UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Tecnologias para Adaptação em Hipermídia

Alfredo Lanari de Aragão
André Carlos Ponce Leon Ferreira de Carvalho

Nº 201

RELATÓRIOS TÉCNICOS

São Carlos – SP
Jul./2003

Tecnologias para Adaptação em Hipermídia

Alfredo Lanari de Aragão

André Carlos Ponce de Leon Ferreira de Carvalho

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 – São Carlos, SP, Brasil

alanari@ec.ucdb.br, andre@icmc.usp.br

JULHO 2003

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Tecnologias para Adaptação em Hipermídia

O desenvolvimento tecnológico aumenta significativamente a capacidade e a velocidade de processamento dos computadores, permitindo que recursos avançados de *hardware* sejam explorados por *softwares* cada vez mais complexos. Basta considerar o avanço das interfaces entre o usuário e o computador, desde o seu surgimento até os dias de hoje.

Observa-se, também, a necessidade que o ser humano possui, atualmente, de encontrar informações importantes e significativas em pouco tempo. Com isto, o homem é capaz de tomar decisões e de aumentar a sua eficiência, em especial no ambiente de trabalho. Além disto, os computadores oferecem um conjunto de elementos e uma mídia capaz de armazenar, processar e transmitir informações com alto desempenho.

Todos estes fatores, juntos, contribuiram para a popularização do uso destas máquinas em praticamente todos os segmentos da atividade humana. Considerando o panorama atual de desenvolvimento da Computação e as necessidades humanas de nossa época, pode-se dizer que não é mais suficiente ter-se sistemas altamente interativos. É preciso que a informação buscada esteja de acordo com os interesses e objetivos do usuário.

Para tanto, é necessário o desenvolvimento e o uso de tecnologias capazes de fazer com que os computadores ofereçam suporte para adaptações automáticas. Isto pode ser feito de várias maneiras, como sugerem as técnicas Modelagem de Usuários

(MU) e de Inteligência Artificial (IA).

Na Seção 1.1 aparecem as motivações para o desenvolvimento da Hipermídia Adaptativa (HA). Uma classificação baseada na interação com o usuário mostra que a adaptação automática de aspectos do sistema representa uma evolução dos sistemas interativos em geral.

Este trabalho aborda, de maneira geral, as duas principais tecnologias usadas para gerar adaptação automática. A primeira delas é a Modelagem de Usuários discutida na Seção 1.2 que trata principalmente das técnicas para aquisição e representação de modelos de usuário. A outra tecnologia é a Inteligência Artificial, que provê técnicas e algoritmos de raciocínio capazes de gerar suposições que são usadas para produzir as adaptações e é abordada na Seção 1.3.

Finalmente, diversos aspectos da tecnologia *World Wide Web* e que auxiliam no desenvolvimento de aplicações de Hipermídia Adaptativa são tratados na Seção 1.4.

1.1 Necessidade de Adaptação

Os computadores são ferramentas bastante úteis e versáteis para uma série de aplicações, auxiliando o homem em diversas atividades complexas. Em geral, estas atividades relacionam-se com o cálculo e o processamento de informações em altas velocidades ou o armazenamento de grandes quantidades de dados. No entanto, em sua origem, os computadores eram máquinas utilizadas apenas por especialistas em grandes centros de pesquisas. Além disto, processavam apenas uma coisa de cada vez.

Com o passar do tempo, novos equipamentos foram desenvolvidos. Com o objetivo de facilitar o controle destes dispositivos e aumentar a capacidade dos computadores, os estudiosos criaram diversos programas que tornaram a operação dos computadores e a criação de outros programas mais poderosos de maneira simples e direta. Com o passar do tempo, a relação custo-benefício estabeleceu os limites daqueles equipamentos que poderiam ser fisicamente criados e daqueles que seriam apenas simulados por programas de computador. Os programas cresceram em tamanho atingindo a condição de sistemas de *software* cada vez mais complexos.

Os *softwares* podem ser classificados de diversas maneiras dependendo do as-

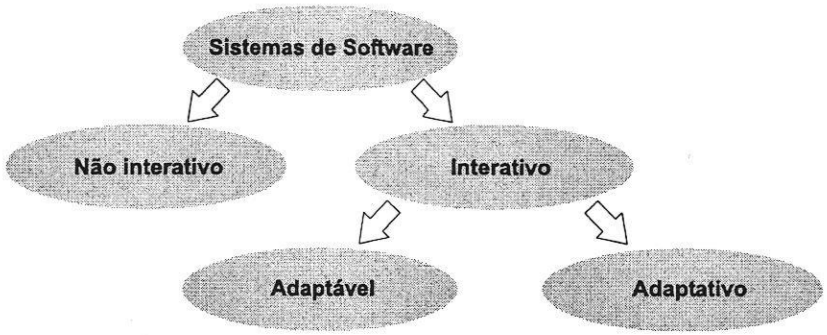

Figura 1.1: Classificação dos softwares quanto à interação com o usuário.

pecto a que se dê importância. A Figura 1.1 mostra uma classificação dos sistemas de *software* quanto à sua interação com o usuário. Pode-se dizer que os sistema interativos são freqüentemente utilizado por uma grande variedade de indivíduos com objetivo, interesse, nível de experiência, habilidades e preferências diferentes. A maior parte dos *softwares* é interativa, tais como o Sistema Operacional, os Sistemas Multimídia, os Sistemas Hipermídia e os Sistemas de Informação. Em geral, esse tipo de sistema costuma ser rígido. Isto é, o mesmo comportamento é apresentado para todos os usuários sem distinção. Portanto, faz-se necessário um recurso capaz de adaptar o sistema para atender as características e necessidades individuais dos usuários (Kobsa, 1994; Brusilovsky, 1996a, 1996b).

A forma mais simples de personalizar o comportamento do programa para os indivíduos é através da adaptação iniciada ou selecionada pelo usuário (Kobsa, 1994). É comum a existência de vários passos e opções de adaptação, o que dificulta a ação dos usuários neste sentido. Esta abordagem é pouco utilizada na prática, porque requer um considerável conhecimento do indivíduo a respeito do sistema e dos seus próprios erros, sobrecarregando o usuário em suas atividades. Quando o sistema permite que os indivíduos indiquem o comportamento que programa deve ter, diz-se que ele é **adaptável**.

Uma outra possibilidade é a de prover meios pelo qual o Sistema Interativo seja capaz de reconhecer a necessidade de adaptação. Para tanto, ele deve criar e manter modelos que registrem as características e/ou o comportamento dos usuários sobre as quais podem ser feitas suposições. Neste caso, o sistema pode adotar qualquer uma das seguintes abordagens (Kobsa, 1994):

1. automaticamente executar a adaptação no lugar do usuário; ou
2. aconselhar o indivíduo sobre a necessidade de adaptação, deixando-lhe a escolha por aceitá-la ou não.

Segundo o autor, o sistemas é **adaptativo** quando adota a primeira abordagem. Enquanto os sistemas com suporte para adaptação controlada pelo usuário adotam a segunda. É interessante observar que nem sempre é recomendável ao sistema realizar personalização automática para as características e necessidades pessoais. Quando a adaptação é pouco freqüente, mas muito significativa, Kobsa aconselha oferecer ao usuário a possibilidade de escolher pela sua efetivação ou não.

Como se pode notar, os diferentes tipos de adaptação dependem, portanto, da quantidade de controle do usuário e do sistema ao longo do processo de personalização. Adaptabilidade acontece quando a personalização é controlada pelo usuário. Por outro lado, a adaptatividade ocorre quando o sistema controla os passos da adaptação. Ambas podem coexistir em uma mesma aplicação em diferentes graus, como mostam Brusilovsky (1995) e Kobsa et al. (1999).

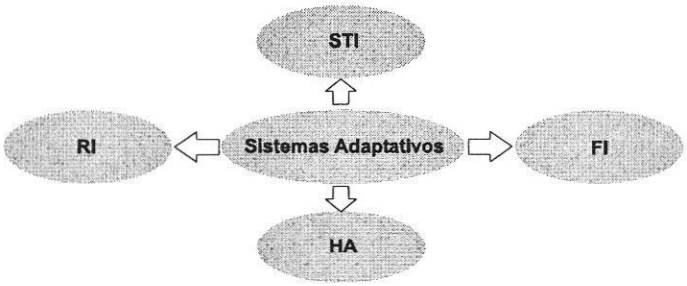

Figura 1.2: Aplicação de Sistemas Adaptativos.

Praticamente, toda aplicação que provê algum tipo de adaptação automática utiliza recursos da Modelagem de Usuários e da Inteligência Artificial (IA). Dentre as quais aparecem a Filtragem de Informação (FI), a Recuperação de Informação (RI), os Sistemas Tutores Inteligentes (STI), os Sistemas Especialistas (SE) e a Hipermídia Adaptativa (HA), entre outros. Assim, todas estas aplicações classificam-se em uma categoria mais geral de sistemas interativos capazes de prover adaptações automáticas - os Sistemas Adaptativos, como mostra a Figura 1.2. A diferença básica entre elas está no alvo da adaptação, na forma de representação e nos algoritmos usados para gerar as deduções sobre o modelo.

O modelo de usuário é o principal recurso usado para reconhecer a necessidade de adaptações. Ele contém as suposições do sistema sobre todos os aspectos relevantes para a realização de personalizações (Kobsa, 1993, 1994). De acordo com (De Bra, 1999), o modelo de usuário representa o estado mental dos indivíduos que acessam o sistema para decidir como realizar uma personalização. Portanto, todo Sistema Adaptativo precisa de um componente de Modelagem de Usuário que é responsável pelas seguintes tarefas:

1. identificar as informações significativas;
2. armazenar as informações em um sistema de representação apropriado;
3. inferir suposições sobre os dados armazenados;
4. manter a consistência do conjunto de suposições correntes; e
5. suprir outros componentes do sistema com suposições sobre os usuários.

