambiente de oferecimento de serviços* (SOE – *Service-Offering Environment*) [GoCS01]. O SOE de um sistema hipermídia é ilustrado na Figura 1. Na figura, os *componentes do usuário* (servidor e formatador em cinza hachurado) possuem *portas de interação básicas* (BL – *Base-Level*)² através das quais se comunicam, por meio de um *provedor de serviços de comunicação*. Tanto os provedores de serviços como os componentes do usuário podem ser compostos de outros componentes e provedores,

² Para não sobrecarregar a figura, apenas alguns componentes tiveram suas portas BL ilustradas.

que, por sua vez, podem ser também compostos. Esse aninhamento pode ocorrer até o nível de detalhe que o projetista do sistema estiver interessado (nível *primitivo*). A Figura 1 exibe com mais detalhes o componente composto representando um formador em um sistema hipermídia. Usualmente, os provedores de serviços de comunicação internos ao formador e ao servidor são os sistemas operacionais nas máquinas cliente e servidora, respectivamente.

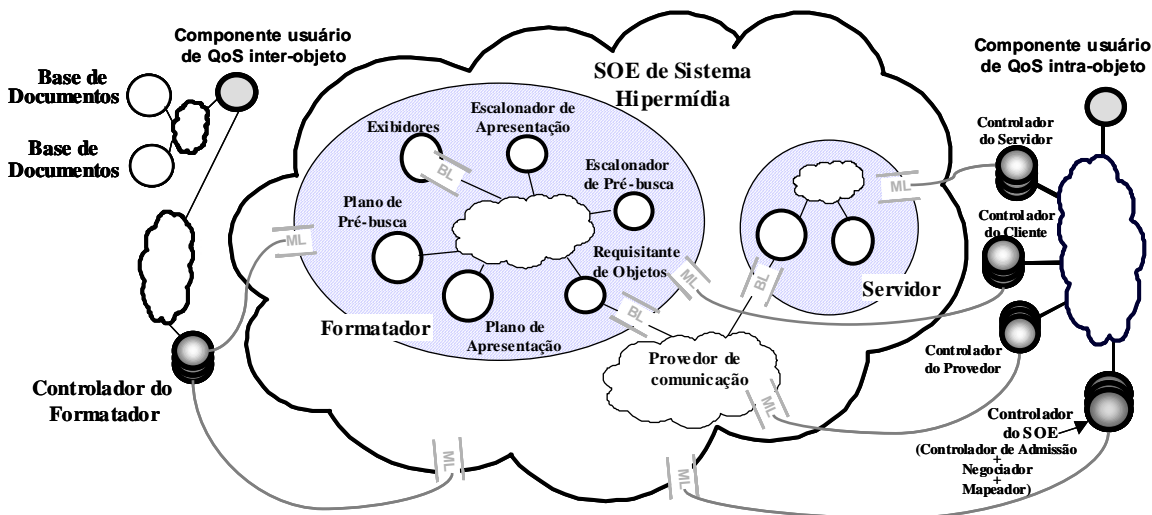


Figura 1 – O SOE de um sistema hipermídia e os meta serviços de QoS

A partir da requisição de um usuário para iniciar a apresentação de um documento, o componente *Escalonador de Apresentação*, membro do componente composto formador, é iniciado. O escalonador de apresentação precisa receber uma estrutura de dados especificando a ordem de exibição dos objetos de mídia do documento. Essa estrutura, chamada de *plano de apresentação*, é construída pelos meta serviços de QoS, conforme será explicado mais adiante. É importante destacar que o plano de apresentação não é necessariamente um *timeline*. Uma abordagem interessante é o uso de procedimentos de escalonamento baseados em eventos e relacionamentos entre eles; por exemplo, o final de um vídeo deve terminar a apresentação de uma trilha sonora (objeto áudio). Uma das primeiras propostas de estrutura de dados baseada em cadeias temporais de eventos pode ser encontrada no sistema Firefly [BuZe93].

Com o objetivo de compensar retardos e variações estatísticas de retardos (*jitters*) no sistema como um todo, e com isso evitar (ou reduzir) a necessidade de realizar adaptações no documento, o formador deve possuir um mecanismo de controle da pré-busca dos objetos a serem exibidos. Na implementação desse mecanismo, um importante componente é o *Escalonador de Pré-busca*. Ele coordena a obtenção dos objetos de seus locais de armazenamento (servidores de mídia, por exemplo), baseando-se em uma estrutura de dados que especifica a ordem de preparação dos mesmos. Essa estrutura de dados, chamada de *plano de pré-busca*, é também construída por meta serviços de QoS. Para recuperar o conteúdo de um novo objeto, o escalonador de pré-busca solicita os serviços do componente *Requisitante de Objetos*.

Uma vez iniciado, o escalonador de apresentação observa o plano de apresentação e, conforme necessário apresentar um objeto de mídia do documento, instancia novos exibidores de mídia, representados pelo componente *Exibidor* na Figura 1. O conteúdo

do objeto, previamente obtido pelo componente *Requisitante de Objetos*, é então passado para o exibidor, para que seja efetivamente apresentado.

