Arquitetura do Sistema HyperProp*

SÉRGIO COLCHER[1]
colcher@inf.puc-rio.br

LUIZ F. G. SOARES[1]
lfgs@inf.puc-rio.br

MARCO A. CASANOVA[2]
casanova@vnet.ibm.com

[1]Departmento de Informática
PUC-Rio
R. Marquês de São Vicente, 225
22453 Rio de Janeiro RJ - Brasil

[2]Centro Científico Rio
IBM Brasil
Av. Presidente Vargas, 824/844
20071 Rio de Janeiro RJ - Brasil

Resumo. No presente trabalho, apresenta-se uma revisão dos conceitos e tecnologia existentes para a implementação de sistemas hipermídia, ao longo da qual, consolidamos as noções modelo conceitual, modelo de apresentação, arquitetura, protocolo, entre outros. Requisitos para a construção de sistemas hipermídia, como os relacionados a sincronização, interatividade em tempo real, qualidade de serviço e distribuição de informação são igualmente apontados. O conjunto de conceitos e requisitos básicos sugere a existência de um conjunto de funções essenciais em sistemas hipermídia, cuja organização é apresentada sob a forma de uma arquitetura genérica. Tal arquitetura vem sendo utilizada no desenvolvimento de uma plataforma básica orientada a objetos para a construção de sistemas hipermídia: o HyperProp, que leva em consideração vários dos recentes trabalhos de desenvolvimento, pesquisa e padronização, entre eles (i) o padrão ISO/MHEG [1], que define a representação para intercâmbio de objetos multimídia/hipermídia, e (ii) o padrão CORBA [2], que define modelo e arquitetura para sistemas orientados a objetos distribuídos.

1 Introdução

Ao tentar responder a célebre pergunta formulada por Halasz e Schwartz em seu trabalho sobre o modelo Dexter [3] a respeito do que existiria de comum nos sistemas hipermídia então existentes, a primeira idéia é associar tais sistemas à presença de uma estrutura baseada em nós interconectados por elos. Historicamente, esse tipo de *modelagem* tornou-se comum. Mas, de um ponto de vista abstrato, a idéia de hiperdocumento pode ser inicialmente expressa de uma maneira mais diretamente relacionada a forma com que Vannevar Bush [4] descreveu seu dispositivo "Memex" ainda em 1945 (vários anos antes de Ted Nelson [5] ter cunhado o termo *hipertexto*):

*Este trabalho foi desenvolvido com o apoio do CNPq através do projeto ProTem-HyperProp. Publicado no I Workshop em Sistemas Hipermídia Distribuídos, 7–8 de Julho de 1995, São Carlos, SP.

> "The human mind (...) operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts...".

A *idéia de hiperdocumento* surge como uma conseqüência direta da observação de Bush sobre a não linearidade com que seres humanos tratam, adqüirem ou geram conhecimento. Desde então, *modelos* tem sido criados de forma a permitir capturar com maior precisão as idéias contidas nesses sistemas. Tais modelos, muitas vezes citados como modelos conceituais, costumam descrever sistemas hipermídia como sendo sistemas capazes de tratar, construir, exibir e processar hiperdocumentos; hiperdocumentos por sua vez, são geralmente modelados através da definição de fragmentos de informação denominados *nós*, interconectados por *elos*. Elos representam a associação de idéias existente entre os fragmentos formulada pelo autor ou autores do documento. Através da marcação de *âncoras*, perceptíveis ao usuário através de regiões sensíveis a seleção, é possível navegar através do documento de forma não seqüencial, segundo o interesse particular do leitor e de acordo com alguma associação feita previamente por autores. A maioria dos modelos descritos até hoje faz uso desse tipo de conceito, elaborando-o e utilizando-o na construção de modelos mais complexos que abordam outros aspectos além daqueles relacionados a estrutura dos hiperdocumentos.

Através de modelos, pode-se reconhecer as funções necessárias para a implementação de um sistema. A variedade de sistemas hipermídia disponíveis oferece um campo de observação suficiente para que se estabeleça um conjunto de abstrações inicial bastante abrangente. Adicionando-se outras abstrações derivadas de críticas, requisitos ainda não tratados ou de novos requisitos, pode-se criar um modelo suficientemente genérico para capturar o comportamento de qualquer sistema existente. Mais ainda, torna-se possível definir um conjunto de funções básicas essenciais que deve estar presente em qualquer sistema hipermídia que deseje apresentar todas as características definidas no modelo. A organização dessas funções através da sua classificação e divisão em módulos que se comunicam, oferecendo e recebendo mutuamente serviços é o que denominamos *arquitetura do sistema*.