Em (Kobsa et al., 1999) e (Brusilovsky, 1996a), os autores referem-se à Modelagem de Usuário como sendo parte de um processo de personalização, em especial para em Aplicações Hipermídia baseadas na WWW. O sistema colhe informações armazenando-as em um modelo que serve como base para a realização de adaptações. Portanto, o processo de adaptação é formado por três estágios, como mostra a Figura 1.3: aquisição, representação e produção da adaptação.

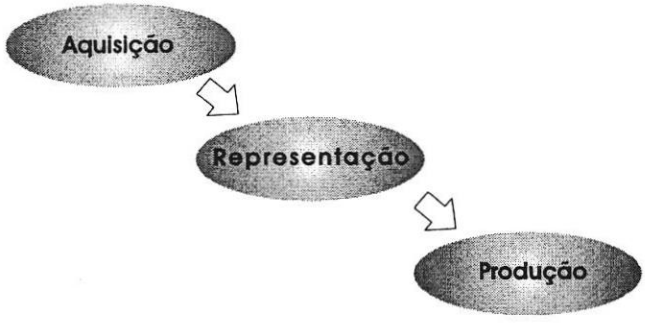

Figura 1.3: Processo de personalização em sistemas adaptativos.

1.2 Modelagem de Usuários

A Modelagem de Usuários (MU) surgiu a partir de investigações no campo dos Sistemas de Diálogo em Linguagem Natural (Kobsa, 1993). Mostra-se particularmente útil para aplicações que oferecem adaptações automáticas para os perfis de indivíduos.

Um conjunto de técnicas da Modelagem de Usuários é proveniente de outros campos de pesquisas, sobretudo da Inteligência Artificial e da Interação Homem-Computador. Outro grupo foi desenvolvido independentemente, encontrando aplicação em diversos Sistemas Interativos acessados por usuários heterogêneos. É o caso das Interfaces Inteligentes, dos Sistemas Hipermídia, dos Sistemas Tutores Inteligentes, dos Sistemas de Ajuda Passiva e Ativa, entre outros. A MU está inserida no processo de adaptação automática de Sistemas Adaptativos através da construção de modelos de usuário que envolve:

1. a coleta de dados significativos;
2. o processamento de dados para construir ou atualizar o modelo de usuário; e
3. a aplicação do modelo para a produção de adaptações.

As duas primeiras fases tratam dos métodos de aquisição e de representação das informações significativas obtidas de diferentes fontes para a construção de um ou mais modelos. Dependendo da representação escolhida, pode ser necessário realizar algum tipo de pré-processamento. A última etapa (produção da adaptação) envolve a aplicação de técnicas de inferência da Inteligência Artificial.

1.2.1 Aquisição de Dados

A fase de aquisição identifica e colhe as informações - suposições - importantes que estão disponíveis a respeito dos usuários, do seu comportamento de uso e do ambiente pelo qual o sistema é acessado. Além disto, a aquisição torna as informações acessíveis por meio de modelos iniciais de usuário, de uso e de ambiente (Kobsa et al., 1999). Nestes casos, normalmente é preciso realizar algum tipo de pré-processamento sobre os dados brutos.

O modelo inicial de usuário registra informações sobre características individuais, enquanto o modelo inicial de uso colhe dados sobre o seu comportamento observável. As informações de usuários e de uso costumam sobrepor-se, dificultando a distinção entre eles. Em geral, os dados de usuário abrangem as informações demográficas, os conhecimentos, as habilidades, os interesses, as preferências e os objetivos particulares.

Os dados de uso são provenientes do comportamento do usuário, sendo que podem ser diretamente observados, produzindo algum tipo de adaptação simples. Consideram-se **informações de uso diretamente observáveis** as ações seletivas e aquelas tomadas ao longo da interação com a aplicação, o comportamento temporal e a classificação sobre itens de interesse do indivíduo (Kobsa et al., 1999).

Em (Nichols, 1997), o autor faz a distinção entre a **classificação implícita** e **explícita**. Nesta, o usuário é diretamente questionado sobre o grau de importância do objeto de interesse. Naquela, a importância é automaticamente deduzida do comportamento do usuário. Para tanto, o autor fornece uma classificação dos tipos de dados observáveis implícitos baseado em ações freqüentemente realizadas ao longo da interação do usuário com a aplicação. Para (Oard & Kim, 1998), o comportamento observável implícito pode ser classificado em categorias gerais - Examinação, Retenção e Referência. A análise destas categorias revela a origem dos dados implícitos.

Outra abordagem, usada com freqüência, consiste em formar uma **base de conhecimento** onde são registrados os dados de uso para descobrir padrões gerais de regularidade. Alguns destes padrões podem manifestar as preferências, as habilidades, os hábitos ou o nível de conhecimento dos usuários que são necessários para gerar adaptações. Nestes casos, são considerdas como **informações de uso passíveis de registro e análise** a freqüência de uso, a correlação entre situação e ação e a seqüência de ações.

Com o objetivo de analisar a validade dos indicadores de comportamento implícitos, (Claypool et al., 2001) procura correlacioná-los com os indicadores explícitos. Para tanto, os autores classificam os indicadores de interesse em categorias (explícito, marcação, manipulação, navegação, externo, repetição e negativo). Um navegador especial foi criado para prover suporte na aquisição dos dados explícitos e implícitos diretamente. O resultado é que alguns indicadores implícitos,

como o tempo de permanência na página e o tempo total gasto com a rolagem do conteúdo com o mouse e o teclado, são bons indicadores de interesse comparados com a classificação explícita.

Um sistemas baseado na *Web* costuma ser acessado por uma grande variedade de plataformas de *hardware* e de *software*. O conjunto de características deste ambiente pode gerar adaptações úteis para melhorar o desempenho do sistema de um modo geral. O modelo de ambiente serve também para adequar os recursos à capacidade do *hardware* pelo qual o usuário acessa o sistema.

Há dois métodos gerais para adquirir os dados em Sistemas Interativos (Kobsa, 1993; Kobsa et al., 1999): aquisição ativa - baseada na entrada explícita de dados - e aquisição passiva - baseada na observação implícita do comportamento. As estratégias mais comuns para adquirir dados para um modelo de usuário são:

1. informação suprida pelo usuário: é uma estratégia de aquisição ativa simples, onde o usuário supre as informações necessárias ou rotula um conjunto de objetos conforme algum critério. Entretanto, nem sempre os indivíduos têm consciência de suas próprias capacidades, nem gostam de perder tempo com manuais, configurações, ajudas nem com a classificação de elementos.

2. regras de aquisição: é uma técnica de aquisição passiva menos perturbadora do que a aquisição ativa e, portanto, não inicia qualquer diálogo. Consiste na observação das ações, ou em uma interpretação mais ou menos simples do comportamento dos usuários. Uma regra de aquisição gera suposições sobre o usuário, e é disparada por eventos ou situações determinantes. Pode ser dependente ou não do domínio. O conjunto de regras dependentes é mais popular e fácil de implementar, mas são pouco flexíveis.

3. reconhecimento de planos: técnica de aquisição passiva que considera a relação que existe entre os objetivos do usuário e a seqüência de ações (plano) necessárias para atingi-los. Inclui um mecanismo para identificar o plano corrente e o objetivo associado pela observação das interações. Parece ser útil em situações com pequeno número de objetivos e meios de executá-los.

4. estereótipos: técnica de aquisição passiva, onde um indivíduo é classificado em uma categoria com um estereótipo que caracteriza os membros daquele grupo.

> O estereótipo presume que o usuário pode ser encaixado em classes previamente identificadas, com características e comportamentos inter-relacionados. A coleção de condições de ativação do estereótipo é avaliada e, se for satisfeita, incorpora o seu conteúdo como suposição no modelo do usuário. A eficiência dos estereótipos depende da qualidade de informações sobre a população e de quantos subgrupos com características relevantes para a aplicação podem ser distinguidos.

Pesquisas em Agentes de Interface ajudaram a desenvolver técnicas sofisticadas para aquisição de modelos de uso pela aplicação de algoritmos de aprendizado sobre a correlação entre situação-ação (Middleton, 2001). O modelo de uso é útil para predizer o comportamento dos usuários em situações futuras, bem como seqüências de ações.

Especificamente no caso de aplicações baseadas na WWW, a coleção de dados sobre o ambiente de *software* pode ser adquirida por meio da análise do cabeçalho das mensagens de pedido de recursos para um servidor HTTP. Algumas vezes, a informação sobre o *hardware* pode ser inferida a partir do tipo de navegador utilizado, mas em geral é difícil de obter este tipo de dado. O conhecimento sobre a largura de banda pode ser útil para decidir quais recursos incluir em uma página para diminuir o seu tempo de carga.

Nas situações onde é preciso utilizar o método de aquisição ativa, (Kobsa et al., 1999) aconselha evitar longos processos de registro ou entrevistas iniciais, limitando a duas ou três questões centrais embutidas no conteúdo principal. Além disto, nem sempre as informações supridas pelos indivíduos são confiáveis. Por esta mesma razão, o sistema que necessita da classificação de itens diretamente do usuário não tem garantia de que todos os elementos serão rotulados. Nem tampouco se esta classificação é significativa ou aleatória.

Quanto a técnica de aquisição passiva de dados de uso, não é recomendável registrar informação em nível de micro interação - como movimentos de mouse - a menos que o propósito desses registros tenham sido especificados. A quantidade de dados é grande, a computação para realizar as recomendações é longa e a confiabilidade na qualidade destas adaptações pode ser baixa. Apesar disto, (Kobsa et al., 1999) e (Brusilovksy, 2001) argumenta que a realização de experimento laboratorial com

dados desta natureza pode ajudar a orientar o desenvolvimento de métodos nesta área.

1.2.2 Representação de Dados

A fase de representação de dados concentra-se na expressão do conteúdo dos modelos em sistemas formais apropriados para que sejam acessado e processados. As informações de usuários, de uso e de ambiente são integradas para realizar suposições adicionais e generalizações pela aplicação de técnicas de inferência provenientes da Inteligência Artificial.