Exibidores devem informar ao escalonador de apresentação a ocorrência de eventos de apresentação (início, término de exibição etc.). Baseado nesses eventos, nos eventos gerados pelo usuário (por exemplo, a seleção de uma âncora) e no plano de apresentação, o escalonador controla a execução de todo o documento.

Durante a exibição de um documento, podem ocorrer diferenças entre os instantes calculados no plano de apresentação para ocorrência dos eventos e os instantes em que as ocorrências são efetivamente reportadas pelos exibidores. Essas variações são percebidas pelo meta serviço de orquestração de QoS inter-objetos, como será visto na Seção 3.1, e podem requerer algumas ações de apresentação. Por exemplo, desvios nos instantes de apresentação previstos podem ser corrigidos através de comandos para os exibidores encurtarem ou aumentarem a duração de apresentação dos objetos.

Com o objetivo de fornecer uma QoS fim a fim (sincronização inter e intra objetos de mídia), é preciso que seja realizada a divisão das responsabilidades de provisão de QoS entre os recursos do SOE do sistema hipermídia (*orquestração da QoS*). Considerando a Figura 1, com o intuito de prover uma apresentação de documento com qualidade de serviço, todos os requisitos de QoS (por exemplo, o retardo fim a fim) precisam ser satisfeitos pelo trabalho conjunto dos componentes de processamento do SOE (formatador e servidor) e de comunicação entre eles (provedor de comunicação). Por sua vez, cada componente e provedor composto precisa orquestrar a porção de responsabilidade da provisão de QoS a ele outorgada entre seus componentes e provedores internos, e assim sucessiva e recorrentemente. Esse processo continua até que um componente ou provedor primitivo seja atingido, quando então recursos de processamento e comunicação são reservados para que se garanta a porção de QoS atribuída ao elemento, o que implica em adaptações no plano de escalonamento dos recursos. O processo inteiro modifica o estado interno do SOE.

Adaptações em um SOE, tais quais as exemplificadas pela provisão de QoS, podem ser representadas tendo por base o conceito de *implementações abertas* [Stey94]. Em uma implementação aberta, além das portas de interação básicas (*base-level*), componentes e provedores precisam também dispor de portas de interação *meta* (*ML - meta-level*), conforme ilustrado na Figura 1. Essas interfaces revelam alguns aspectos internos dos componentes e provedores, tornando-os susceptíveis a adaptações. Componentes responsáveis por fazer a adaptação são chamados *componentes meta*. Como qualquer componente, um componente meta pode comunicar-se com outros componentes meta através de um provedor, definindo com isso um *meta sistema*. Um SOE pode ser alvo de diversos componentes meta, pertencentes, ou não, a diferentes meta sistemas. Tanto a orquestração da QoS inter-objetos como a orquestração da QoS intra-objetos são exemplos de meta sistemas que serão discutidos em detalhes nas duas próximas seções.

3.1 Orquestração da QoS inter-objetos

O meta serviço de QoS inter-objetos tem como tarefa a orquestração dos recursos do SOE para que seja provida a sincronização inter-mídia. Como é muito raro um provedor de comunicação que permita negociar a QoS inter-objetos, e também porque é muito difícil prover uma QoS inter-objetos no lado do servidor, uma vez que os fluxos de mídia podem ser originados de diferentes máquinas, neste artigo define-se a provisão da

QoS inter-objetos como sendo de responsabilidade exclusiva do formatador. A Figura 2 ilustra em mais detalhes um dos meta serviços apresentados na Figura 1.

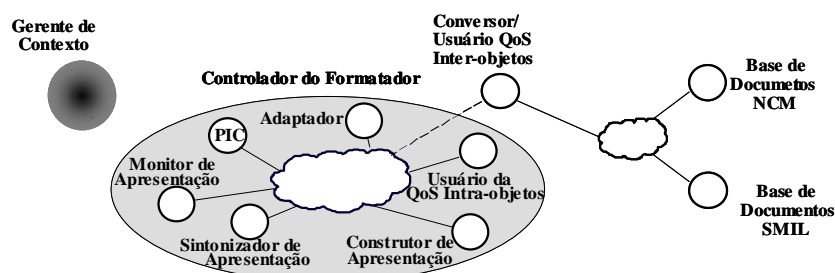


Figura 2 – Meta serviço de QoS inter-objetos

O *Conversor/Usuário da QoS Inter-Objetos* (componente *Usuário de QoS inter-objetos* na Figura 1) obtém a descrição de um documento, ou de parte dele, a partir de uma base de documentos. Se o modelo conceitual por trás da descrição do documento for diferente do modelo utilizado pelo meta serviço, a descrição obtida é então traduzida. Um contêiner com a informação traduzida é criado, tendo todas as especificações de QoS herdadas da descrição do documento, tais como os relacionamentos entre os objetos de mídia, as características de apresentação para cada objeto etc. O contêiner, junto com a indicação do evento de apresentação que inicia a apresentação do contêiner, é então entregue ao *Controlador do Formatador*, mais especificamente, a seu componente interno *Construtor de Apresentação*.