No presente trabalho, apresenta-se uma arquitetura genérica para sistemas hipermídia com base no conjunto funções reconhecidas como essenciais nesses sistemas, adicionadas de funções provenientes de requisitos de tempo real, interatividade e qualidade de serviço. O conjunto de funções essenciais é estabelecido a partir da definição de um modelo conceitual genérico, tratado na seção 3, logo após a caracterização do papel que modelos desempenham na definição de sistemas hipermídia, na seção 2. Os requisitos mencionados são apresentados na seção 4.

O paradigma da orientação por objetos tem sido aplicado com sucesso tanto na definição conceitual das entidades tratadas em modelos, como nós e elos, como na implementação de sistemas hipermídia, conforme apresentamos na seção 5. Com efeito, a definição de interfaces bem definidas facilita a implementação desses sistemas, a continuidade da informação e o compartilhamento de dados. Em particular, a arquitetura proposta tem uma preocupação especial com aspecto de intercâmbio de informação dado que o investimento em criação e geração de informação tem se tornado cada vez mais alto, tornando ainda mais atrativo o compartilhamento e a reutilização. A seção 6 [1] trata do padrão MHEG, que define a estrutura e a representação dos objetos hipermídia para intercâmbio.

A seção 7 é dedicada a apresentação da arquitetura genérica para sistemas hipermídia que vem sendo utilizada como base para a construção do HyperProp. Trata-se de uma plataforma básica que permite o compartilhamento de informações seguindo as premissas básicas para a construção de sistemas abertos, como mencionamos na seção 4, apresentando aos usuários ou aplicações um ambiente distribuído orientado a objetos nos moldes definidos pelo padrão CORBA [2]. Trata-se de uma evolução do trabalho apresentado em [6].

Por fim, a seção 8 concentra-se em mostrar as contribuições, apontar os trabalhos que se encontram atualmente em desenvolvimento e sugerir os trabalhos que ainda planejamos dentro da continuação do projeto do HyperProp.

2 Modelos

Do ponto de vista de ciência da computação, a descrição da noção de hiperdocumento é um processo de modelagem ou criação de uma abstração a partir da idéia ou comportamento original. Tais abstrações ou modelos procuram capturar os aspectos relativos ao comportamento do sistema de uma forma independente de sua realização. As entidades descritas em tais modelos não necessariamente existem numa determinada implementação. A idéia, sim, é o núcleo que deve ser capturado. Por isso, tais modelos são comumente denominados *modelos conceituais*. A definição de um modelo conceitual para sistemas hipermídia abrange vários aspectos além dos aspectos puramente estruturais dando origem, muitas vezes, a diferentes modelos que juntos compõem o modelo conceitual como um todo. Alguns tipos de modelos comumente citados incluem: modelos estruturais, modelos de apresentação, modelos de versões, modelos de navegação e modelos de autoria.

Modelos estruturais são os modelos que definem a estrutura do hiperdocumento, utilizando, em geral, as noções de nós e elos. Modelos estruturais são essencialmente estáticos, pois dizem respeito tão somente às estruturas de dados cujo objetivo é conter as informações necessárias para o correto funcionamento do sistema.

Modelos de apresentação são os modelos que, ao contrário dos modelos estruturais, procuram capturar a *dinâmica* do funcionamento dos sistemas hipermídia. Modelos de apresentação descrevem o funcionamento ou processo que rege o sistema definindo, por exemplo, eventos como o disparo de âncoras, o comportamento do sistema quando eventos ocorrem, ou conceitos como o de apresentação sincronizada de nós. O modelo de apresentação costuma utilizar características definidas pelo modelo estrutural, estando ambos intimamente relacionados. Nesse relacionamento, modelos de apresentação definem o comportamento do sistema do ponto de vista da recuperação e apresentação adequada dos nós do hiperdocumento, dando semântica aos relacionamentos definidos pelos nós e elos do modelo estrutural.

Modelos de versões definem mecanismos que permitem facilidades para trabalho cooperativo, exploração de configurações alternativas, manutenção de histórico de modificações em documentos, representações distintas de uma mesma informação etc., conforme pode-se encontrar detalhadamente em [7].