A principal forma de representação para modelos de usuários baseia-se na Lógica Clássica (Kobsa, 1993, 1994; Kobsa et al., 1999). Neste caso, o modelo encapsula o conhecimento a respeito do domínio ou do usuário na forma de **regras**. Este conhecimento é usado para disparar as adaptações necessárias. Existem duas restrições com esta abordagem: a capacidade limitada para lidar com a incerteza e com modificações nos modelos. Para representar o raciocínio com incerteza costuma-se empregar outros tipos de Lógica ou modelos probabilísticos que, no entanto, são mais difíceis de utilizar.

Uma forma de representação para o conhecimento dos usuários bastante comum é o **modelo de sobreposição** (Brusilovsky, 1996a; Marietto, 2000). Costuma ser aplicado em domínios estruturados como uma rede de conceitos rotulados e inter-relacionados, tal como em uma Rede Semântica. Para cada conceito do domínio, um modelo de sobreposição individual armazena um valor que corresponde a estimativa do nível de conhecimento do usuário sobre este conceito. Valores contínuos são um indicativo de incerteza ou do grau em que determinada característica está associada ao indivíduo. Um esquema como este é facilmente representado por um conjunto de pares de atributo-valor para cada conceito coberto pelo domínio. Algumas modificações no modelo de sobreposição permitem que outras características dos usuários sejam modeladas (Brusilovsky, 1996a).

Um perfil implícito ou explícito pode ser usado para gerar conclusões a respeito do comportamento do usuário. O processamento adicional sobre estes dados pode revelar o interesse, a preferência, o hábito, o nível de experiência ou a habilidade de um ou mais indivíduos. Nesta forma de representação, as características são

generalizadas para possibilitar adaptações em novos contextos (Brusilovsky, 1996a).

O **perfil de interesse explícito** é construído com base na análise das características dos objetos em que o usuário demonstra seu interesse por um mecanismo de votação ou classificação destes objetos. Neste caso, o perfil costuma ser expresso por meio de um vetor de características. Abordagens que consideram as características dos elementos são denominadas tecnologias de filtragem baseadas em conteúdos (Kobsa et al., 1999).

Por sua vez, o **perfil de interesse implícito** tenta encontrar usuários que apresentam comportamento de interação semelhante. Neste caso, o sistema adapta para um usuário individual baseado no comportamento de vizinhos de interesse (Kobsa et al., 1999). A classe ao qual um indivíduo pertence reflete um grupo de usuários com comportamentos e interesses similares. Desta forma, há uma forte possibilidade de que os objetos pertencentes a esta classe sirvam como recomendação útil aos seus membros. Essa abordagem é conhecida por alguns autores como tecnologia de filtragem baseada em clique.

É possível agrupar os perfis explícitos de usuários em categorias, de modo a explorar as semelhanças entre os usuários. O problema é que a classificação, neste caso é rígida, ou seja, é difícil especificar características individuais que não combinam com o grupo encontrado. Qualquer modificação no número de classes que não foram consideradas previamente requer a reestruturação do modelo de usuário.

A representação dos modelos de usuário, de uso e de ambiente na *World Wide Web* pode ser armazenada tanto no cliente quanto no servidor (De Bra, 1999). O armazenamento do modelo é temporário ou permanente. A manutenção de modelos no cliente prevê o uso de *cookies* que apresentam uma série de desvantagens e restrições. Por esta razão, o autor aponta o armazenamento do modelo de usuário no servidor como um importante recurso para a personalização de Sistemas Hipermídia baseados na *Web*.

1.2.3 Produção da Adaptação

O objetivo da MU é prover um modelo sobre a qual seja possível realizar a produção automática de personalizações sob vários aspectos de um Sistema Interativo. Adaptações são produzidas depois que o modelo foi adquirido e representado

de maneira formal, servindo de base para um módulo inteligente realizar inferências automaticamente.

A medida em que o usuário interage com o sistema, normalmente os modelos de usuário e de uso são atualizados e revistos. Há, basicamente, cinco situações em que os modelos podem sofrer modificações em Sistemas Adaptativos, segundo (De Bra et al., 2000):

1. o usuário faz uma seleção/escolha;
2. o usuário realiza um teste;
3. informações sobre os usuários são importadas de um sistema externo;
4. uma preferência do usuário é explicitamente declarada; e
5. uma preferência do usuário é automaticamente inferida a partir do seu comportamento.

Um Sistema Adaptativo é centrado no autor quando inclui revisões baseadas nas situações do tipo 2 e 3. Quando inclui revisões nas situações do tipo 4 e 5, o sistema é centrado no usuário. É possível, no entanto, combiná-las em uma mesma aplicação (De Bra et al., 2000). Quando a adaptação é oferecida como uma opção para o usuário, o sistema deve suprir meios para corrigir, desfazer ou até ignorar as modificações sugeridas.

A produção de adaptação prevê a modificação de alguma característica do Sistema Interativo. Quando se trata de aplicações de Hipermídia, (Brusilovsky, 1996a, 1996b, 2000a) mostram que apenas dois aspectos podem ser adaptados automaticamente: conteúdos e ligações. Mais tarde, talvez influenciado por (Kobsa et al., 1999), Peter Brusilovsky (2001) passa a considerar também a personalização dos formatos de mídia e dos elementos de interação.

A **adaptação de conteúdo** consiste na modificação do conteúdo Hipermídia para os usuário. Geralmente o modelo considera informações como os conhecimentos prévios e as demais páginas acessadas. Para realizar esta adaptação, utiliza-se, com freqüência, cinco funções de personalização sobre o modelo, segundo (Kobsa et al., 1999):

1. explanação opcional;
2. informações opcionais detalhadas;
3. recomendações personalizadas;
4. apresentação orientada a teoria; e
5. aconselhamento da oportunidade ótima.

De uma maneira geral, a adaptação da apresentação de objetos Hipermídia prevê a modificação do formato e da disposição dos elementos sem alterar o conteúdo. A modalidade é um tipo especial de **adaptação da apresentação** que atua sobre a natureza da comunicação, por exemplo de texto para áudio. A adaptação da apresentação geralmente considera as preferências, os interesses e as habilidades físicas dos usuários.

A **adaptação de estrutura** diz respeito às mudanças na maneira como as ligações são organizadas nas em Aplicações Hipermídia. Com freqüência, são empregadas as seguintes funções de personalização da estrutura sobre o modelo de usuário (Kobsa et al., 1999):

1. recomendação (sobre produtos, informações e navegação);
2. orientação ou guia adaptativo; e
3. visões pessoais.

1.2.4 Métodos e Técnicas de Adaptação

A adaptação de conteúdo e de estrutura envolve a aplicação de métodos e técnicas bastante pesquisadas e que tem sido empregados com regularidade em Aplicações Hipermídia Adaptativas baseadas na WWW. A primeira ampla classificação de métodos e técnicas com este propósito aparece em (Brusilovsky, 1996a).

Recentemente, novos aspectos e tecnologias para adaptação Hipermídia foram desenvolvidas. Como resultado, a taxonomia foi revista e modificada em (Brusilovsky, 2001), como mostra a Figura 1.4. No entanto, o primeiro trabalho ainda é uma

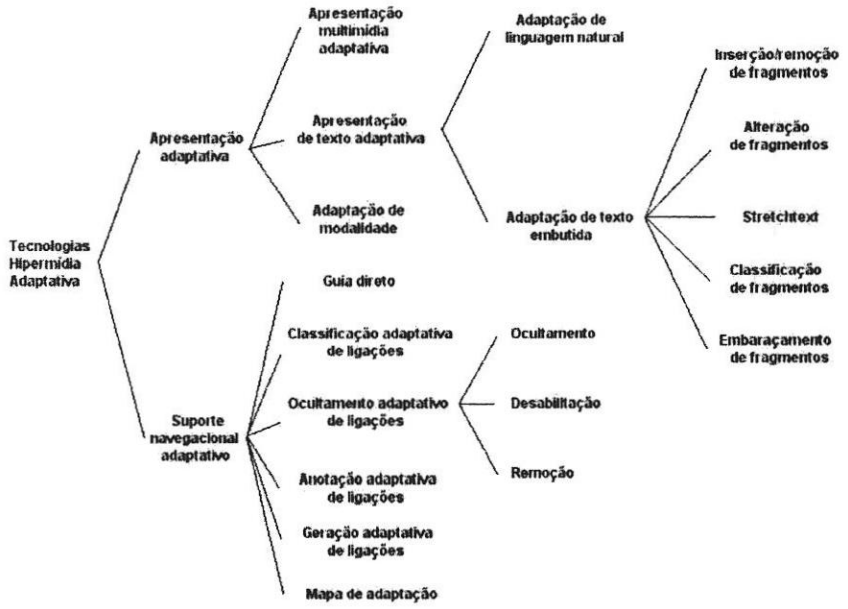

Figura 1.4: Tecnologias de personalização Hipermídia (Brusilovsky, 2001).

importante referência contendo uma descrição detalhada das principais tecnologias para a personalização no ambiente WWW.

O termo adaptação de conteúdo empregado por (Kobsa et al., 1999), (De Bra, 1999) e (De Bra et al., 1999a, 1999b) é chamada de apresentação adaptativa por Peter Brusilovsky. Enquanto o termo adaptação de estrutura utilizado por (Kobsa et al., 1999) é equivalente a adaptação de ligações em (De Bra, 1999; De Bra et al, 1999a, 1999b) e a suporte navegacional adaptativo (Brusilovsky, 1996a, 2001).

O conjunto de métodos e técnicas para **apresentação adaptativa de textos** é

Tabela 1.1: Métodos para a apresentação adaptativa de texto.