Baseado nos dados recebidos, o componente *Construtor de Apresentação* irá montar o plano de apresentação. Através da intervenção do *Proxy para Informações Contextuais* (PIC na Figura 2), ele obtém os parâmetros que definem o contexto de exibição (Seção 2) e requisita ao componente *Adaptador* que implemente os ajustes necessários, como, por exemplo, escolher a melhor alternativa dentre uma coleção de objetos de mídia, ajustar a duração de apresentação dos objetos de mídia etc. Nessa fase, a consistência temporal e espacial dos relacionamentos definidos no documento pode ser verificada.

O resultado final do processamento realizado pelo *Construtor de Apresentação* é um conjunto de cadeias temporais que compõem o plano de apresentação, disponibilizando o ponto de entrada, originalmente informado pelo componente *Conversor/Usuário da QoS Inter-Objetos*, que permite iniciar a execução do documento.

Após a construção do plano de apresentação, o componente *Usuário da QoS Intra-Objetos* é chamado (na verdade seu componente interno, *Construtor de Pré-busca*, como será explicado na próxima seção). Utilizando o plano de apresentação como guia, esse componente composto solicita os serviços do meta serviço de orquestração intra-objetos, também discutido na próxima seção. Se em qualquer momento esse último meta serviço não consegue garantir a carga dos objetos do documento de acordo com o plano de apresentação, o componente *Usuário da QoS Intra-Objetos* é notificado e repassa esse aviso para o componente *Monitor de Apresentação*.

Durante a apresentação do documento, o componente *Monitor de Apresentação* observa os eventos reportados pelos exibidores do SOE do sistema hipermídia (Seção 3) e verifica se eles casam com o tempo de ocorrência previsto no plano de apresentação. O monitor também gerencia qualquer notificação proveniente do componente *Usuário da*

QoS Intra-Objetos e da gerência de contexto³ (através do componente PIC). Se for detectada uma mudança de contexto ou violação que cause impacto na apresentação, o *Monitor de Apresentação* solicita os serviços do componente *Sintonizador de Apresentação*, para que o plano de apresentação seja ajustado. Para tanto, podem ser necessários os serviços do componente *Adaptador*. É importante notar que, nesse caso, as adaptações ocorrem durante a apresentação. Em algumas situações, o componente *Escalonador da Apresentação* do SOE do sistema hipermídia pode também ser chamado para atuar sobre os exibidores de mídia, para resolver ou minimizar algum problema de sincronismo (corrigir taxas de exibição de mídias contínuas, por exemplo).

3.2 Orquestração da QoS intra-objetos

O componente *Usuário da QoS Intra-Objetos* do meta serviço de QoS inter-objetos é ao mesmo tempo cliente do meta serviço de QoS intra-objetos. A Figura 3 ilustra os detalhes desse componente composto.

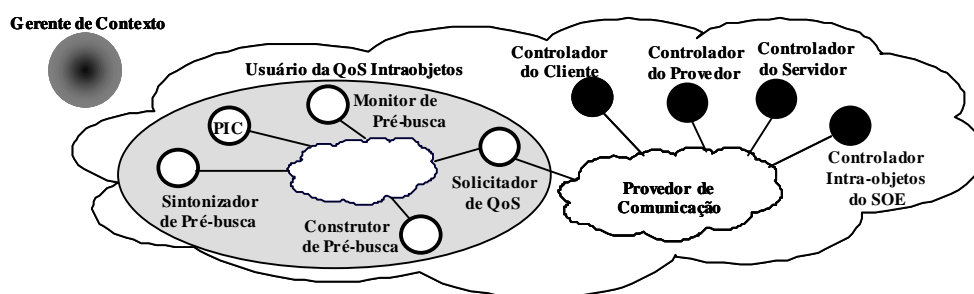


Figura 3 – Meta serviço de QoS intra-objetos

Assim que o plano de apresentação é construído, o componente *Construtor de Pré-busca* é chamado. Ele então inicia a elaboração do plano de pré-busca: uma estrutura de dados definindo quando cada objeto deve ser trazido do local de armazenamento. O plano de pré-busca pode ser baseado apenas em informações do plano de apresentação. Contudo, é importante que esse plano também leve em consideração os parâmetros de desempenho do SOE do sistema hipermídia, tais como retardos impostos pela rede, pelos sistemas operacionais, pelos próprios exibidores etc. Essas informações, quando disponíveis, devem ser obtidas através do *PIC* (Figura 3) e de mecanismos de negociação de QoS antecipada [PBL99]. Reserva de recursos antecipada vai retornar o melhor momento para efetuar a pré-busca e reservar os recursos no SOE do sistema hipermídia, para garantir que a pré-busca ocorra respeitando os parâmetros de QoS almejados. No entanto, nem os protocolos de sinalização de redes, nem os sistemas operacionais oferecem esse tipo de reserva em suas implementações atuais. Dessa forma, heurísticas precisam ser empregadas para construir o plano de pré-busca, baseando-se apenas em características de desempenho do SOE. Algumas estratégias para construção do plano são propostas na literatura [JeHK97, KhTa01].

