

2 Fundamentos Teóricos

2.1. Hipermídia adaptativa

2.1.1. Definição

Pode-se considerar que sistemas hipermídia são, de certa forma, sempre adaptativos, uma vez que ao proporcionarem liberdade de navegação aos diferentes usuários, estão possibilitando que eles adaptem o sistema às suas necessidades de informação, ou seja, o próprio usuário é responsável pela adaptação. Entretanto, dependendo da área de aplicação, é importante que o próprio sistema se adapte ao usuário (Brusilovsky, 1996a), caracterizando a hipermídia adaptativa.

O objetivo da hipermídia adaptativa é aumentar a funcionalidade da hipermídia, considerando o contexto de utilização. Sistemas hipermídia são desenvolvidos sem levar em consideração as características de uso real do sistema. Assim, independentemente, por exemplo, dos objetivos ou conhecimento prévio dos usuários, o sistema vai se comportar sempre da mesma maneira durante sua utilização, diminuindo sua eficácia. Como consequência, pode acontecer de um usuário demorar muito a encontrar o que procura ou, até mesmo, desistir da navegação antes de atingir o seu objetivo.

Segundo Brusilovsky (1996a), AHSs constroem um modelo de objetivos, preferências e conhecimento de cada usuário individualmente e usam este modelo durante a interação com o usuário para adaptar o sistema às necessidades deste usuário. Como já visto, além do Modelo do Usuário, AHSs, em geral, utilizam também modelos do Domínio (para estruturar o conteúdo) e de Adaptação.

AHSs podem ser úteis em áreas de aplicação onde o hiperespaço é razoavelmente grande e o sistema é utilizado por pessoas com objetivos e conhecimento diversos, interessadas em diferentes trechos de informação apresentados em uma página hipermídia. De acordo com o UM construído, os

sistemas podem ajudar os usuários na interação restringindo o espaço de navegação, sugerindo elos relevantes a serem seguidos ou adicionando comentários aos elos, para citar alguns exemplos.

2.1.2. Técnicas mais utilizadas

Brusilovsky (1996a) define algumas técnicas de adaptação, reconhecidas pela maioria dos autores. Entre as mais utilizadas, podemos citar:

- Inclusão condicional de fragmentos (*conditional fragments*): é feita uma seleção de quais informações devem ser mostradas;
- Visita guiada ou orientação direta (*direct guidance*): o sistema indica o melhor caminho a seguir através de um botão (*button*) “próximo”, por exemplo;
- Anotação de elos (*link annotation*): as âncoras dos elos são apresentadas com aparências diferentes para indicar o grau de relevância. Uma metáfora bastante utilizada é a do “sinal de trânsito” (*traffic light metaphor*), onde a cor vermelha indica que os elos não devem ser explorados, a cor verde é usada para elos recomendados e a cor amarela é utilizada para indicar elos que podem ser visitados, mas não são recomendados;
- Elos escondidos (*link hiding*): é um caso especial de anotação de elos. O elo existe, mas a âncora não pode ser identificada como tal: tem a mesma aparência que o texto;
- Ordenação de elos (*link sorting*): os elos são apresentados em ordem decrescente de relevância. A desvantagem desta técnica é que a mesma página pode aparecer diferentemente a cada acesso do usuário, muitas vezes causando desorientação.

2.1.3. Principais Questões

A caracterização dos AHSs pode ser feita respondendo-se a quatro questões principais: “o quê” é adaptado; “em função de quê” a adaptação é feita; “como” ela é feita e “quando” (em que granularidade de tempo) ela ocorre. Esta última dimensão é, na verdade, uma resposta a “em função de quê” tempo a adaptação

ocorre. Cada questão tem várias respostas possíveis; alguns AHSs consideram todas, enquanto outros abordam apenas algumas respostas.

2.1.3.1.

O quê é adaptado

Segundo Brusilovsky (1996a), sistemas hipermídia consistem de páginas conectadas por elos, podendo incluir índices e um mapa global com elos para todas as páginas disponíveis. Cada página contém algumas informações locais e elos para páginas relacionadas. Por conseguinte, a adaptação pode ocorrer no conteúdo das páginas (*content-level adaptation*) e nos elos: das páginas, dos índices e dos mapas (*link-level adaptation*). Via de regra (Brusilovsky, 1996b), o primeiro tipo de adaptação (nível de conteúdo) é utilizado para atender a diferentes classes de usuários, sendo classificado como adaptação da apresentação. Já a adaptação que ocorre nos elos é considerada como adaptação do suporte à navegação, já que seu objetivo é auxiliar a navegação dos usuários, evitando que eles se percam no hiperespaço. O suporte à navegação pode ocorrer através da limitação do espaço de navegação, da sugestão dos elos mais relevantes e da anotação dos elos (Brusilovsky, 2003).