Additional explanation	oculta partes de informações sobre um conceito em particular que não seja relevante.
Prerequisite explanation	modifica informações apresentadas dependendo do conhecimento sobre conceitos relacionados.
Comparative explanation	oferece explicações comparando conceitos bem conhecidos pelo usuário.
Explanation variants	o sistema mantém uma grande variação de explicações sobre um mesmo conceito.
Sorting	classifica trechos de informações sobre conceitos relevantes ao usuário.

Tabela 1.2: Técnicas para apresentação adaptativa de texto.

Conditional text	divide os conceitos em pedaços menores associados a condições sobre o nível de conhecimento dos usuários.
Stretchtext	a ativação de uma palavra-chave causa a inserção de um texto associado dentro do nó corrente.
Fragment variants	usuário acessa explicações para os conceitos fragmentados em diversos níveis de conhecimento.
Page variants	duas ou mais versões para um mesmo conceito são preparadas para possíveis classes de usuários.
Frame-based technique	informações são apresentadas em *frames* com variações nas explicações, conceitos, ligações selecionadas por regras especiais.

Tabela 1.3: Métodos para suporte navegacional adaptativo.

Global guidance	Provê informações gerais que auxiliam o usuário a encontrar o menor caminho até a informação que precisa.
Local guidance	Ajuda o usuário a decidir pelo melhor nó a ser visitado a seguir a partir do estado corrente.
Local orientation support	Ajuda o usuário a localizar sua posição relativa dentro do documento.
Global orientation support	Ajuda o usuário a entender a estrutura geral e sua localização absoluta dentro dele.

descrito com detalhes em (Brusilovsky, 1996a) e aparece resumido nas Tabelas 1.1 e 1.2, respectivamente. Tecnologias mais recentes, como adaptação multimídia e de modalidade, não são cobertas com detalhes por (Brusilovsky, 2001) e, por isto, não aparecem nas Tabelas.

Trabalhos como (Brusilovsky, 1995, 1996b), (Mullier, 1995) e (De Bra et al., 1999b) consideram que o usuário inexperiente precisa de explicações adicionais e um controle maior do sistema, enquanto que o usuário avançado está interessado em detalhes e na liberdade de navegação.

De acordo com De Bra et al. (1999b), a técnica básica para apresentação de texto adaptativa é o *conditional text* que pode ser facilmente usado para implementar *stretchtext*, *fragment variants* e *page variants*. Em geral, o Sistema Adaptativo oculta fragmentos de conteúdos considerados indesejáveis aos interesses do usuário. No entanto, vários autores sugerem que o sistema deve permitir que o usuário veja todos os itens e não apenas o subconjunto recomendado.

A adaptação de ligações, segundo (De Bra et al., 1999b), modifica ou anota a estrutura de ligações de modo a orientar o usuário em direção à informações interes-

Tabela 1.4: Técnicas para o suporte navegacional adaptativo

Direct guidance	o sistema decide qual o melhor nó a ser visitado pelo usuário no passo seguinte.
Sorting	utiliza algum critério útil para ordenar um conjunto de ligações colocando os mais significativos em primeiro lugar.
Hiding	restringe o espaço de navegação ocultando ligações pouco relevantes para os usuários.
Annotation	coloca comnetários ao lado das ligações que indiquem algo sobre o estado corrente dos nós apontados pelas ligações.
Map adaptation	compreende diferentes maneiras de adaptar mapas hipermídia locais e globais.

santes e relevantes. Adaptação de ligações simplifica a estrutura Hipermídia para reduzir o problema da desorientação, ao mesmo tempo em que mantém a liberdade de navegação. Os métodos e técnicas para **suporte navegacional adaptativo** são classificados por (Brusilovsky, 1996a) como mostram as Tabelas 1.3 e 1.4, respectivamente.

O suporte à orientação indica, de alguma forma, quais ligações são preferíveis sobre outras. Para tanto, depende da determinação de contextos e requer um mapa da estrutura de ligações ao redor do nó corrente. Um conjunto de técnicas sugeridas em (De Bra et al., 1999b) sugere a desabilitação ou remoção de ligações. No entanto, existem evidências de que não é desejável utilizar este tipo de recurso, pois diminuem a capacidade de compreensão e podem tornar certos conteúdos inacessíveis.

Não é possível implementar todas as técnicas em um único sistema adaptativo, pois isto o tornaria inutilizável. O projetista deve escolher a(s) técnica(s) desejada(s) e desenvolver ou selecionar um sistema que suporte a escolha.

1.2.5 Considerações Finais

Tradicionalmente, é construído um modelo simbólico para os objetivos, conhecimentos, experiência e preferências individuais dos usuários. Estas características costumam ser significativas e fáceis de modelar. Além disto, podem ser interpretadas e compreendidas com facilidade.

Recentemente, os pesquisadores têm investigado o uso de características individuais em modelos de usuários para Sistemas Adaptativos (Brusilovsky, 2001). O

conjunto de peculiaridades que define uma pessoa (personalidade, fatores cognitivos e estilos de aprendizado) costuma ser estável e compõe as características individuais. Este tipo de informação pode ser extraída por meio de testes psicológicos especialmente desenvolvidos. Pesquisadores concordam que características individuais são importantes, mas ainda não sabem quais características podem ser usadas de fato e nem como emprega-las (Brusilovsky, 2001).

Segundo (Mullier, 1995) e (Mullier et al., 1999), a MU é criticada porque pode ser aplicada apenas para domínios restritos. Duncan Mullier afirma que a representação do modelo de usuário baseada em conhecimento é incompleta, incerta, ambígua, desestruturada e instável, pois considera suposições e regras pré-definidas. O problema é que as dependências do domínio herdadas de regras podem não descrever ou predizer grande parte do comportamento humano. Por isto, diversos autores sugerem o uso de outras tecnologias e abordagens inteligentes para melhorar a Modelagem de Usuários.

Uma discussão sobre a confiabilidade da adaptação e da MU aparece em (Brusilovsky, 1996a). Nela, o autor mostra que o sistema pode errar quando deduz o modelo de usuário ou quando provê a adaptação e o modelo está incorreto. Quando o Sistema Adaptativo for um Sistema Hipermídia baseado na WWW a Modelagem de Usuários fica ainda mais difícil, pois a fonte de dados é restrita e incompleta, dificultando a atualização do modelo de usuário. A maneira encontrada pelo autor para resolver o problema é envolver o usuário em um processo de modelagem colaborativa. Desta maneira, o sistema colhe os dados sobre usuários a partir da aplicação e do próprio indivíduo. O modelo de usuário gerado pode ser modificado pelo administrador do sistema ou pelo próprio indivíduo.

Enfim, o progresso em Sistemas Adaptativos depende dos avanços nos campos da Modelagem de Usuários e da Extração Automática de Conhecimentos (Brusilovsky, 2001). Entretanto, a MU revela apenas uma parte do processo de adaptação automática. Para que esta seja realmente efetivada é necessário recorrer a técnicas provenientes de diversas áreas. A principal delas, segundo o autor, é o Aprendizado de Máquina e a aplicação dos algoritmos de filtragem e recuperação de informações.

Convém notar que não existe algoritmo que resolva todos os problemas de adaptação e nem tampouco que seja adequado para todos os domínios. Recentemente, investiga-se a utilização de abordagens híbridas com bons resultados, como

mostram (Kobsa et al., 1999) e (Brusilovsky, 2001). As próximas Seções investigam tópicos de Inteligência Artificial e outras tecnologias que têm sido utilizadas para a produção automática de adaptações.

1.3 Inteligência Artificial

A Inteligência Artificial (IA) é uma área de pesquisa bastante ativa da Computação e que têm sido crescentemente utilizada em aplicações práticas. Ela herda muitas idéias, pontos de vista e técnicas de outras disciplinas, como Filosofia, Matemática, Psicologia e Lingüística (Russel & Norvig, 1995). Esta Seção aborda alguns tópicos da IA úteis para o desenvolvimento de Sistemas Adaptativos de maneira geral, em especial o Aprendizado de Máquina.

Epistemologicamente, existem três tipos de raciocínio, segundo (Kobsa et al., 1999). A dedução parte de fatos gerais para produzir suposições a respeito de casos mais específicos. A indução considera os casos específicos para obter conclusões gerais. Por sua vez, a analogia faz uso de casos similares para supor algo a respeito do caso atual. Para cada tipo de raciocínio um conjunto diferente de métodos e técnicas de IA é selecionado, definindo paradigmas distintos.

A abordagem baseada em conhecimento pressupõe uma base de fatos sobre o mundo. O raciocínio dedutivo é aplicado para determinar possíveis ações. Para tanto, costuma-se utilizar a Lógica Clássica: um sistema formal de representação de conhecimento que oferece um conjunto de regras de dedução na forma de uma teoria de provas (Russel & Norvig, 1995). Sistemas baseados em conhecimento são comuns e poderosos, capazes de se adaptarem ao ambiente.

O raciocínio analógico é comum em Sistemas de Recomendação. Decisões são tomadas com base na identificação de um indivíduo que atribui rótulos a certos elementos que sejam semelhantes aos rótulos feitos por outros usuários. O conjunto de indivíduos que classificaram elementos de maneira semelhante formam *clusteres*[1] que ajudam a predizer o interesse do usuário sobre novos objetos. Em geral, o processo de gerar recomendações possui três fases (Kobsa et al., 1999): encontrar vizinhos semelhantes, selecionar um grupo de comparação de vizinhos e computar

[1] agrupamentos de elementos com características semelhantes.

predições. Em geral, emprega-se métodos estatísticos ou Redes Neurais Artificiais para resolver problemas de *Clusterização*[2].

A abordagem de aprendizado da IA considera o monitoramento das interações dos usuários com a aplicação e do seu comportamento. O sistema utiliza o raciocínio indutivo para produzir conclusões gerais a partir de uma série de observações sobre os dados obtidos na monitoração. Com base nestas conclusões gerais o sistema pode tomar decisões que melhoram o próprio desempenho.