³ A gerência de contexto é definida como um outro meta serviço no sistema, cujo papel é manter as informações a respeito do contexto de exibição, sendo ilustrada na Figura 2 como um provedor meta separado. O componente *PIC* atua como cliente do meta serviço de gerência de contexto, fazendo a ponte entre esse meta serviço e os meta serviços de orquestração de QoS.

Se o plano de pré-busca for construído fazendo uso de mecanismos convencionais de reserva de recursos (sem ser antecipada), o componente *Usuário da QoS intra-objetos* precisa iniciar uma solicitação de QoS para cada objeto, baseando-se no tempo especificado pelo plano de pré-busca. O componente *Solicitador de QoS* é responsável por desempenhar essa função. Uma solicitação de QoS é enviada ao *Controlador Intra-Objetos do SOE*, que irá implementar os mecanismos de controle de admissão, negociação e mapeamento (Figura 1). O mecanismo de negociação irá dividir a responsabilidade de prover os requisitos de QoS intra-mídia entre os recursos do SOE do sistema hipermídia (formatador, provedor de comunicação e servidor). Essa orquestração pode ser feita através de um *broker* (*Controlador Intra-Objetos do SOE* centralizado) [Durh00], como exemplificado na figura, ou através de um protocolo distribuído (*Controlador Intra-Objetos do SOE* distribuído) [Brad97].

Cada recurso (componente e provedor composto) do SOE deverá orquestrar, entre seus componentes e provedores internos, a porção de QoS a ele previamente alocada. Esse processo continua até que seja alcançado um componente, ou provedor, primitivo. Recursos de processamento e comunicação tais como CPU, memória, buffers de comunicação e canais de comunicação devem então ser reservados para garantir a porção de QoS atribuída a esses elementos. Se uma requisição de QoS não puder ser atendida, uma notificação deverá ser enviada ao componente *Monitor de Pré-busca*. Esse componente deverá também receber notificações se ocorrer qualquer violação no contrato de QoS estabelecido com o *Controlador Intra-Objetos do SOE* e ele não puder efetuar uma re-orquestração. O *Monitor de Pré-busca* pode então acionar o componente *Sintonizador de Pré-busca* para tentar uma outra configuração para o plano de pré-busca ou, se isso não for possível, notificar o problema ao meta serviço de QoS inter-objetos.

Alguns SOEs hipermídia não permitem a negociação da QoS intra-objetos em todos os seus recursos; por exemplo, um sistema hipermídia utilizando um provedor de comunicação implementado por uma rede apenas com serviço de melhor esforço. Nesse caso, cabe ao componente *Monitor de Pré-busca* monitorar os parâmetros reais de pré-busca para cada objeto. Se forem percebidas diferenças entre esses parâmetros e aqueles estimados no plano de pré-busca, o componente envia uma notificação para o *Sintonizador de Pré-busca*, que agirá de forma idêntica ao caso da violação de QoS.

Conforme discutido na Seção 3.1, o *Monitor de Apresentação* do meta serviço de QoS inter-objetos monitora os avisos vindos do meta serviço de QoS intra-objetos. Se uma alteração ou violação é percebida, ele procura fazer com que o plano de apresentação seja ajustado. A qualquer momento e por qualquer motivo, uma vez que o plano de apresentação seja modificado, uma notificação deve ser enviada ao *Monitor de Pré-busca*, que acionará o *Sintonizador de Pré-busca*, disparando o processo para adaptação do plano de pré-busca.

4. Provisão de QoS no Formatador HyperProp

Baseado na versão anterior do sistema HyperProp [SoRM00] e seguindo o framework apresentado na Seção 3, mecanismos de provisão de QoS foram incorporados à implementação do sistema HyperProp, fornecendo suporte para exibições de documentos sincronizadas e adaptativas [Rodr03].

O formatador HyperProp utiliza um modelo de execução de documentos baseado em eventos e relacionamentos entre os eventos [SoRM00, Rodr03]. Eventos podem ser

instantâneos, como o início ou o fim da apresentação de um objeto áudio, o clique de um usuário sobre uma âncora de texto etc. Por outro lado, a exibição de um conjunto de quadros de um vídeo e o arraste de objetos são exemplos de *eventos não instantâneos*. É importante observar que todo evento não instantâneo define dois eventos instantâneos, correspondendo às extremidades do intervalo de sua duração. O modelo permite que tanto relacionamentos de causalidade como relacionamentos de restrição sejam especificados entre os eventos.

O sistema possui três usuários de QoS inter-mídia implementados, seguindo as definições da Seção 3. Eles são conversores que geram, a partir de diferentes fontes de modelos hipermídia, descrições de apresentação de documento seguindo o modelo de execução do HyperProp. Os diferentes modelos contemplados são: a linguagem SMIL [W3C01], a linguagem gráfica NCM [SoRM00] e a linguagem NCL [Much03].