Modelos de navegação estabelecem, conceitualmente, as forma com que o usuário pode navegar através do documento, incluindo a definição dos mecanismos de navegação por elos e navegação em profundidade, navegação por trilhas ou pelo histórico de um documento.

Modelos de autoria se prestam a criação de formas de organização que facilitem o trabalho dos autores de hiperdocumentos, permitindo a criação de documentos mais bem organizados.

Cada sistema hipermídia particular define suas próprias características ao estabelecer as abstrações relativas a cada um dos modelos que acabamos de descrever. Tal descrição, porém, nem sempre segue a divisão ou classificação nos modelos que apresentamos, sendo não pouco comum a utilização de uma única descrição monolítica que engloba todos os aspectos mencionados. No entanto, a subdivisão do modelo conceitual em outros modelos mais simples permite o melhor domínio da complexidade desses sistemas.

3 Um Modelo Genérico de Referência para Sistemas Hipermídia

Ao definir uma arquitetura genérica para sistemas hipermídia, procuramos levar em consideração as características encontradas nos sistemas hipermídia disponíveis. Apesar de bastante ambiciosa, a idéia é propor um modelo suficientemente abrangente para modelar o comportamento dos sistemas existentes e ainda ser suficientemente genérico para acomodar as abstrações de quaisquer novos sistemas que venham a surgir. Definiremos tal modelo como um modelo de referência, no sentido de que as funcionalidades que podem ser encontradas em sistema hipermídia deverão estar, de alguma forma, representadas. Sistemas reais, na prática, deverão, na sua maioria, conter apenas um subconjunto das características que vamos apresentar. A idéia de compatibilidade entre sistemas pode ser explorada ao definir possíveis *perfis* para os sistemas, que nada mais são do que subconjuntos das características dadas pelo modelo.

Para capturar os conceitos de um número grande de sistemas nada melhor do que partir do modelo Dexter [3], cujo objetivo é servir de referência para nomeclatura e modelagem de sistemas hipermídia. O modelo Dexter é capaz de modelar, de forma genérica, as abstrações encontradas em vários sistemas existentes. A partir desse modelo, durante o andamento do projeto HyperProp, definiu-se um novo modelo, ainda mais abrangente, capaz de englobar os conceitos definidos pelo modelo Dexter e adicionar novos conceitos: o *Modelo de Contextos Aninhados* — MCA [8]. Nessa seção, procuramos apresentar os principais conceitos necessários para a construção de um sistema hipermídia presentes no MCA. Nos limitaremos a apresentação apenas dos conceitos que influenciam diretamente nos mecanismos funcionais e que, por conseguinte, terão reflexo direto na definição da arquitetura. Abstrações do modelo que refletem outros aspectos como os de orientação do usuário, identificação de fragmentos e âncoras etc. podem ser encontrados em [8, 9, 7].

Como já mencionamos, a subdivisão do modelo conceitual em outros modelos mais simples facilita tanto a definição quanto a apreensão dos conceitos. No modelo Dexter, por exemplo, utiliza-se o conceito de divisão em camadas,[1] como ilustrado na Figura 1.

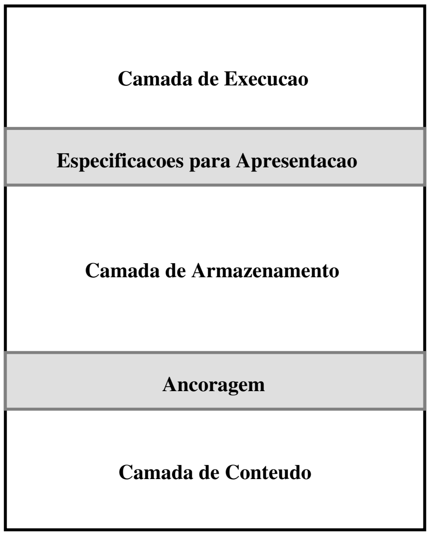

Figura 1: O Modelo Dexter

O modelo Dexter particiona um sistema hipermídia em três camadas: a *camada de execução*, a *camada de armazenamento* e a *camada de conteúdo*. O foco principal do modelo concentra-se na camada de armazenamento, onde modela-se a estrutura básica de nós e elos que constitui-se na essência do modelo estrutural de hiperdocumentos.