De acordo com (Russel & Norvig, 1995), o objetivo da Inteligência Artificial é desenvolver um Agente. Trata-se de um programa de computador capaz de raciocinar e/ou atuar na realização de alguma tarefa específica. Desta maneira, o estudo dos Agentes fornece uma ampla abordagem para o estudo das técnicas de IA. A próxima seção oferece uma visão geral a respeito da Tecnologia dos Agentes, suas aplicações e arquiteturas.

1.3.1 Tecnologia dos Agentes

Na literatura não existe uma única definição de Agente, mas sim definições contextuais que dependem dos objetivos e interesses das aplicações ou do pesquisador. Neste trabalho, será considerada a definição contextual estabelecida por Russel & Norvig (1995) como aparece na Figura 1.5. Por esta definição, o Agente é algo capaz de perceber e agir sobre o ambiente através de sensores e de realizadores, respectivamente.

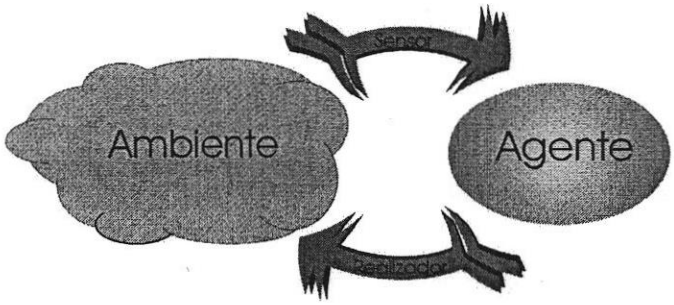

Figura 1.5: Representação de um Agente.

O programa Agente executa em uma arquitetura que pode ser um computador

[2]O objetivo da *clusterização* é reduzir a quantidade de dados agrupando-os em categorias ou *clusters*.

normal ou um dispositivo de *hardware* especialmente desenvolvido para certas atividades. Simplificadamente pode-se dizer que a arquitetura percebe o ambiente por meio de sensores, executa o programa e escolhe os realizadores para o qual a ação foi gerada (Russel & Norvig, 1995).

A racionalidade - fazer a coisa certa - compreende o raciocínio correto e a ação em um ambiente. A Lógica oferece um sistema formal que fundamenta o raciocínio correto. O comportamento humano, por sua vez, é altamente adaptado a um ambiente específico e seu produto - a ação - é um processo evolucionário complexo e desconhecido. Para que um Agente seja considerado racional, deve haver critérios que determinam o quanto ele é bem sucedido. O ser humano, atuando como observador externo, estabelece um padrão do que significa ser bem sucedido em um ambiente. Este padrão é, então, usado como medida de desempenho do Agente. O instante em que o desempenho é avaliado importa para decidir se recompensamos ou punimos um Agente.

Um Agente não pode ser culpado por falhar ao tomar conta de alguma coisa que ele não percebeu, ou decidindo por uma ação que ele é incapaz de realizar. Afinal, é impossível projeta-lo de modo a satisfazer todas as condições pré-estabelecidas. Assim, o que é racional em um dado instante depende de quatro fatores, conforme (Russel & Norvig, 1995):

1. a medida de desempenho, que define o grau de sucesso;
2. a seqüência de percepção, isto é, tudo aquilo que o Agente consegue perceber até um determinado momento;
3. o conhecimento sobre o ambiente; e
4. as ações que podem ser realizadas.

O comportamento de um Agente depende apenas de dados da seqüência percebida. Desta forma, uma arquitetura inicial pode ser descrita por meio de uma tabela de ações tomadas em resposta a cada possível seqüência percebida. No entanto, nem sempre é preciso criar uma tabela explícita com uma entrada para cada seqüência percebida possível. Assim, um Agente Racional mais compacto pode ser construído

```
Função AGENTE(percepção)
  estático: memória

  memória = AtualizaMemória(memória, percepção)
  ação = EscolheMelhorAção(memória)
  memória= AtualizaMemória(memória, ação)
  retorna ação
Fim função
```

Figura 1.6: Estrutura genérica de um programa Agente.

através de uma especificação do mapeamento sem a necessidade de enumerá-lo explicitamente.

Antes de projetar um programa agente, é possível ter uma boa idéia das possíveis percepções e ações, quais objetivos ou medidas de desempenho o suposto Agente deve executar, bem como o ambiente em que ele irá operar. A estrutura geral de programas agentes aparece na Figura 1.6. O programa agente recebe uma percepção simples como entrada. A seqüência percebida pode ser montada em memória, se desejável. O objetivo ou medida de desempenho não faz parte do programa agente, pois é aplicada externamente para julgar o seu comportamento. Em (Russel & Norvig, 1995), a estrutura geral do programa é modificada para acomodar cada um dos seguintes tipos de Agentes:

1. agente reflexo simples: resume partes da tabela de mapeamento do agente por meio de certas associações comuns que ocorrem na entrada/saída. Essas conexões são chamadas de regra condição-ação.

2. agente reflexo com estado: decisões podem considerar também um estado interno que ajuda a escolher a ação mais adequada. O estado interno provê informações para distinguir as condições do mundo que geram a mesma entrada percebida, mas, no entanto, são significativamente diferentes. Ou seja, o Agente precisa saber como o mundo evolui e como a sua ação pode afeta-lo.

3. agentes baseados em objetivos: precisam de algum tipo de informação a respeito do objetivo, isto é, situações que desejam alcançar. A combinação do resultado de possíveis ações com objetivos ajudam a escolher as melhores ações para atingir o alvo desejado.

4. agentes baseados em utilidade: geram comportamentos de alta qualidade. Algumas decisões podem ser mais seguras, mais confiáveis ou mais baratas do que outras. Uma medida de desempenho permite a comparação de diferentes estados do mundo identificando as melhores ações em determinadas situações. A esta característica dá-se o nome de utilidade do Agente. É uma função que mapeia o estado em um número real que descreve o grau de satisfação associado.

Um tipo especial de programa agente - o Agente de Interface - surgiu da integração da IA com a Interação Homem-Computador. O seu objetivo é liberar o ser humano de atividades consideradas repetitivas e cansativas. Pela definição de Maes apud Middleton (2001), um Agente de Interface possibilita o engajamento entre indivíduos e computadores em um processo de comunicação, monitoração de eventos e realização de tarefas de maneira colaborativa. Para ser bem sucedido, um Agente de Interface precisa conhecer, interagir e ajudar os usuários de maneira competente.

Figura 1.7: Tipos de Agentes de Interface.

Um Agente de Interface pode ser classificado sob várias perspectivas: do algoritmo de aprendizado, da função que executa, da tecnologia utilizada e do domínio ao qual faz parte (Middleton, 2001). Para o autor, a tecnologia muda com menos freqüência e, portanto, oferece uma base mais estável para construir uma taxonomia útil e não exclusiva. Assim, uma aplicação pode ser classificada em mais de uma categoria. A Figura 1.7 mostra a taxonomia do autor. Agentes sociais raciocinam por analogia sobre padrões de comportamento e/ou de interesse de outros usuários. Os agentes que aprendem sobre os usuários utilizam o raciocínio indutivo e podem adotar uma das seguintes abordagens: monitorar o comportamento do usuário, receber um *feedback* explícito ou implícito sobre algum item de interesse ou por programação. Os agentes com modelos de usuários, por sua vez, empregam o raciocínio

dedutivo sober um conjunto de regras que representam os usuários em um determinado domínio.

O Sistema de Recomendação é um tipo de Agente de Interface que tem sido comumente utilizado para resolver o problema de sobrecarga de informações na WWW. Ele filtra o tipo de informação desejada e apresentar apenas os itens úteis no momento da consulta. Várias abordagens podem ser utilizadas:

1. o indivíduo pode classificar itens em um conjunto de exemplos positivo (interessantes) e um conjunto de exemplos negativos (não interessantes). Com uma quantidade suficiente de exemplos positivos e negativos, o algoritmo de Aprendizado de Máquina pode classificar novas páginas com boa precisão.

2. o conjunto de exemplos positivos pode ser deduzido a partir da monitoração do que o usuário está procurando, sem interferir nas suas atividades normais. O conjunto de exemplos negativos pode ser deduzido por meio de heurísticas. Típico em Sistemas de Recomendação baseados em conteúdo para sugerir novos itens que combinam com o perfil do usuário.

3. sistema de recomendação colaborativo considera um conjunto de rótulos feito por outras pessoas que tiveram acesso ao mesmo conjunto de itens. Para tanto, obtem-se uma classificação explícita do usuário para o item e procura-se por uma classe de usuários que tenham atribuído taxas similares.

O Sistema de Recomendação pode utilizar perfis de usuários baseado em conhecimento ou no comportamento. O primeiro faz uso de um modelo estático de usuário, procurando combinar as características individuais com aquelas existentes no modelo. O segundo usa o próprio comportamento de interação do indivíduo com o sistema como um modelo de usuário dinâmico. O perfil baseado em comportamento tem sido empregado com mais freqüência, como mostra a análise feita por Middleton (2001).

Resumidamente, pode-se dizer que os agentes são programas de computadores que raciocinam e/ou atuam na realização de tarefas de maneira automática no lugar do ser humano. Assim, a tecnologia dos agentes revela-se bastante útil em Hipermídia Adaptativa. De maneira que pode-se criar um programa que observa,

registra e analisa o comportamento e as ações do usuário ao mesmo tempo em que busca na WWW por informações que combinem com o seu interesse.

1.3.2 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é uma sub-área da IA que tem sido utilizado para prover adaptações a partir de modelos de usuários. O raciocínio indutivo característico dos algoritmos de aprendizado é capaz de predizer evento futuros, fazer generalizações sobre características específicas, bem como encontrar agrupamentos naturais a partir de características semelhantes. Ainda que as conclusões obtidas pelo raciocínio indutivo possam ser incorretas, a inferência indutiva, de acordo com (Baranauskas & Monard, 2000), é um dos principais métodos para derivar conhecimento novo e predizer eventos futuros.