O *Construtor de Apresentação* do HyperProp recebe de um dos conversores a requisição da QoS inter-mídia e constrói as cadeias temporais de apresentação do documento. Na construção de cada cadeia, o *construtor* (ou *compilador*) utiliza os serviços de dois componentes adaptadores: o *Adaptador de Contexto* e o *Adaptador de Tempo Elástico*. O *Adaptador de Contexto* seleciona a estrutura do documento baseando-se em alternativas de execução e informações do contexto de exibição (Seção 2). Como na linguagem SMIL, uma alternativa de execução especifica um conjunto de objetos de mídia ou um conjunto de descritores de apresentação do qual uma opção deve ser selecionada, dependendo das características da plataforma ou do usuário (ou de ambos). Para obter os dados do contexto, o *Adaptador de Contexto* consulta o *PIC*, que, por sua vez, consulta um gerente de contexto⁴.

O *adaptador de contexto* gera uma apresentação adaptada em um primeiro nível, onde cada alternativa de execução resolvida possui uma opção selecionada. Aquelas alternativas que não podem ser resolvidas são colocadas como iniciando uma nova cadeia temporal auxiliar. Isso ocorre sempre que as regras de seleção das alternativas dependem de parâmetros dinâmicos ou parâmetros cujos valores não podem ainda ser informados pelo *PIC*/gerente de contexto. Esse fato irá levar a um ajuste tardio, que o *adaptador de contexto* tentará concluir durante a execução do documento. Mesmo para aquelas alternativas de execução resolvidas, o *construtor* grava a informação referente às regras de escolha nas cadeias temporais, permitindo que a seleção seja modificada em tempo de execução.

O segundo nível de adaptação é feito pelo componente *Adaptador de Tempos Elásticos*. O objetivo principal desse componente é calcular a duração esperada para cada evento não instantâneo com duração finita e previsível. Como consequência, o adaptador também encontra, para os eventos instantâneos não interativos, os seus tempos esperados com relação ao início da cadeia temporal a qual eles pertencem. Esse cálculo é realizado fazendo uso de um algoritmo de computação elástica dos tempos que, baseado em descrições do custo para encolher ou esticar os eventos (funções de custo), estima as durações ótimas que conferem a melhor configuração para a apresentação [BMRS00]. Na implementação atual, os custos para modificar as durações são descritos como duas funções lineares (uma para aumentar a duração e outra para diminuí-la) e a

⁴ Dependendo da informação requisitada pelo *adaptador de contexto*, o *PIC* responde diretamente, uma vez que alguns dados são mantidos em cache (principalmente as informações referentes ao cliente).

otimização é feita com base em grafos de tensão que são derivados das próprias cadeias temporais.

Uma vez concluída a compilação do documento, o *Construtor de Apresentação* do HyperProp requisita ao *Construtor de Pré-busca* que inicie sua tarefa. O *Construtor de Pré-busca* primeiro estima, para cada objeto de mídia (na verdade, cada evento não instantâneo correspondente à exibição de um objeto de mídia), o tempo de preparação do seu conteúdo. Isso é feito com heurísticas simples, utilizando informações oriundas do próprio objeto de mídia (quantidade de informação que precisa ser previamente colocada em *buffer* e a maior latência aceitável para retardar o início da apresentação do objeto) e do gerente de contexto (taxa média de transferência para recuperação do conteúdo, retardo da plataforma e, para eventos imprevisíveis, sua probabilidade de ocorrência). Em seguida, o construtor de pré-busca reúne todas essas informações e atribui o instante esperado para o início e a duração da busca de cada objeto. A implementação atual do mecanismo de pré-busca do sistema HyperProp [RoSo02] dispõe de duas estratégias de compilação que pode ser escolhida pelo próprio usuário.

Uma estratégia serializa as preparações de tal forma que cada objeto é programado para ter seu conteúdo buscado isoladamente. Se dois objetos precisam ser preparados simultaneamente, um deles é programado para ter seu conteúdo trazido primeiro, para que não haja superposição nas buscas. O plano de pré-busca é construído percorrendo o plano de apresentação de trás para frente, tendo ao final do processo uma estimativa do retardo mínimo para iniciar a apresentação que evita a ocorrência de perda no sincronismo. A outra estratégia de pré-busca utiliza uma abordagem gulosa, não importando que ocorram preparações em paralelo. Essa estratégia simplesmente subtrai do tempo esperado para iniciar a exibição de cada objeto a duração estimada para a pré-busca. Independente da estratégia, o retardo para iniciar a exibição do documento como um todo é calculado pela simples diferença entre o tempo para que o primeiro objeto seja exibido e o tempo para que o primeiro objeto seja preparado.