Koch (2000) considera, ainda, uma adaptação no nível da apresentação, englobando as alterações no *layout* que não afetam o conteúdo. Assim, enquanto Brusilovsky (1996b) relaciona apenas as adaptações da apresentação e do suporte à navegação, Koch (2000) considera os seguintes tipos de adaptação:

- Adaptação de conteúdo (seleção diferenciada de informação, pela utilização, por exemplo, de conteúdos alternativos e adição ou ocultação de conteúdo);
- Adaptação de navegação (mudanças nas aparências dos elos, nos destinos dos elos, no número de elos ou na ordem em que estes elos são apresentados ao usuário);
- Adaptação de apresentação (*design* diferente dos elementos de interface tais como: tipo de mídia; ordenação; cores; tipo da fonte; tamanho das imagens).

Entretanto, como será visto adiante (seção 2.2), a navegação pode ser considerada como sendo composta por nós e elos, onde um nó representa uma “visão” (como na área de banco de dados) sobre a classe conceitual

correspondente e, analogamente, os elos sobre as relações conceituais. A idéia geral baseia-se no método OOHDM que postula uma separação clara entre conteúdo, navegação e apresentação. Assim, e com o objetivo de caracterizar mais precisamente o quê pode ser adaptado – ou seja, qual o resultado real da adaptação, qual aspecto da aplicação muda –, propõe-se aqui uma diferenciação entre a adaptação do teor do conteúdo e a adaptação da estrutura do conteúdo, esta última considerada como um tipo de adaptação de navegação. Distingue-se, ainda, um outro tipo de adaptação de navegação: a adaptação da topologia do hiperespaço (alterações nos destinos dos elos ou na quantidade de elos). Em relação à apresentação (interface), separa-se as alterações na aparência do conteúdo das alterações na aparência dos elos. A Tabela 1 apresenta a classificação proposta.

O QUÊ É ADAPTADO		DESCRIÇÃO e EXEMPLOS
Conteúdo		O conteúdo propriamente dito. <u>Exemplos:</u> texto coloquial para leigos <i>versus</i> texto técnico para especialistas; textos simples para novatos <i>versus</i> textos detalhados para usuários experientes.
Navegação	Estrutura do Conteúdo	Nós – a maneira pela qual os conceitos são colocados juntos para propósitos de navegação. <u>Exemplos:</u> inclusão da introdução para novatos <i>versus</i> pular introdução para usuários avançados; inclusão de advertências de segurança da primeira vez que um conteúdo crítico é visitado <i>versus</i> a não inclusão em visitas subseqüentes.
	Topologia do hiperespaço	Elos – alterações nas âncoras e índices modificam os caminhos de navegação. <u>Exemplos:</u> inclusão de um elo para a “solução” de um problema utilizado como exemplo <i>versus</i> a não inclusão se o problema for utilizado como teste.
Apresentação	Interface do Conteúdo	Fragmentos. <u>Exemplos:</u> utilização de fontes de tamanhos diferentes de acordo com a idade do usuário; emprego de realce para enfatizar certos tipos de informação; uso de mídias diferentes (<i>cross-media</i>).
	Interface de suporte à navegação	Âncoras. <u>Exemplos:</u> utilização de menu <i>drop-down</i> <i>versus</i> utilização de lista explícita de âncoras; uso de âncoras textuais <i>versus</i> uso de ícones como âncoras.

Tabela 1 – “O quê”

2.1.3.2.

Em função de quê é feita a adaptação

Esta questão diz respeito aos vários parâmetros que podem ser usados para determinar a adaptação. A Tabela 2 detalha os aspectos básicos que podem ser

levados em consideração na adaptação.