De acordo com (Mitchell, 1997), aprendizado refere-se a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Desta maneira, o Aprendizado de Máquina (AM) consiste em um programa de computador que realiza um conjunto de atividades capaz de melhorar seu desempenho a cada experiência, em um processo de aquisição automática de conhecimento. De modo geral, o aprendizado indutivo divide-se em **supervisionado** e **não-supervisionado**, como mostra a Figura 1.8. A diferença entre eles está na maneira como o aprendizado ocorre. No primeiro, cada exemplo é rotulado para uma classe conhecida através de um agente externo. No caso do aprendizado não supervisionado, os exemplos não são rotulados por nenhum especialista. O conjunto de exemplos é analisado para tentar encontrar *clusters*.

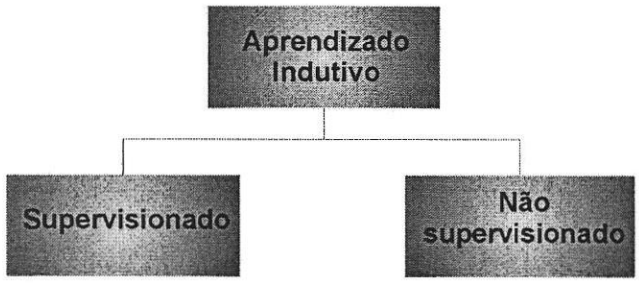

Figura 1.8: Tipos de Aprendizado Indutivo.

Conceitos Gerais

O aprendizado por indução acontece a partir da análise sobre um conjunto de exemplos fornecidos por um processo externo ao sistema. Em um processo de adaptação, equivale à etapa de aquisição de dados. O objetivo dos **algoritmos de aprendizado indutivo**[3] é encontrar a hipótese de indução - uma função que aproxima o conjunto de exemplos - com a menor taxa de erro possível (Russel & Norvig, 1995).

O algoritmo de aprendizado deve encontrar corretamente a classe de novos exemplos ainda não considerados utilizando, para isto, o **classificador** [4] obtido. Este processo pode ser uma **classificação** - quando os rótulos possuem valores discretos - ou uma **regressão** - quando são usados valores contínuos para os rótulos.

Cada **exemplo**[5] é descrito por uma tupla contendo o valor de características descritas por **atributos**. No caso do aprendizado supervisionado, adiciona-se a cada exemplo um atributo que é o rótulo da **classe** associada. O atributo classe serve para descrever o fenômeno de interesse. Representa a meta que se deseja aprender e sobre a qual se pode gerar predições a respeito. Existe, em geral, dois tipos básicos de atributos: nominal, quando não existe nenhuma ordem de importância entre os valores; e contínuo, quando os valores possuem uma ordem linear.

Normalmente, o **conjunto de exemplos** é dividido em dois subconjuntos disjuntos: o conjunto de treinamento, usado para o aprendizado do conceito; e o conjunto de teste, para medir o grau de efetividade do conceito aprendido. Muitas vezes, os exemplos contém **ruído**, isto é, erros na sua descrição que podem ser expressos por ausência de valores nos atributos ou pela classificação de um exemplo com rótulo incorreto.

O algoritmo de aprendizado pode chegar a diferentes hipóteses para uma mesma função ou problema. Neste caso, a escolha da melhor aproximação é determinada por um *bias* de aprendizado que representa o conhecimento adicional necessário para determinar a alternativa mais adequada (Baranaukas & Monard, 2000). Quando a hipótese for muito específica e ajustar-se excessivamente para um conjunto de treinamento, diz-se que ocorreu um *overfitting*[6]. Quando a hipótese ajusta-se muito pouco

[3] também chamado *indutor*
[4] a saída do indutor
[5] caso, registro ou instância
[6] sobre ajuste do algoritmo para o conjunto de dados.

ao conjunto de treinamento, ocorre um *underfitting*[7]. Quando os parâmetros de um algoritmo de aprendizado são ajustados em excesso para otimizar seu desempenho em todos os exemplos disponíveis, diz-se que ocorre um *overtuning*[8].

O modo de aprendizado pode ser **incremental** ou **não incremental**. Este assume que o conjunto de treinamento está presente para o aprendizado. Enquanto o aprendizado incremental acontece quando o conjunto de treinamento não está disponível desde o início do aprendizado. Desta forma, a hipótese é atualizada cada vez que um novo exemplo é acrescentado ao conjunto de exemplos.

O desempenho de um algoritmo de aprendizado é expresso pela **taxa de erro** (*err*) de um classificador. É obtida quando se compara a diferença entre a classe verdadeira de um exemplo e o rótulo atribuído pelo classificador. Quando a saída do indutor classifica corretamente um exemplo, isto é, quando coincide com a classe verdadeira, o operador ||E|| retorna 1, caso contrário, retorna 0. A **precisão** (*acc*)do classificador é o complemento da taxa de erro.

$$\mathrm{err(h)} = \frac{1}{n} \sum_{i=1}^{n} \parallel y_i \neq \mathrm{h}(x_i) \parallel \qquad \text{(taxa de erro)}$$

$$\mathrm{acc(h)} = 1 - \mathrm{err(h)} \qquad \text{(precisão)}$$

Em problemas de regressão, o **erro da hipótese** de um classificador é estimado pela distância entre o valor real e o valor atribuído pela hipótese induzida. Duas medidas para o erro da hipótese são normalmente usadas: o erro médio quadrado (*mse-err*) e a distância absoluta média (*mad-err*).

$$\text{mse-err(h)} = \frac{1}{n} \sum_{i=1}^{n} (\mathrm{y}_i - \mathrm{h}(x_i))^2 \qquad \text{(erro médio quadrado)}$$

$$\text{mad-err(h)} = \frac{1}{n} \sum_{i=1}^{n} \mid y_i - \mathrm{h}(x_i) \mid \qquad \text{(distância absoluta média)}$$

A **distribuição da classe** (*distr*) representa a proporção de exemplos em cada classe. A classe com maior distribuição é chamada de majoritária ou prevalente, ao passo que a classe com menor distribuição é denominada de minoritária. Com base na distribuição é possível calcular o **erro majoritário** (*maj-err*), isto é, o limite máximo para o erro de um classificador.

[7] sub ajuste do algorimos em relação aos dados.
[8] afinação excessiva dos parâmetros para os dados.

$\text{distr}(C_j) = \frac{1}{n} \sum_{i=1}^{n} \parallel y_i = C_j \parallel$ (distribuição de classe)

$\text{maj-err(T)} = 1 - max_{i=1,\cdots,k}\ \text{distr}(C_i)$ (erro majoriário)

A classificação ou predição correta para cada classe sobre um conjunto de exemplos podem ser representada em um **matriz de confusão.** Nela, aparece o número de classificações corretas e o número de classificações preditas para cada classe. A matriz é formada por elementos M(C_i,C_j), onde 1 $\leq$ i,j $\leq$ k e k representa a quantidade de classes diferentes no conjunto de exemplos E. Cada elemento é calculado pelo número de exemplos que realmente pertencem à clasase C_i, mas que foram classificados como C_j.

$\text{M}(C_i, C_j) = \sum_{\{}\forall(\text{x,y})\in\text{E:y}=C_i\}\parallel \text{h(x)} = C_j \parallel$ (matriz de confusão)

A diagonal principal da matriz de confusão indica o número de acertos para cada classe. Os outros elementos de M(C_i,C_j), onde $i \neq j$ são representam os erros para um tipo de classificação em particular.

A partir da matriz de confusão é possível derivar uma série de erros. Considerando um problema com duas classes (positivo e negativo), os erros possíveis são falso positivo e falso negativo. As classificações corretas podem ser verdadeiro positivo e verdadeiro negativo. Nestas condições, a **taxa de erro da classe** é definida pela relação existente entre o número de exemplos classificados como falso negativos e a soma do número de exemplos rotulados pelo classificador como verdadeiro positivo e falso negativo.

O Aprendizado de Máquina é uma sub-área importante e poderosa de IA. Normalmente, para resolver um problema é possível selecionar um conjunto de algoritmos de aprendizado. Cada um deles faz parte de um dos cinco paradigmas de AM:

1. simbólico: o aprendizado acontece por representação simbólica de conceitos, por meio da Lógica, Rede Semântica, Regras de Produção ou Árvores de Decisão;

2. estatístico: emprega medidas estatísticas encontrar uma boa aproximação do conceito induzido, como o aprendizado Bayesiano;

3. *instanced based* (baseado em exemplos): classifica exemplos com base em outros exemplos semelhantes, tal como *Nearest Neighbours* e Raciocínio Baseado em Casos;

4. conexionista: inspirado no modelo biológico do cérebro e tem sido empregado com sucesso para resolver problemas que exigem intenso processamento sensorial humano, como o Reconhecimento de Voz. Emprega Redes Neurais; e

5. evolutivo: fundamenta-se na genética e na seleção natural, onde os elementos de classificação competem de maneira que os mais forte proliferam, produzindo variações de si mesmo.

Um sistema de aprendizado de caixa-preta produz uma representação interna que não é facilmente interpretada pelo ser humano ou que não ofereça meios de explicar o processo de reconhecimento. Assim acontece com as Redes Neurais. Por outro lado, sistemas orientados a conhecimentos fazem uso de uma estrutura simbólica compreensível ao ser humano. A Rede Semântica ilustra este tipo de sistema. Para resolver o problema da dificuldade de entender o raciocínio usado pelos sistemas de aprendizado de caixa-preta é que foram desenvolvidas técnicas de Extração de Conhecimentos a partir do indutor obtido.

Avaliação de Algoritmos de AM

Não é possível determinar um único algoritmo de AM que apresente o melhor desempenho em todos os problemas. Por esta razão faz-se necessário empregar algum mecanismo que permita avaliar e comparar estes algoritmos. O principal método usado para atingir este objetivo é a amostragem (*resampling*).