A camada de armazenamento descreve a base de dados composta de *componentes* que podem estar organizados de forma hierárquica e ao mesmo tempo conectados através de *elos*. Componentes podem ser *atômicos* ou *compostos*. *Componentes atômicos* são tratados pela camada de armazenamento como repositórios de dados genéricos, cuja estrutura interna relativa ao conteúdo propriamente dito não é modelada. Em outras palavras, um componente atômico é um repositório de informações para qualquer tipo de dados ou documentos cuja estrutura interna não é objeto de definição do modelo de hiperdocumento; exemplos são informações das diversas mídias como texto, gráfico, etc., e informações estruturadas segundo outros modelos que não o modelo de hiperdocumento proposto (documentos ODA, HyperODA, GDID, etc.).

[1] Não confundir com o conceito de *arquitetura em camadas* ou *em níveis* que trataremos mais adiante.

Componentes compostos permitem a organização hierárquica da estrutura dos hiperdocumentos, sendo definidos como um conjunto de outros componentes (atômicos ou compostos) interligados por elos.

Já a *camada de conteúdo* trata especificamente da estrutura interna dos componentes atômicos. Assume-se que vários modelos podem conviver para que se modele os diferentes tipos de informação tanto das mídias em geral (com os vários padrões de codificação existentes como ASCII e EBCDIC para texto, JPEG para imagem, MPEG para vídeo, etc.) como para documentos estruturados (com os padrões para organização de documentos como ODA, HyperODA, IGES, etc.).

Uma parte crítica do modelo Dexter é a interface entre as camadas de conteúdo e de armazenamento. Sistemas hipermídia necessitam de mecanismos que permitam referenciar regiões, itens ou estruturas no conteúdo dos componentes. Tal mecanismo é conhecido como *mecanismo de ancoragem*. Ao se definir o mecanismo de ancoragem, deseja-se preservar a fronteira entre a camada de armazenamento e a camada de conteúdo, o que requer a total transparência sobre a estrutura interna do conteúdo dos componentes atômicos. Para tanto, o modelo Dexter define uma forma de indireção para as *âncoras* definidas em componentes atômicos do modelo. Maiores detalhes podem ser encontrados em [3].

As camadas de armazenamento e conteúdo tratam dos componentes de um hiperdocumento como estruturas essencialmente passivas. A funcionalidade associada às ferramentas de acesso, visualização, manipulação ou edição da estrutura e conteúdo de hiperdocucumentos é capturada pela *camada de execução* do modelo. Assim como no caso da camada de conteúdo, a camada de execução engloba um conjunto vasto de opções que correspondem, basicamente, à especificação de modelos de apresentação.

Assim como no caso do mecanismo de ancoragem, um aspecto crítico na definição do modelo Dexter é a interface entre as camadas de armazenamento e execução. Ao definir tal fronteira, observa-se, na figura 1, a presença das *especificações de apresentação*. Estas especificações consistem em algum tipo de informação sobre a forma através da qual um componente qualquer do hiperdocumento deve ser apresentado. O modelo define que tais especificações podem então ser armazenadas em conjunto com os respectivos componentes, muito embora a interpretação destas especificações fique toda a cargo da camada de execução no momento da apresentação destes componentes. Assim, a forma com que um componente é apresentado ao usuário pode variar de acordo não só com as ferramentas de exibição utilizadas na camada de execução, mas também pode ser uma propriedade do próprio componente ou do caminho (navegação) tomado até atingí-lo. Um exemplo da importância das especificações de apresentação pode ser encontrado em [3, 8].

O conceito fundamental para a camada de execução é a *instanciação* de um componente. A idéia de instanciação tem como objetivo capturar e descrever a apresentação de um componente durante a execuçao de uma sessão do sistema. No momento da apresentação, uma instanciação do componente é criada, correspondendo a uma espécie de cópia ou "personificação" desse componente, e, caso alterada, passa a provocar atualizações ou "escritas" no componente original. Note que pode-se ter mais de uma instanciação para um mesmo componente simultaneamente. Quando (por ação do usuário, por exemplo) um componente não está mais sendo apresentado numa sessão, significa que ele passou pelo processo de "desapresentação", que corresponde a destruição da instanciação daquele componente naquela sessão.