ASPECTO	EXPLICAÇÃO E EXEMPLO
Modelo do Domínio (<i>Domain Model</i>)	A maneira como o conteúdo é estruturado no DM pode afetar a adaptação. <u>Exemplo:</u> um conceito (hierarquicamente) composto ser considerado como aprendido somente quando todos os conceitos elementares que o compõem tiverem sido aprendidos.
Perfil do Usuário (<i>profile</i>) (armazenado no UM)	Preferências e características do usuário; papel desempenhado por ele. <u>Exemplos:</u> se o usuário é novato ou experiente; se ele prefere textos ou imagens; em que língua ele prefere acessar; se ele é estudante ou professor; se ele se identificou ou está acessando como anônimo; se ele gosta mais de acessar textos inteiros ou resumos; conhecimento dele sobre o domínio; objetivos dele.
Contexto (Ambiente) ¹ (<i>Context</i>) (Situação de uso)	Sob que condições o sistema é utilizado. <u>Exemplos:</u> se o usuário está navegando a partir de um computador pessoal ou a partir de um telefone celular; de qual lugar (tais como: rua, bairro, cidade, casa, universidade, escritório) o acesso está sendo feito; qual a banda de acesso.
Histórico de Navegação (<i>User Navigational Behavior</i>)	Caminho que o usuário percorre enquanto navega. <u>Exemplo:</u> nós de navegação acessados previamente.
Histórico de Interação (<i>User Browsing Behavior</i>)	As ações que o usuário executa (interações em nós de navegação) enquanto navega. <u>Exemplos:</u> quais artigos o usuário só lê e quais imprime; quantas vezes ele vê um vídeo; que imagens ele amplia.
Utilização da Funcionalidade da Aplicação (<i>Application Functionality Usage</i>)	Como a funcionalidade da aplicação está sendo usada. <u>Exemplos:</u> resultados de testes; itens comprados previamente.
Granularidade de Tempo (<i>Granularity of Time</i>) (Quando)	A adaptação é sempre feita em algum instante ou período de tempo. A questão é em que granularidade este tempo é considerado, podendo ir desde uma adaptação pontual até episódica até contínua. <u>Exemplos:</u> adaptação feita no <i>login</i> (customização ²); adaptação após uma configuração do usuário; revisão e adaptação do sistema a cada passo da navegação.

Tabela 2 – "Em função de quê"

Cabe observar que a identificação do histórico de interação como fator de influência, bem como o discernimento dos parâmetros, também foram ensejados pela proposta do OOHDMM de separação entre conteúdo, navegação e apresentação.

¹Neste caso, "contexto" é empregado no sentido de situação / ambiente de uso e não no sentido mais amplo que engloba, além do dispositivo, o perfil, navegação, localização, etc.

² Neologismo largamente utilizado no sentido de "personalizar", "fazer sob encomenda".

2.1.3.3. Como é feita a adaptação

Esta questão diz respeito aos mecanismos utilizados na execução da adaptação:

- Modelo do Usuário;
- Modelo de Adaptação;
- Agente de entrada (*input*);
- Agente de atualização (*update*).

Cada sistema possui uma maneira específica de implementar um ou mais destes mecanismos. O Modelo do Usuário pode ser implementado, por exemplo, como pares atributo-valor (Wu et al., 2000a, 2000b); modelo bayesiano (Henze & Nejdí, 1999b); abordagem probabilística e episódica (*probabilistic and episodic approach*) (Specht & Oppermann, 1998); rede de Petri (Medina-Medina et al., 2002); redes semânticas; arquivos de transações (*log files*); tabela em bancos de dados relacionais; classes orientadas a objetos. Algumas vezes, o Modelo do Usuário se sobrepõe ao Modelo do Domínio (*overlay model*).

O Modelo de Adaptação pode ser implementado, por exemplo, usando regras, onde as pré-condições (histórico, perfil, localização, etc.) definem o “em função de quê” e as pós-condições determinam “o quê” é alterado / adaptado.

A captura e atualização dos dados podem ser automáticas ou manuais, feitas pelo usuário ou pelo autor. A informação sobre o usuário pode ser obtida por observação ou o usuário pode preencher um formulário.

2.1.3.4. Quando é feita a adaptação

Algumas características do usuário podem ser determinadas antes da utilização do sistema. A adaptação feita em função de tais características é denominada customização. Neste caso, as alterações ocorrem antes do efetivo uso do sistema pelo usuário e, uma vez iniciada a sua utilização, a aplicação permanece inalterada. Considera-se que a adaptação propriamente dita acontece quando as características influenciam dinamicamente as alterações no sistema. Como visto (Tabela 2), a granularidade de tempo, ou seja, “em função de que” tempo a adaptação ocorre, pode ser um parâmetro usado para determinar a adaptação.