Por este método, o desempenho do classificador induzido é estimado usando um conjunto de exemplos (uma distribuição) em um certo domínio. Seja uma distribuição D e um conjunto de amostras extraídas de D, obtém-se a estimativa da precisão e do erro treinando-se o indutor com essas amostrar e testando seu desempenho em exemplos fora da amostra usada para treinamento. Para que a medida seja verdadeira, a amostra deve ser aleatória. Existem vários métodos baseado em amostragem que podem ser aplicados para avaliar o desempenho de algoritmos de Aprendizado de Máquina:

1. *holdout*: divide o conjunto de exemplos em porcentagens fixas para o treina-

mento e o teste. Costuma-se empregar 2/3 do conjunto de exemplos para o treinamento e 1/3 para os testes. Em geral, considera-se o conjunto de exemplos para treinamento maior que 1/2;

2. *cross-validation*: em *k-fold cross validation* os exemplos são aleatoreamente divididos em k partições disjuntas de tamanho n/r, aproximadamente. Utiliza-se (k − 1) partições para o treinamento e a partição restante é usada para teste. O processo é repetido k vezes e o erro é a média dos erros calculados em cada k partição; e

3. *leave-one-out*: é um caso especial de *cross-validation* aplicado para pequenos conjuntos de exemplos. São utilizados (n − 1) exemplos para o treinamento e 1 exemplo para teste. Repete-se o processo testando um exemplo diferente a cada iteração. O erro é obtido pela média dos erros em cada iteração.

Em geral, para um algoritmo de aprendizado A e um conjunto de exemplos que foi particionado, calcula-se a média (*mean*), a variância (*var*) e o desvio-padrão (*sd*). Para tanto, considera-se o erro do classificador h_i (*err*(h_i)) de cada partição.

$$\text{mean(A)} = \frac{1}{r} \sum_{i=1}^{r} \text{err}(h_i) \qquad \text{(média)}$$

$$\text{var(A)} = \frac{1}{r} \left\lfloor \frac{1}{r-1} \sum_{i=1}^{r} (err(\text{h}_i) - \text{mean(A)})^2 \right\rfloor \qquad \text{(variância)}$$

$$\text{sd(A)} = \sqrt{var(A)} \qquad \text{(desvio-padrão)}$$

O erro de um algoritmo de AM é expresso por sua média seguida pelo símbolo ± e pelo seu desvio padrão (8,30±1,00, por exemplo). O desvio padrão representa a robustez do algoritmo. Isto é, as variações nos erros provenientes de classificadores usando diferentes conjunto de treinamento e teste. Quando essa variação é muito grande de um experimento para outro, pode-se dizer que o indutor não é robusto a mudanças no conjunto de treinamento obtido de uma mesma distribuição.

Para determinar, com grau de confiança de 95 por cento, qual algoritmo de AM é melhor, dever-se assumir o caso geral para determinar se a diferença entre os dois algoritmos é significante ou não. Suponha que A_s é o algoritmo padrão e A_p o algoritmo proposto. A comparação é efetuada combinando-se a média e o desvio padrão de ambos os algoritmos.

1.3.3 Considerações Finais

Pelo que se pode observar, um Sistema Adaptativo requer, basicamente, dois recursos para realizar personalizações para os usuários. Primeiro, um modelo do domínio e/ou do usuário atualizado com dados significativos. Segundo, um motor de inferência aplicado sobre o modelo para ajudar o sistema a tomar decisões de adaptação.

A Inteligência Artificial oferece meios para a construção de sistemas capazes de raciocinar e tomar decisões de maneira autônoma. A estes sistemas costuma-se dar o nome de Agente. É com esta tecnologia que funciona uma boa parte dos motores de muitos sistemas adaptativos, em especial os Sistemas de Hipermídia Adaptativa da WWW.

Duas abordagens podem ser utilizadas em uma ampla variedade de sistemas inteligentes: a abordagem baseada em conhecimento e a abordagem de aprendizado ou comportamental. Esta Seção focaliza a abordagem de aprendizado não supervisionado por ser menos intrusiva. Outra vantagem consiste na criação de um modelo de usuário que é independente do domínio para o qual foi projetado. A próxima seção apresenta as principais tecnologias WWW e como elas podem contribuir para o desenvolvimento de Sistemas Adaptativos para a *World Wide Web*.

1.4 World Wide Web

A plataforma WWW estabelece um ambiente Hipermídia altamente distribuído formado por diversos elementos - o servidor HTTP, o navegador, a linguagem HTML e os recursos (páginas *Web*). Como é um serviço da Internet, a *World Wide Web* pode, ainda, atingir um número considerável de usuários em todo o mundo. Isto faz com que a WWW seja a plataforma preferida para o desenvolvimento e o teste de Aplicações Hipermídia principalmente nas áreas de Recuperação e Filtragem de Informação e Educação.

Devido à natureza aberta da *World Wide Web* e a formação do W3C, as tecnologias de suporte puderam desenvolver-se com bastante rapidez e consistência. Para a criação de Aplicações e Sistemas de Hipermídia Adaptativa é importante conhecer estas tecnologias para melhor aproveitá-las.

As folhas de estilo (*Cascad Style Sheet* - CSS) oferecem mecanismos poderosos para aumentar o controle sobre a apresentação e a expressividade do conteúdo. Com as tecnologias XML (*Extensible Markup Language*, acrescenta-se, ao mesmo tempo, poder de processamento sobre o conteúdo, controle na apresentação e na estruturação de documentos e facilidade na manipulação de ligações. Através das tecnologias que permitem ao servidor HTTP executar programas que processem informações, como CGI, ASP e *servlets*, podem ser criadas Aplicações Hipermídia dinâmicas resultantes do processamento de informações que trafegam pela Internet.

O conhecimento sobre o funcionamento do protocolo HTTP pode ser útil para prover extensões, na medida em que forem necessárias. É este protocolo que oferece a capacidade Hipermídia sobre a Internet. Com ele é possível realizar uma infinidade de combinações, gerando arquiteturas bastante variadas. Diversas extensões podem ser realizadas no protocolo HTTP para suportar características específicas e avançadas, como o controle de concorrência oferecido pelo WebDAV para permitir que vários usuários modifiquem e gerenciem arquivos de maneira colaborativa.

1.4.1 O Protocolo HTTP

O HTTP é um protocolo de requisição e resposta em nível de aplicação para sistemas de informação Hipermídia, colaborativo e distribuído. Pode ser usado de forma genérica, para comunicação entre agentes de usuários, ou como *proxies* ou *gateways* para outros sistemas Internet. Trata-se, portanto, de um protocolo que permite acesso não linear aos recursos disponíveis de diversas aplicações. Comunicações HTTP normalmente acontecem sobre conexões TCP/IP. A porta padrão é o TCP 80, mas outras portas podem ser usadas. O HTTP pode ser implementado no topo de qualquer outro protocolo sobre a Internet, ou sobre outras redes. Há uma série de arquiteturas e configurações de servidores sendo experimentadas com o emprego da WWW.

A carateristica principal do HTTP é o processo de requisição e resposta de recursos. Recurso é qualquer objeto de dado ou serviço disponível na WWW que pode ser disponibilizado em diferentes formatos e apresentar uma série de variações: uma página HTML, uma imagem, um *script* CGI, entre outros. Para localizar os recursos na *Web*, o servidor HTTP utiliza um mecanismo de localização de recursos na

Internet denominado URI[9]. Alguns autores, como (Guelich et al., 2001), consideram mais correto o uso do termo URL[10] em lugar de URI.

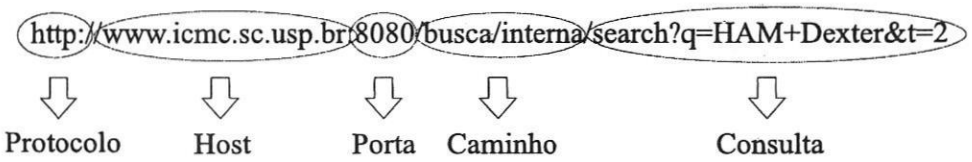

Figura 1.9: Esquema de uma URL.

Uma URL é definida por um esquema como mostra a Figura 1.9. Neste esquema, aparece o protocolo que inicia a comunicação. O host é o endereço ou o número IP do servidor WWW que provê o recurso. Opcionalmente pode ser especificada a porta para a conexão com o servidor HTTP. O caminho indica o local onde o recurso está armazenado. E consulta é uma cadeia contendo parâmetros adicionais na forma de pares de nomes e valores. Fragmento indica uma parte específica do recurso e não são enviados ao servidor. O cliente usa para localizar um determinado trecho do recurso recuperado, em geral uma âncora interna.

Um servidor HTTP recebe uma cadeia de requisição, registrando-a no arquivo de *log*. A requisição inclui o método usado para comunicação, a recurso solicitado e a versão do protocolo HTTP utilizada pelo navegador. Os métodos mais utilizados são GET e POST (Guelich et al., 2001). Em resposta à requisição do recurso, o servidor HTTP envia uma mensagem de resposta.

O conhecimento do protocolo HTTP é importante para o projeto de Sistemas Hipermidia Adaptativos por apresentar uma série de topicos que influenciam na construção do motor adaptativo. Dentre eles, destacam-se a arquitetura bastante variável dos servidores HTTP em uma sucessão de *proxies* e *caches* que pode dificultar a aquisição de dados; a predominância de conexões não persistentes e a possibilidade de estabelecer conexões persistentes com a versão 1.1 do protocolo; e as diversas configurações de armazenamento *cache* do servidor.

[9] *Universal Resource Identifiers* especificada em (RFC2396)
[10] *Universal Resource Locators*

1.4.2 Páginas Dinâmicas

Mensagens entre o cliente e o servidor podem ser processadas por programas externos ao HTTP. Desta maneira, é possível gerar páginas dinâmicas, isto é, páginas em que o conteúdo varia de requisição para requisição através da passagem de parâmetros pela cadeia de consulta.