O término da construção do plano de pré-busca é notificado ao *Escalonador de Apresentação*, sendo passado como parâmetro o retardo total inicial estimado. Ao receber a notificação, o *Escalonador de Apresentação* pode imediatamente iniciar a fase de execução do documento. Se o usuário concordar, o *Escalonador de Apresentação* dispara a execução do *Escalonador de Pré-busca*, que é responsável por cuidar das ações programadas no plano de pré-busca, e então espera pelo retardo de iniciação estimado pelo *Construtor de Pré-busca*, para após essa espera começar a exibição do documento. O sistema também permite que o usuário especifique o retardo máximo que ele aceita aguardar, o qual pode forçar o *Escalonador de Apresentação* a iniciar a exibição mais cedo. O sistema permite ainda que o usuário controle as fases de compilação e execução separadamente, iniciando a busca e exibição em um outro momento futuro qualquer.

Sempre que o *Escalonador de Pré-busca* alcança o tempo esperado para a preparação de um objeto, ele demanda que o *Requisitante de Objetos* busque o conteúdo do objeto de mídia. No formatador HyperProp, o componente *Requisitante de Objetos* é implementado de maneira distribuída e embutida em cada *Exibidor* (Figura 1). Com isso, para cada objeto que precisa ser preparado, o *Escalonador de Pré-busca* solicita que seja instanciado um *Exibidor* apropriado (se ainda não houver um) e passa a requisição de busca diretamente ao *Exibidor*.

Atualmente, o formatador HyperProp utiliza os serviços de melhor esforço da rede e do sistema operacional. Essa abordagem obriga que todos os ajustes em tempo de execução do documento sejam feitos no nível da orquestração inter-objetos, conforme será explicado mais adiante. Como trabalho futuro, pretende-se implementar no sistema o mecanismo de negociação e manutenção da QoS intra-objetos integrado aos provedores que ofereçam reserva de recursos, o que exigirá a implementação de um *Solicitador de QoS* e de um *Monitor de Pré-busca* para o sistema.

Durante a execução do documento, o *Monitor de Apresentação* interage com os *Exibidores*, que sinalizam as ocorrências dos eventos instantâneos. O monitor possui um módulo para avaliar os relacionamentos especificados pelo autor e, sempre que um evento imprevisível ocorre (iniciando uma nova cadeia auxiliar), o monitor pede ao *Sintonizador de Apresentação* que una a cadeia recém iniciada à cadeia temporal do documento. O monitor também compara o instante sinalizado para cada ocorrência de evento com o instante previsto. Ao perceber uma diferença significativa (cada evento possui uma variação aceitável descrita na especificação de apresentação do objeto de mídia), ele também solicita providências ao *Sintonizador de Apresentação*.

Tanto a junção de cadeias como a percepção de desvios nos instantes programados para ocorrência dos eventos fazem com que o *Adaptador de Tempo Elástico* reveja a duração de cada evento não instantâneo. Os ajustes no plano de apresentação geram uma notificação para o *Sintonizador de Pré-busca*, para que as modificações sejam refletidas também no plano de pré-busca.

Embora a arquitetura do formatador HyperProp separe os algoritmos da fase de compilação dos algoritmos da fase de execução para construção e ajuste dos planos, a implementação corrente utiliza algoritmos idênticos. Como outro trabalho futuro, pretende-se testar novas estratégias mais adequadas para cada fase.

Outro tipo de adaptação em tempo de execução ocorre quando o componente *PIC* percebe uma mudança no valor de algum parâmetro do contexto de exibição. O *Adaptador de Contexto* revê as regras de exibição e todos os conseqüentes ajustes de duração e pré-busca são então disparados. Novamente, a perspectiva futura é de utilizar algoritmos de adaptação de tempo de execução diferentes dos utilizados na fase de compilação, uma vez que as restrições temporais tendem a ser muito mais rígidas durante a apresentação do documento.

A arquitetura de implementação do formatador HyperProp define todos os seus componentes (construtores/compiladores, adaptadores, escalonadores, monitores, sintonizadores e solicitador de QoS) como pontos de flexibilização do framework (*hotspots*), a fim de facilitar modificações [RoSo02]. Mais ainda, conforme mencionado anteriormente, o framework de formatação separa as estratégias da fase de compilação daquelas da fase de execução. Juntas, essas características criam mecanismos para testar e comparar diferentes algoritmos que venham a ser desenvolvidos para as várias tarefas de orquestração de apresentações hipermédia.

5. Trabalhos Relacionados

O formatador do sistema Firefly [BuZe93] trouxe várias contribuições para o framework proposto neste artigo. Dentre elas, é possível destacar: a definição de cadeias temporais para as apresentações e a separação do controle da apresentação nas fases de compilação e execução.

Uma das principais desvantagens da proposta do sistema Firefly é que o plano de apresentação é um simples *timeline* das ações de exibição, dificultando a implementação de ajustes em tempo de execução. Mais ainda, o sistema lida apenas com a adaptação dos tempos elásticos. Alternativas de objetos e ajustes em função do contexto de exibição não são considerados. A arquitetura do formatador também não prevê a existência de mecanismos de pré-busca.