2.1.4. Meta-adaptação

Considera-se adaptação aquela que ocorre nos documentos. Já a meta-adaptação diz respeito às alterações (i.e, adaptações) que podem ser aplicadas ao próprio mecanismo de adaptação (Francisco-Revilla & Shipman, 2003). Mais especificamente, considera-se como meta-adaptação as adaptações que ocorrem nos modelos propriamente ditos. No Modelo de Adaptação, tanto pode ser alterado o modelo em si (a granularidade da adaptação, por exemplo), como os mecanismos que o implementam (por exemplo, regras).

A importância da meta-adaptação é confirmada pela observação de que algumas técnicas de adaptação funcionam mais eficientemente em certas circunstâncias e menos em outras. Em particular no domínio da educação, a informação sobre estilos de aprendizagem pode contribuir consideravelmente para a decisão das técnicas de adaptação mais adequadas ao perfil do usuário, já que as categorizações específicas dos referidos estilos parecem se adequar melhor a tecnologias de adaptação específicas (Papanikolaou & Grigoriadou, 2004). O ideal seria que Sistemas Hiperídia Adaptativos tivessem à disposição (a maioria das) técnicas de adaptação, além do entendimento sobre a aplicabilidade e a limitação de cada uma delas, individualmente ou combinadas entre si (Ahmad et al., 2004), como fatores para a seleção dinâmica das técnicas mais adequadas.

A necessidade de meta-adaptação é maior quando há adaptação ao estilo pessoal, à experiência e aos objetivos do usuário (por exemplo, aprendizado ou referência) (Brusilovsky, 2000). Tais objetivos podem, inclusive, ser usados para determinar as características que precisam ser incluídas no Modelo do Usuário. Ou seja, de acordo com os objetivos, define-se o tipo de UM que será usado.

2.2. OOHDM e SHDM

O OOHDM (Rossi et al., 1999) é um método que utiliza uma abordagem baseada em modelos para projetar aplicações *Web* em um processo composto de cinco etapas: Levantamento de Requisitos, Modelagem Conceitual, Modelagem Navegacional, Projeto da Interface Abstrata e Implementação. A cada etapa, um conjunto de modelos orientados a objetos é construído ou enriquecido,

descrevendo detalhes do projeto.

O SHDM (Lima, 2003) enriquece o OOHDm com novos mecanismos inspirados nas linguagens propostas para a *Web Semântica*. Na etapa do Projeto Conceitual, o domínio da aplicação é estudado. Durante o Projeto Navegacional são identificadas as tarefas a serem executadas. O Modelo Navegacional obtido é derivado do Modelo Conceitual a partir de um mapeamento explícito e representa as possíveis visões sobre objetos conceituais que atendem às necessidades de usuários com perfis distintos ou desempenhando tarefas diferentes.

2.2.1.

Modelo Conceitual

O resultado da etapa de modelagem conceitual do OOHDm é um esquema conceitual contendo os objetos do domínio da aplicação (basicamente, classes e relacionamentos). Em sua proposta original, o SHDM estende o OOHDm usando um modelo orientado a objetos para a especificação do esquema conceitual e uma ontologia conceitual para a representação desse esquema em um formato implementável na *Web Semântica*. Os dados propriamente ditos são representados por instâncias dessa ontologia. Szundy (2004) emprega diretamente o modelo RDF como a base estrutural do esquema conceitual, permitindo “a utilização de qualquer ontologia definida para a *Web Semântica* (em OWL ou RDFS) como esquema conceitual da aplicação, sem a necessidade de adaptações”. O esquema e a ontologia conceituais são, desta forma, unificados. Essa abordagem, em muitos casos, evita um trabalho adicional de adaptação dos dados para a representação específica do método.

2.2.2.