4 Princípios para Construção da Arquitetura

Uma arquitetura para sistemas sistemas hipermídia deve implementar o comportamento definido pelos modelos que apresentamos na seção 2. Alguns requisitos relativos a construção desses sistemas podem ser acrescentadas:

1. a arquitetura deve ser *aberta*, o que corresponde a presença de dois fatores fundamentais:
   (a) abertura em relação à conectividade, significando a adoção de padrões internacionais para definição os elementos de dados intercambiáveis e para os protocolos utilizados, permitindo a interoperabilidade e disponibilidade de informação entre sistemas;
   (b) abertura em relação à estrutura, o que implica na presença de interfaces bem definidas que correspondem a diferentes níveis de serviço na arquitetura, cujo objetivo principal é permitir: (i) que sistemas possam ser adequadamente projetados de forma a serem portáveis para várias plataformas com um número mínimo de modificações, e (ii) máximo reuso de código para várias aplicações;
2. tratamento adequado a dados multimídia e hipermídia de forma a manter a interatividade, características de tempo real, qualidade de serviço e sincronização entre fragmentos de informação quando assim especificado;
3. a arquitetura deve levar em consideração a sua implementação em ambientes distribuídos, que além de permitir o compartilhamento de informações numa comunidade, torna-se necessária para fornecer:
   (a) as características de tempo de real e qualidade de serviço necessárias, dada a grande capacidade de armazenamento e banda passante requisitadas pela informação manipulada;
   (b) serviços de comunicação multimídia interativos como os de teleconferência, videofonia, TV interativa etc.

Da experiência obtida no projeto de redes, vários princípios de organização surgiram, dentre os quais destaca-se a idéia de estruturar as funções de comunicação como um conjunto de camadas hierárquicas, cada uma sendo construída utilizando as funções e serviços oferecidos pelas camadas inferiores [10].

Na camada de aplicação, utilizamos os conceitos de organização conforme os definidos no modelo OSI [11]. Dentro de um sistema OSI, um *processo de aplicação* é uma representação abstrata de um conjunto de recursos que podem ser usados para executar uma atividade de processamento de dados específica. A partes de um processo de aplicação responsáveis pelos aspectos relacionados diretamente a comunicação são denominadas *entidades de aplicação* (application entities – AEs). Um processo de aplicação pode ter várias entidades de aplicação, cada uma responsável por um determinado aspecto de comunicação; uma entidade de aplicação, porém, reside em um e somente um processo de aplicação. Entidades de aplicação são compostas de *elementos do serviço de aplicação* (Application Service Elements – ASEs), que nada mais são do que a implementação das funções básicas de comunicação. Além das entidades de aplicação, cada processo tem uma parte de módulos específicos responsáveis por implementar as funções de processamento desejadas e fazer acessos aos serviços de aplicação OSI, através dos ASEs.

ASEs podem ser classificados em comuns ou específicos. *ASEs específicos* são aqueles criados para atender a necessidades próprias de uma aplicação ou conjunto de aplicações particular. *ASEs comuns* são aqueles geralmente utilizados por quase todas as aplicações e, por essa razão, definidos por padrões específicos da ISO. Exemplos são o ACSE, RTSE, ROSE e o CCR. Descrições detalhadas desses elementos podem ser encontardas em [10, 12].

O elemento do serviço de aplicação denominado ROSE (*Remote Operation Service Element*) está particularmente relacionado com vários aspectos propostas no presente trabalho. O ROSE foi elaborado para facilitar o desenvolvimento de aplicações distribuídas através do oferecimento de um serviço que permite a solicitação de execução de uma operação em um sistema remoto. O modelo de aplicações distribuídas, segundo a perspectiva do ROSE, é baseado no conceito de tipo abstrato de dados.

5 Orientação por Objetos

Através da extensão do conceito de tipos abstratos de pode-se chegar na abstração de entidades denominadas objetos, que constituem o elemento básico do paradigma de orientação por objetos. *Objetos* podem ser encarados como como entidades que agem como servidores, fornecendo uma interface aos seus serviços que esconde detalhes de implementação dos clientes. Clientes podem ter acesso aos serviços de um objeto enviando a ele *solicitações* que especificam a operação desejada (juntamente com eventuais parâmetros necessários). Todos os aspectos de implementação não diretamente refletidos na interface de um objeto permanecem desconhecidos para os clientes, definindo-se o conceito de *encapsulação* que, juntamente com o conceito de herança forma a base da tecnologia de orientação por objetos [13].

Mecanismos de *herança* permitem a reutilização da definição de uma classe (denominada *superclasse*) por outras classes (denominadas suas *subclasses*). Subclasses podem ser utilizadas para especializar uma classe de forma que as propriedades ou descrições válidas para a superclasse são encontradas também nas suas subclasses sem, contudo, a necessidade de redefiní-las.