O sistema hipermídia Madeus [LaSR02] também oferece flexibilidade temporal nas especificações de duração dos objetos, que podem ser definidas como intervalos. Na fase de compilação, o formatador Madeus calcula, para os objetos previsíveis (*controllable*), suas durações esperadas (*nominal duration*) e constrói o plano de apresentação através de um grafo temporal, denominado HSTP (*Hypergraph of Simple Temporal Problems*). Na fase de execução, o formatador monitora a duração efetiva de cada objeto de mídia e as compara com os valores esperados. Quando desvios que possam causar perdas de sincronização são percebidos, o formatador dispara um algoritmo de ajuste que, no entanto, não considera métricas de otimização.

O formatador Madeus não lida com questões de orquestração de pré-busca e de adaptação orientada ao contexto. Embora ofereça um algoritmo para adaptação em tempo de execução, a estratégia usa uma abordagem gulosa, sem levar em consideração a qualidade do resultado obtido.

O formatador CMIFed (chamado de *CMIFed Player*) [RJMB93] é o elemento responsável por controlar a apresentação de documentos CMIF, base para a especificação da linguagem SMIL [W3C01]. O formatador CMIFed utiliza as informações do grafo de dependências temporais, por ele construído baseado nos relacionamentos entre os objetos de mídia, para efetuar uma pré-busca dos conteúdos sempre que o escalonador da apresentação se torna ocioso. O sistema pode efetuar alguns tipos de adaptação baseada em contexto, mas nada é mencionado a respeito de ajuste das durações dos objetos. Na versão 2.0 da linguagem SMIL, a introdução de intervalos para a definição da duração dos objetos permite que seja realizado o cálculo de ajuste das durações, mas como no sistema Madeus, nenhuma métrica está disponível para medir a qualidade da apresentação adaptada.

Jeong, Ham e Kim [JeHK97] desenvolveram um mecanismo para pré-busca em apresentações multimídia. O mecanismo estima o tempo máximo para recuperar o conteúdo inteiro de cada objeto do documento e constrói um plano de pré-busca com o objetivo de ter todos os conteúdos preparados no momento de suas exibições. O algoritmo retarda o início da apresentação do documento como um todo, a fim de evitar *gaps* e perda de sincronização durante a execução. Na fase de execução, existe um monitor que observa os tempos efetivamente gastos na preparação. Se uma duração real ultrapassa o tempo previsto no plano, o sistema executa um algoritmo de ajuste para recalcular as durações dos objetos, buscando manter a sincronização inter-objetos. Quando necessário, o algoritmo sacrifica os objetos de mídia estática, encurtando suas durações. Além de compensar atrasos introduzidos pelo sistema operacional, o algoritmo permite compensar a sua própria sobrecarga de execução.

O mecanismo proposto contribui com uma implementação de estratégia de compilação para o plano de pré-busca. No entanto, o trabalho considera apenas documentos contendo exclusivamente eventos previsíveis e obriga que todo conteúdo encontre-se localmente armazenado e contíguo em disco. A estratégia sempre prepara o conteúdo

inteiro dos objetos, não prevendo a recuperação dos dados em paralelo com a sua apresentação.

Por fim, é importante mencionar que nenhuma das propostas comentadas trata da questão da negociação e manutenção da QoS intra objetos de mídia.

6. Conclusões

Este trabalho apresenta a modelagem de formataadores hipermídia focando na provisão de QoS em apresentações de documentos. O artigo descreve a organização do sistema utilizando um modelo de composição de serviços, sendo as funções de provisão de QoS divididas em dois planos: um plano contendo os componentes dedicados ao oferecimento do serviço propriamente dito (SOE hipermídia) e um outro plano reunindo os elementos envolvidos na configuração e adaptação do serviço (plano meta).

A principal contribuição deste artigo é a descrição do framework e a visão integrada da orquestração de QoS inter-objetos e intra-objetos em apresentações hipermídia. A estrutura sugerida para a inclusão de orquestração de QoS em formataadores hipermídia utiliza uma abordagem recorrente, repetindo em um nível mais alto de abstração os mesmos conceitos apresentados nas infra-estruturas tradicionais (rede e sistemas operacionais) [GoCS01, More03]. Espera-se com essa proposta de middleware facilitar o processo de negociação e manutenção de QoS em provedores de infra-estrutura e, com isso, oferecer um suporte para os projetistas de aplicações e de documentos hipermídia.

Como trabalho futuro pretende-se explorar o framework e a implementação do sistema HyperProp como uma plataforma para teste de algoritmos de formatação, tanto para tempo de compilação como para a fase de execução. Os principais tipos de algoritmos que podem ser destacados são: adaptação elástica dos tempos, adaptação baseada em contexto, compilação de plano de pré-busca, escalonamento de plano de pré-busca (com suporte do provedor para reserva de recursos convencional e antecipada) e ajustes do planos de apresentação e de pré-busca.