Modelo de Navegação

Na abordagem utilizada pelo método SHDM, considera-se que usuários com perfis distintos ou desempenhando tarefas variadas podem necessitar de diferentes visões navegacionais sobre um mesmo conjunto de informações. Portanto, uma aplicação hipermídia define uma visão navegacional sobre dados da *Web Semântica*, que representam o modelo conceitual da aplicação. A definição desta visão é feita com base na análise de cenários e casos de uso e descreve as tarefas a serem desempenhadas pelos usuários, identificando as informações e operações envolvidas nessas tarefas. Já Szundy (2004) considera que as visões navegacionais

da aplicação são especificadas sobre modelos conceituais definidos por quaisquer ontologias da *Web Semântica*. Assim, o modelo navegacional é derivado do modelo conceitual segundo um mapeamento explícito que estabelece uma relação entre os dados de ambos os modelos. A especificação do Modelo de Navegação é feita por um vocabulário definido como uma ontologia e independe da implementação.

O Modelo de Navegação se divide em dois esquemas:

- Esquema de Classes Navegacionais, englobando a definição de **classes e elos**, que estabelece quais informações serão apresentadas;
- Esquema de Contextos Navegacionais que, através da definição dos **contextos navegacionais e estruturas de acesso**, determina como será a navegação pelas informações apresentadas.

Classes navegacionais definem quais dados do modelo conceitual poderão ser acessados. São especificadas em função de uma classe conceitual (classe base) através de um mapeamento que estabelece quais instâncias da classe conceitual serão também instâncias da classe navegacional e quais os atributos da classe navegacional. Os valores destes atributos podem ser valores simples; listas; âncoras ou índices. Em termos práticos, cada atributo corresponde a um padrão de recuperação de dados, a partir de uma instância da classe navegacional.

Elos definem os possíveis caminhos de navegação da aplicação e podem ser determinados pelo mapeamento de propriedades conceituais; pela declaração de âncoras em estruturas de acesso ou classes navegacionais; ou pela definição de padrões de navegação intracontextuais.

Contextos navegacionais são conjuntos de nós selecionados em função de alguma característica comum, como serem do mesmo tipo ou apresentarem determinado valor para um atributo. O padrão de navegação interna determina como os nós podem ser acessados uns a partir dos outros. Pode ser especificada uma ordenação para a exploração dos elementos do contexto.

Estruturas de acesso, ou índices, representam conjuntos de âncoras que apontam para outras estruturas de acesso ou nós dentro de contextos específicos. As informações são apresentadas como entradas de dados que contêm atributos simples ou âncoras (pelo menos uma). Podem estar organizadas hierarquicamente e podem ser ordenadas segundo valores dos atributos. Os índices podem ser derivados de consulta; derivados de contexto; arbitrários ou facetados.

Quando estruturas de acesso ou nós em um contexto podem ser acessados diretamente de qualquer ponto da aplicação, é possível definir-se um *landmark*, que é um elo para o elemento em questão. O *landmark* é declarado como uma âncora, definida uma única vez para todo o modelo navegacional.

2.2.3. Modelo de Interface

A proposta de Moura (2004) separa os aspectos essenciais, independentes de tecnologia e de padrões de implementação (interface abstrata), dos aspectos específicos de cada ambiente de execução (interface concreta). Tanto o modelo de interface abstrata quanto o de interface concreta especificam os *widgets* (objetos de interface) que estarão disponíveis ao usuário e que permitem a sua interação com o sistema.

A modelagem da interface abstrata é definida por uma Ontologia de *Widgets* Abstratos que descreve os possíveis elementos que representam as interações entre o usuário e a aplicação. A partir do modelo navegacional e da análise de requisitos, o projetista consegue identificar todas as necessidades de trocas de informação que precisam acontecer entre o usuário e a aplicação, podendo, assim, decidir quais *widgets* serão os mais adequados para tal interação. A instanciação dessa ontologia é que caracteriza abstratamente uma interface. A interface abstrata pode ser mapeada para uma interface concreta, composta por *widgets* concretos. Estes, por sua vez, são instâncias de uma Ontologia de *Widgets* Concretos e podem ser diretamente traduzidos para elementos de interface com suporte em ambientes de implementação disponíveis como, por exemplo, HTML.

O Modelo de Interface é, portanto, definido por uma arquitetura de *widgets* abstratos, pelo mapeamento do abstrato para o concreto e por *widgets* concretos.