O modo mais popular e conhecido de permitir que a WWW ofereça recursos dinâmicos é através da CGI (*Common Gateway Interface*). Trata-se de uma interface simples que estabelece uma comunicação entre o servidor HTTP e programas externos de modo que estes produzam páginas Web. Recursos dinâmicos são solicitados ao servidor WWW da mesma forma que recursos estáticos: através de mensagens formatadas de pedido. Em geral, servidores HTTP reservam um diretório comum ou uma extensão para CGIs.

A CGI é a interface pela qual servidor HTTP e o *software* externo se comunicam. O *software* pode ser escrito em uma linguagem de programação ou um *script* cuja entrada é obtida a partir da entrada padrão - STDIN - ou a partir de certas variáveis de ambiente. A saída de um normalmente é um conteúdo em HTML enviado para a saída padrão - STDOUT - ou um redirecionamento para outra URL.

O servidor HTTP acrescenta suas próprias linhas de cabeçalho na saída do *software* externo antes de retornar para o cliente que solicitou o recurso. Desta forma, a saída de um CGI não se parece de forma nenhuma diferente de qualquer outra resposta na Web. Ou seja, não é possível distinguir entre o acesso a um recurso gerado dinamicamente e um recurso estático (Guelich et al., 2001). Esta flexibilidade permite que se gere, virtualmente, qualquer coisa com um CGI.

Existem outras alternativas para a construção de páginas dinâmicas que procuram eliminar duas desvantagens dos CGIs: a execução de um processo separado para cada solicitação e a distinção entre páginas HTML e o código que processa as informações. Estas tecnologias são: ASP (*Active Server Pages*), PHP, *ColdFusion*, Java *Servlets*, FAST CGI e *mod_perl*.

1.4.3 XML

As tecnologias XML[11] oferecem, juntas, um poder adicional de processamento, estruturação e apresentação de documentos no ambiente WWW. Por esta razão, apresentam-se como forte candidata para o desenvolvimento de Sistemas Hipermidia Adaptativos.

Definindo-se uma estrutura e uma semântica comum aos documentos, é possível estabelecer visões distintas de um mesmo conteúdo. Esse conteúdo pode ser facilmente armazenado ou transformado em outros formatos, inclusive em bancos de dados. O processo inverso também pode ser efetuado com certa facilidade, isto é, obter um arquivo em formato XML a partir de um banco de dados.

Apenas recentemente os pesquisadores têm investigado o uso das tecnologias de XML em Sistemas de Hipermídia Adaptativa. Um dos maiores empecilhos reside no fato de que as empresas que desenvolvem os navegadores precisam implementa-las e que nem sempre elas obedecem às especificações originais da *World Wide Web Consortium*.

1.4.4 Considerações Finais

A produção de Sistemas Adaptativos no ambiente WWW implica no estabelecimento das características arquiteturais do sistema e na escolha das tecnologias mais apropriadas. Cada uma delas apresenta vantagens e desvantages que precisam ser analisadas cuidadosamente. Além disto, deve-se considerar a natureza da aplicação que se pretende desenvolver.

Uma vez que as características básicas tenham sido determinadas, é preciso cuidar do motor adaptativo propriamente dito. Deve-se escolher a melhor abordagem para cada situação, pois não existe solução única que resolva todos os problemas de maneira satisfatória. Neste aspecto, devem ser escolhidas as técnicas de aquisição, representação e produção da adaptação, bem como o tipo de raciocínio que será usado para implementar o motor adaptativo.

Por estas razões é que contamos hoje com uma grande diversidade de Sistemas de Hipermídia Adaptativo no ambiente WWW com técnicas variadas. Isto dificulta

[11] XMLSchema, XSL, RDF e *parsers* XML, entre outros

a classificação e a comparação entre diferentes sistemas e aplicações de Hipermídia Adaptativa.

Como se pode notar, a Hipermídia Adaptativa é uma área difícil que abrange conhecimentos e técnicas provenientes de outras áreas da Computação. Um pesquisador pode dedicar-se as aspecto arquitetural, enquanto outro preocupa-se com a criação de uma abordagem para o desenvolvimento de sistemas e aplicaçãoes de HA. Alguns poderão dedicar-se às técnicas de adaptação de conteúdo, enquanto outros poderão focalizar as técnicas de adaptação de ligações.

Referências Bibliográficas

(Claypool et al., 2001) Claypool, M.; Waseda, M.; Le, P.; Brown, D. Implicit Interest Indicator. Technical Report, Worcester Polytechnic Institute, Massachusetts, 01609-2280, 2001.

(Baranauskas & Monard, 2000) Baranauskas, Joé A.; Monard, Maria Carolina. *Revising Some Machine Language Concepts and Methods.* Relatório Técnico RT102, ICMC/USP, São Carlos, Fevereiro, 2000.

(Brusilovsky, 1995) Brusilovsky, Peter. *Intelligent Tutoring Systems for World-Wide Web. Proceedings of Third International WWW Conference, Darmstadt, pp. 42-45, 1995.*

(Brusilovsky, 1996a) Brusilovsky, P. *Methods and Techniques of Adaptive Hypermedia.* User Modeling and User-Adapted Interaction 6, Kluwer Academic Publishers, pp. 87-129, 1996.

(Brusilovsky, 1996b) Brusilovsky, P. *Adaptive Hypermedia: an Attempt to Analyze and Generalize.* P. Brusilovsky, P. Kommers and N. Streitz (eds.), Multimedia, Hypermedia, and Virtual Reality - Lecture Notes in Computer Science, vol. 1077. Berlin: Springer-Verlag, pp. 288-304, 1996.

(Brusilovsky, 2000) Brusilovsky, Peter. *Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education.* G. Gauthier, C. Frasson and K. VanLehn (eds.) Intelligent Tutoring Systems - Lecture Notes in Computer Science, vol. 1839, Berlin: Spring Verlag, pp. 1-7, 2000.

(Brusilovsky, 2001) Brusilovsky, Peter. *Adaptive Hypermedia.* User Modeling and User-Adapted Interaction 11, pp. 87-110, Netherlands: Kluwer Academic Publisher, 2001.

(CERN) http://public.web.cern.ch/Public/ACHIEVEMENTS/web.html.

(De Bra, 1999) De Bra, Paul. *Design Issues in Adaptive Web-Site Development.* Proceedings of 2nd Workshop on Adaptive Systems and User Modeling on the WWW, 1999. http://wwwis.win.tue.nl/asum99/debra/debra.html.

(De Bra et al., 1999a) De Bra, Paul; Houben, Geert-Jan; Wu, Hongjing. *Authoring Support for Adaptive Hypermedia Applications.* Proceedings of ED-MEDIA'99, Seattle, pp. 364-369, 1999.

(De Bra et al., 1999b) De Bra, Paul; Houben, Geert-Jan; Wu, Hongjing. *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia.* Proceedings of ACM Hypertext'99 Conference, Darmstadt, Alemanha, pp. 147-156, 1999.

(De Bra et al., 2000) De Bra, Paul; Houben, Geert-Jan; Wu, Hongjing. *Supporting User Adaptation in Adaptive Hypermedia Applications.* Online Conference and Informatiewetenschap, Rotterdam, 2000. Disponível no endereço Internet http://wwwis.win.tue.nl/ houben/respub/infwet00.pdf.

(Guelich et al., 2001) Guelich, Scott; Gundavaram, Shishir; Birznieks, Gunther. *Programação CGI com Perl.* Rio de Janeiro: Editora Ciência Moderna, 2001.

(Kobsa, 1993) Kobsa, A. *User Modeling: Recent Work, Prospects and Hazards.* H. Schneider Hufschmidt, T. Kuhme and U. Malinowski (eds.): Adaptive Users Interfaces: Principles and Practice. Amsterdam, North-Holland, pp. 111-123.

(Kobsa, 1994) Kobsa, A. *User Modeling and User-Adapted Interaction.* CHI'94 Tutorial Notes. URL: http://www.acm.org/sichi/chivas/kobsa.html.

(Kobsa et al., 1999) Kobsa, A; Koenemann, J.; Pohl, W. *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships.* Technical Report No. 66 GMD, German National Research Center for Information Technology, St. Augustin, Alemanha, 1999.

(Marietto, 2000) Marietto, Maria das Graças. *Definição Dinâmica de Estratégias Institucionais em Sistemas de Tutoria Inteligente: Uma Abordagem Multiagentes na WWW.* Tese de Doutorado, ITA, 2000.

(Middleton, 2001) Middleton, Stuart E. *Capture Knowledge of User Preferences with Recommender Systems.* Universidade de Southampton, Julho, 2001, Tese de PhD.

(Mitchell, 1997) Mitchell, Tom M. *Machine Learning.* New York: McGraw-Hill,

1997.

(Mullier, 1995) Mullier, D. J. *A Hybrid Connectionist/Knowledge-Based Approach to Intelligent Tutoring with Hypermedia.* Acessado em 17/05/2002 na URL: http://www.lmu.ac.uk/PLYMOUTH.htm.

(Mullier et al., 1999) Mullier, D. J.; Hobbs, D. J.; Moore, D .J. *A Hybrid Semantic/Connectionist Approach to Adaptivity in Educacional Hypermedia Systems.* Proceedings of ED-MEDIA'99, Seattle, WA. http://www.lmu.ac.uk/ies/comp/staff/dmullier/icai.pdf.

(Nichols, 1987) Nichols, David N. Implicit Rating and Filtering. In: Proceedings of the 5^{th} DELOS Workshop on Filtering and Collaborative Filtering, 10-12, Buddapest, Hungria, ERCIM, 1987.

(Oard & Kim, 1998) Oard, D. W.; Kim, J. Implicit Feedback for Recommender Systems. Proceedings of the AAAI Workshop on Recommender Systems, July, 1998.

(Russel & Norvig, 1995) Russel, Stuart; Norvig, Peter. *Artificial Intelligence: A Modern Approach.* New Jersey: Prentice Hall, 1995.