Objetos de dados definem os elementos de intercâmbio e manipulação das aplicações. A definição desses elementos é tomada da descrição dos objetos MHEG, donde distinguem-se dois tipos de elementos:

- *atributos de descrição*, que contém informações sobre o objeto, seu tipo etc., e
- *conteúdo*, que pode variar de acordo com o tipo do objeto. No caso de nós terminais, por exemplo, o conteúdo poderá corresponder a um fragmento de uma determinada mídia. Já em um nó de contexto, será o conjunto de nós e elos que o compõe.

Tal separação torna-se importante quando observamos que, do ponto de vista da manipulação, recuperação e armazenamento, os requisitos de qualidade de serviço e sincronização estarão coerentemente definidos apenas sobre o conteúdo do objeto.

6 Intercâmbio de Objetos Hipermídia

O padrão MHEG [1] define a estrutura e representação de entidades de informação em sua forma final para intercâmbio, seja em tempo real ou não, entre aplicações ou usuários. A especificação dessas unidades, de agora em diante denominadas *MH_objects*, segue a estratégia de definição de classes utilizando o conceito de herança proveniente do paradigma de orientação por objetos. Classes são definidas, correspondendo a definição das estruturas de atributos dos objetos intercabiáveis. A partir da definição de classes básicas, subclasses são definidas correspondendo a especializações geradas por herança, conforme descrevemos na seção 5.

O principal foco do MHEG é fornecer a *sintaxe abstrata* de suas classes, descrevendo-as através dos atributos intercambiáveis de maneira independente da codificação utilizada em aplicações, máquinas ou linguagens. A forma utilizada para a descrição destas classes utiliza uma notação padronizada sugerida pela própria ISO denominada *Abstract Syntax Notation One* — ASN.1 [14], porém, futuramente, segundo própria sugestão contida no texto do padrão, outras especificações em notações alternativas podem ser fornecidas. Em particular, é intenção do MHEG fornecer uma notação alternativa para os mesmos tipos definidos em ASN.1, utilizando-se SGML. Na prática, qualquer linguagem ou notação capaz de expressar tipos de dados de forma abstrata (independente de codificação) poderá ser futuramente sugerida.

Independente da notação utilizada para especificar a sintaxe abstrata de um conjunto de tipos, várias representações concretas podem ser obtidas para suas instâncias, correspondendo a codificações dos valores de seus atributos de forma única. Uma representação concreta utilizada para intercâmbio deverá permitir o correto reconhecimento do tipo de dados ao qual a instância sendo trocada pertence, bem como a correta interpretação dos valores de seus atributos. Dada uma sintaxe abstrata para um conjunto de tipos, é possível definir uma representação concreta, correspondendo a uma codificação para as instancias desse tipo. A sintaxe utilizada numa representação concreta com finalidade de intercâmbio de informação é denominada *sintaxe de transferência*. Uma sintaxe de tranferência deve definir, para cada tipo de dados especificado pela sintaxe abstrata, de que forma uma instância daquele tipo deve ser codificada no momento da troca de informação. No escopo do modelo OSI, a especificação dessa sintaxe é feita de forma genérica, segundo um conjunto de regras definido de forma a permitir a codificação de qualquer instância de um tipo de dados qualquer especificado em ASN.1. Esse conjunto de regras é fornecido também em um padrão internacional denominado *ASN.1 Basic Encoding Rules* — ASN.1 BER [15].

7 Arquitetura Aberta para Sistemas Hipermídia

A arquitetura proposta para sistemas hipermídia é uma arquitetura genérica que permite acomodar o conjunto de funções capturado pelo modelo conceitual mencionado na seção 2, apresentando ainda as funções e estrutura necessárias para garantir as características e requisitos levantados na seção 4. Essa arquitetura vem sendo utilizada na implementação do sistema Hyperprop e pode ser visualizada na Figura 2.