A Ontologia de *Widgets* Abstratos proposta por Moura (2004) é composta por 11 conceitos que pretendem representar todas as interações do usuário com o sistema, dando suporte às entradas do usuário, às saídas do sistema e às operações solicitadas pelo usuário. As entradas e saídas tanto podem ser itens de dados como estruturas fornecidas pelo usuário ou retornadas pelo sistema, respectivamente. A ontologia possui, ainda, 12 propriedades³. Esta ontologia (Figura 3) descreve

³Divididas em `ObjectProperty` (`hasInterfaceElement`, `targetInterface`, `mapsTo`) e `DatatypeProperty` (`blockElement`, `isRepeated`, `compositionTag`, `fromAnchor`, `fromContext`,

como os objetos navegacionais serão apresentados, especificando os elementos perceptíveis que estarão disponíveis para o usuário⁴.

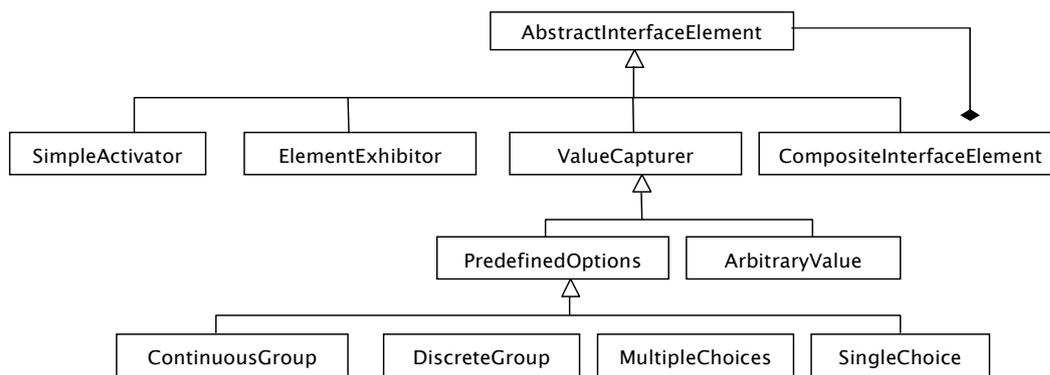


Figura 3 – Ontologia de *Widgets* Abstratos

A Ontologia de *Widgets* Concretos – apresentada na Figura 4 – descreve, de maneira simples os elementos concretos de interface das aplicações hipermídia. O objetivo é facilitar a definição de qual elemento concreto corresponde a cada elemento abstrato, possibilitando que a interface abstrata seja independente da tecnologia. “Os elementos descritos nessa ontologia são os mais utilizados atualmente nas aplicações; no entanto, na URL <http://www.xulplanet.com> é possível encontrar outros elementos” (Moura, 2004).

```

<ConcreteInterfaceElement rdf:ID="Button" />
<ConcreteInterfaceElement rdf:ID="CheckBox" />
<ConcreteInterfaceElement rdf:ID="CheckBoxAction" />
<ConcreteInterfaceElement rdf:ID="ComboBox" />
<ConcreteInterfaceElement rdf:ID="ComboBoxAction" />
<ConcreteInterfaceElement rdf:ID="ComboBoxTarget" />
<ConcreteInterfaceElement rdf:ID="Composition" />
<ConcreteInterfaceElement rdf:ID="Form" />
<ConcreteInterfaceElement rdf:ID="Image" />
<ConcreteInterfaceElement rdf:ID="Label" />
<ConcreteInterfaceElement rdf:ID="Link" />
<ConcreteInterfaceElement rdf:ID="RadioButon" />
<ConcreteInterfaceElement rdf:ID="RadioButonAction" />
<ConcreteInterfaceElement rdf:ID="RadioButonTarget" />
<ConcreteInterfaceElement rdf:ID="TextArea" />
<ConcreteInterfaceElement rdf:ID="TextBox" />
  
```

Figura 4 – Instâncias que representam os elementos concretos

A interface concreta é gerada pelo mapeamento de cada elemento abstrato da instância de interface abstrata, para um elemento concreto, definido pela Ontologia de *Widgets* Concretos. Este mapeamento segue uma regra de consistência específica que determina quais os elementos concretos em que o

fromIndex, fromAttribute, fromElement, visualizationText).

elemento abstrato pode ser mapeado. A propriedade *mapsTo*, presente em todas as classes da Ontologia de *Widgets* Abstratos, é utilizada para indicar o elemento da ontologia de *widgets* concretos selecionado.