Atualmente está sendo desenvolvida a integração do formataador HyperProp com o sistema operacional QoSOS [More03]. Esse sistema é uma extensão ao Linux que implementa um mecanismo de controle de QoS sobre as filas de pacotes do protocolo de comunicação e também sobre o uso da CPU. O objetivo principal é implementar o meta serviço de QoS intra-mídia do sistema HyperProp como um usuário dos serviços oferecidos pelo QoSOS. Outro trabalho em andamento é a utilização da linguagem LindaQoS [More03], uma linguagem específica de domínio (DSL - *Domain-Specific Language*), buscando formalizar a descrição da provisão de QoS em formataadores hipermídia.

Referências

- [BMRS00] Bachelet B., Mahey P., Rodrigues R.F., Soares L.F.G. "Elastic Time Computation for Hypermedia Documents". *VI Simpósio Brasileiro em Sistemas Multimídia e Hipermídia*, Natal, Brasil, junho de 2000, pp. 47-62.
- [BoKW99] Boll S., Klas W., Wandel J. "A Cross-Media Adaptation Strategy for Multimedia Presentation". *ACM International Conference on Multimedia*, Orlando, EUA, 1999.
- [Brad97] Braden R. et al. "Resource ReSerVation Protocol (RSVP): Version 1 Functional Specification". *IETF Request for Comments (RFC) 2205*, setembro de 1997.

- [Brus96] Brusilovsky P. "Methods and Techniques of Adaptive Hypermedia". *Journal of User Modelling and User-Adaptive Interaction*, 6(2-3), 1996.
- [BuZe93] Buchanan M.C., Zellweger P.T. "Automatic Temporal Layout Mechanisms". *ACM International Conference on Multimedia*, Anaheim, EUA, 1993.
- [Durh00] Durham D. et al. "The COPS (Common Open Policy Service) Protocol". *IETF Request for Comments (RFC) 2748*, 2000.
- [GoCS01] Gomes A.T.A., Colcher S., Soares L.F.G. "Modeling QoS provision on Adaptable Communication Environments". *IEEE International Conference on Communications*, Helsinque, Finlândia, junho de 2001.
- [JeHK97] Jeong T., Ham J., Kim S. "A Pre-scheduling Mechanism for Multimedia Presentation Synchronization". *IEEE International Conference on Multimedia Computing and Systems*, Ottawa, Canadá, 1997, pp. 379-386.
- [KhTa01] Khan J., Tao Q. "Prefetch Scheduling for Composite Hypermedia". *IEEE International Conference on Communications*, Helsinque, Finlândia, junho de 2001.
- [KiSo95] Kim M., Song J. "Multimedia Documents with Elastic Time". *ACM International Conference on Multimedia*, São Francisco, EUA, novembro de 1995.
- [LaSR02] Layaïda N., Sabry-Ismail L., Roisin C. "Dealing with uncertain durations in synchronized multimedia presentations". *Multimedia Tools and Applications Journal*, Kluwer Academic Publishers, 18(3), dezembro de 2002. pp. 213-231.
- [More03] Moreno M.F. et al. "QoSOS: An adaptable architecture for QoS provisioning in network operating systems". *Brazilian Journal on Telecommunications*, 18(2), out de 2003.
- [Much03] Muchaluat-Saade D.C. "Relações em Linguagens de Autoria Hipermédia: Aumentando Reuso e Expressividade". *Tese de Doutorado, Departamento de Informática, PUC-Rio*, Rio de Janeiro, Brasil, março de 2003.
- [PBLS99] Pajares A., Berier N., Lars W., Steinmetz R. "An Approach to Support Mobile QoS in an Inegrated Services Packet Network". *I Workshop on IP Quality of Service for Wireless and Mobile Networks (IQWiM '99)*, Aachen, Alemanha, 1999.
- [RJMB93] van Rossum G., Jansen J., Mullender K.S., Buterman D. "CMIFed: A Presentation Environment for Portable Hypermedia Documents". *ACM International Conference on Multimedia*, Anaheim, EUA, agosto de 1993.
- [Rodr03] Rodrigues R.F. "Formatação e Controle de Apresentações Hipermédia com Mecanismos de Adaptação Temporal". *Tese de Doutorado, Departamento de Informática, PUC-Rio*, Rio de Janeiro, Brasil, março de 2003.
- [RoSo02] Rodrigues R.F., Soares L.F.G. "A Framework for Prefetching Mechanisms in Hypermedia Presentations". *IEEE 4th International Symposium on Multimedia Software Engineering*, Newport Beach, EUA, dezembro de 2002.
- [ScAW94] Schilit B., Adams N., Want R. "Context-aware computing applications". *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, EUA, 1994.
- [SoRM00] Soares L.F.G., Rodrigues R.F., Muchaluat-Saade D.C. "Modeling, Authoring and Formatting Hypermedia Documents in the HyperProp System". *ACM Multimedia Systems Journal*, Springer-Verlag, 8(2), 2000, pp. 118-134.
- [Stey94] Steyaert P. "Open design of object-oriented languages". *Tese de Doutorado, Vrije Universiteit Brussel*, 1994.
- [W3C01] World-Wide Web Consortium. "Synchronized Multimedia Integration Language (SMIL 2.0) Specification". *W3C Recommendation*, agosto de 2001.