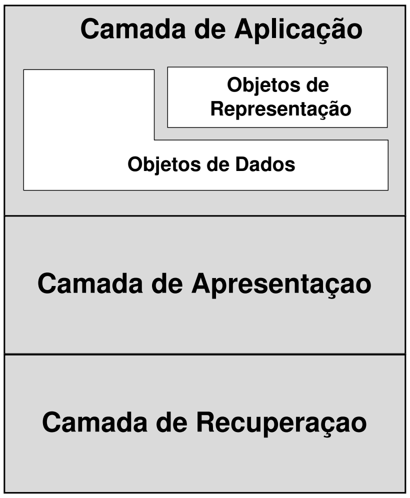

Figura 2: Arquitetura Genérica para Sistemas Hipermídia

A estrutura proposta segue os mesmos princípios, conceitos e parte das definições encontradas no modelo OSI, incluindo os conceitos de camadas, protocolos e interfaces.

A camada de recuperação engloba as funcionalidades encontradas nas camadas 1 a 5 do modelo OSI acrescidas de características que permitam a troca de informação em tempo real levando em consideração os requisitos de qualidade de serviço e sincronização. Extensões necessárias ao modelo OSI referente a essas camadas que incluem essas características têm sido propostas. Exemplos podem ser encontrados em [16, 17]. Em particular, a tecnologia relativa a sub-rede de comunicação (camadas

1, 2 e 3) tem sido alvo de pesquisa no intuito de garantir a qualidade de serviço em conexões estabelecidas por camadas superiores. Redes de alta velocidade como as redes ATM têm sido citadas como grandes candidatas nessa função [10]. Acima dessas camadas, ainda dentro do que chamamos genericamente de camada de recuperação, novas camadas responsáveis pela interconexão de redes e pelo controle fim a fim da transferência de informação podem ser colocadas e deverão, além de suas funções tradicionais, se encarregar da *orquestração* das informações trocadas pelas conexões estabelecidas na sub-rede de forma a manter a sincronização desejada.

A principal função do nível de apresentação, segundo o modelo OSI, é compatibilizar a forma com são representadas as informações intercambiadas pelas entidades de aplicação. O nível de apresentação deverá, com o conhecimento dos tipos de dados manipulados por essas entidades, realizar a conversão entre a representação interna utilizada e a representação utilizada para o intercâmbio (definida pela sintaxe de transferência). Em particular, na arquitetura apresentada, os tipos de dados são definidos pelo padrão MHEG.

O principal conceito envolvido na definição do objetos de dados é o da construção de um repositório virtual. A função do *repositório virtual* é fornecer a abstração da existência de uma base de dados cujo acesso pelas aplicações seja direto e transparente a questões como: forma ou tipo de armazenamento, unidade de armazenamento, localização dos objetos, estratégias de armazenamento ou recuperação.

A configuração utilizada na implementação dos objetos de dados em ambientes distribuídos permite várias opções como, por exemplo, a definição de interações do tipo cliente-servidor que ilustramos na Figura 3. Nessa configuração, os objetos de dados no sistema servidor atuam como elementos capazes de fornecer armazenamento para as unidades de informação, enquanto que os objetos de dados clientes se prestam a função de provedores de acesso aos objetos armazenados de maneira a criar a abstração de repositório virtual.

Numa configuração que apresente uma distribuição mais uniforme, cada sistema poderá atuar tanto como servidor ou cliente de outros sistemas, caso em que alguns objetos de dados servirão de provedores de acesso a sistemas remotos enquanto outros proverão acesso diretamente a um sistema de armazenamento local. Para representar de forma única qualquer opção de configuração para recuperação e armazenamento de objetos, utilizamos uma representação simplificada da arquitetura, que omite os detalhes de distribuição e comunicação entre sistemas, conforme apresentamos na Figura 4

Do ponto de vista da organização dos objetos, a arquitetura proposta define o seguinte mecanismo. Objetos podem ser recuperados ou armazenados local ou remotamente. Cada objeto de dados torna transparente todos os detalhes da implementação dos mecanismos de recuperação ou armazenamento. Os serviços que, genericamente, podem ser obtidos da camada de armazenamento são os de recuperação e armazenamento de todo ou parte dos objetos de acordo com parâmetros de qualidade de serviço especificados por uma aplicação. Lembrando que a camada de armazenamento é uma camada genérica que pode encorporar tanto a funcionalidade de armazenamento local

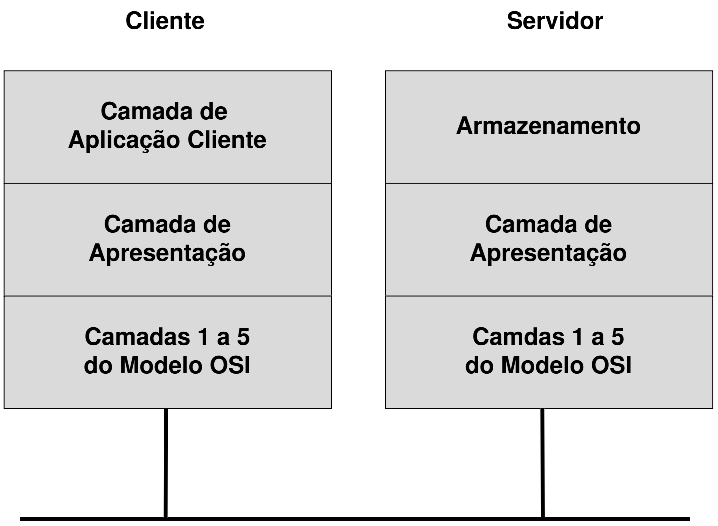

Figura 3: Implementação em Arquitetura Cliente-Servidor

quanto a das camadas 1 a 5 do modelo OSI, a definição desses serviços poderá variar de acordo com a configuração. Para usuários dos objetos de dados, no entanto, é indiferente se cada objeto em particular está sendo armazenado e recuperado localmente ou não. Objetos de representação são especializações dos objetos de dados que permitem sua apresentação e manipulação através da adição de métodos que correspondem a utilização de exibidores, editores, ferramentas etc. disponíveis no ambiente da aplicação. Sempre que um objeto é apresentado, um objeto de representação é criado e o respectivo método de exibição é disparado. Ao visitar e apresentar nós da base pública, versões são criadas na base privada, correspondendo a criação de objetos de representação.

Figura 4: Arquitetura Simplificada para Armazenamento

8 Conclusões

Referências

[1] ISO/IEC JTC1/SC29/WG12 Multimedia and Hypermedia Information coding Expert Group (MHEG). *ISO/IEC DIS 13522-1 — Coded Representation of Multimedia and Hypermedia Information (MHEG), Part I: Base Notation (ASN.1)*, october 1994.

[2] Object Management Group (OMG). *The Common Object Request Broker: Architecture and Specification*, 1991. Document number 91.12.1.

[3] F. G. Halasz and M. Schwartz. The dexter hypertext reference model. In *NIST Hypertext Standardization Workshop*, Gaithersburg, january 1990.

[4] V. Bush. As we may think. *Atlantic Monthly*, 176(1):101–108, july 1945.

[5] T. H. Nelson. Getting it out of our system. In G. Schechter, editor, *Information Retrieval: A Critical Review*, pages 191–210. Thompson Books, Washington, D. C., 1967.

[6] L. F. G. Soares, M. A Casanova, and S. Colcher. An architecture for hypermedia systems using MHEG standard objects interchange. *Information Services & Use*, 13(2):131–139, 1993. Special Issue: Hypermedia and Hypertext Standards.

[7] L. F. G. Soares, M. A. Casanova, and Noemi deLaRocque Rodriguez. Nested Composite Nodes and Version Control in a Open Hypermedia System. *Information Systems Journal*, 1995. Aceito para publicação.

[8] M. A. Casanova, L. Tucherman, M. J. Lima, J. L. Rangel Netto, N. de La Roque Rodriguez, and L. F. G. Soares. The nested context model for hyperdocuments. In *Proceedings of Hypertext'91*, Texas, december 1991.

[9] M. A Casanova and L. F. G Soares. Versions in the nested context hypermedia model. Technical report, Departamento de Informática PUC-Rio, august 1992.

[10] L.F.G. Soares, G. Lemos, and S. Colcher. *Redes de Computadores: das LANs, MANs e WANs às Redes ATM.* Ed. Campus, 1995. Segunda Edição.

[11] ISO/IEC 9545. *Information Processing Systems – Open Systems Interconnection – Application Layer Structure.*

[12] M. T. Rose. *The Open Book: A Practical Perspective on OSI.* Prentice Hall, 1989.

[13] B. Meyer. *Object-oriented software construction.* Prentice Hall, 1988.

[14] ISO/IEC 8824 : 1990. *Information technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).*

[15] ISO/IEC 8825 : 1990. *Information technology — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.

[16] D. Shepherd and M. Salmony. Extending OSI to support synchronization required by multimedia applications. *Computer Communications*, 13(7):399–406, september 1990.

[17] G. Coulson. *Multimedia Application Support in Open Distributed Systems*. PhD thesis, Computer Department — Lancaster University, april 1993.