Depois da especificação da interface abstrata e do mapeamento de seus elementos, pode-se definir um *layout* para a interface. A proposta de Moura (2004) utiliza o padrão CSS, mais especificamente, o modelo de caixas (*box model*). São fornecidas *tags* HTML e classes do CSS que definirão uma “caixa” CSS para cada elemento descrito na interface abstrata. Assim, para a “definição de um *layout*, para qualquer elemento abstrato, é necessário apenas definir *tags* HTML e classes CSS na propriedade ‘blockElement’ do elemento abstrato” (Moura, 2004). A integração dos elementos do CSS ao HTML é feita na geração da página.

2.3.

Padrões para Representação de Informações sobre Usuário

2.3.1. Modelagem do Usuário

A construção e manutenção de Modelos de Usuário para adaptação pode ser um processo muito trabalhoso. A reutilização de tais modelos entre aplicações permite que essa tarefa seja dividida entre diversos sistemas, desonerando uma aplicação em particular. O reuso de UMs também é interessante do ponto de vista do usuário, trazendo consistência entre aplicações e liberando-o de treinar novos sistemas. É razoável que após um tempo usando uma aplicação, melhorando o UM e, conseqüentemente, a habilidade de interagir efetivamente com o sistema, o usuário espere que uma outra aplicação use os aspectos relevantes do mesmo UM (Kay et al., 2002; Kay, 2000).

Ademais, quando a aplicação é usada pela primeira vez, valores *default* são assumidos. Conforme ela vai sendo utilizada, o UM vai ficando mais acurado e a adaptação mais eficaz. Uma vez que freqüentemente o UM é gerenciado pela própria aplicação, como parte interna do sistema, é necessário repetir o processo de inicialização em cada aplicação utilizada pelo usuário (Kuruc, 2005).

Por conseguinte, e como cada vez mais os usuários trabalham simultaneamente com múltiplas aplicações para a execução de tarefas diversas

⁴Para uma explicação sobre os conceitos e propriedades, consulte (Moura, 2004).

(Chepegin, 2004), o foco das pesquisas na área de modelagem de usuários tem se voltado para a separação do UM das aplicações hipermídia adaptativas, com a criação de bases centrais de conhecimento⁵ sobre o usuário que possam ser utilizadas por diferentes sistemas. Aplicações do mesmo domínio ou de domínios similares cooperariam com a mesma parte do UM (Kuruc, 2005).

Grosso modo, a utilização de tais bases de conhecimento pressupõe a existência de mecanismos adequados para:

- Capturar as informações sobre o usuário, baseadas na interação com os sistemas;
- Predizer interesses e preferências de usuários de acordo com vários possíveis mecanismos, tais como estereótipos;
- Representar e armazenar essas informações de forma consistente, segura e atualizada;
- Possibilitar que o usuário inspecione e altere o próprio modelo;
- Fornecer as informações do usuário para os sistemas, quando solicitado, respeitando a privacidade.

Esses mecanismos, em geral, são implementados por servidores de UM e fogem ao escopo deste trabalho. Para maiores informações sobre servidores e compartilhamento de UMs, os leitores devem se referir a (Kobsa & Pohl, 1995; Kobsa, 2001; Fink & Kobsa, 2002; Kay, 2000; Kay et al., 2002; Kuruc, 2005; Chepegin, 2004). Dentro do escopo deste trabalho, interessa saber, basicamente, quais informações são representadas, como elas são modeladas e como se dá a atualização do UM.

A reutilização de um UM pressupõe, ainda, que o modelo seja expresso em termos de um vocabulário coerente, que permita aos programadores de diferentes sistemas descrever o usuário com termos padronizados. Esta padronização pode ser feita por uma ontologia de alto nível, comumente aceita, representada em uma linguagem da *Web* semântica, ficando disponível simultaneamente, via Internet, para as aplicações adaptativas. Uma proposta é o GUMO (*General User Model Ontology*) (Heckmann et al., 2005).

⁵As bases de conhecimento podem ser implementadas de diversas formas, tais como: Sistemas de Representação do Conhecimento (*Knowledge Representation Systems*), Sistemas de Gerenciamento de Dados (*Database Management Systems*) ou (Sistemas de Gerenciamento de Diretórios (*Directory Management Systems* ou, simplesmente, *Directories*).

Na área educacional, diversos estudos vêm sendo feitos, no sentido de se padronizar as informações sobre o usuário que devem ser mantidas pelos sistemas. Segundo Dolog et al. (2003), os dois exemplos mais importantes destes padrões são o PAPI (*Public And Private Information*) (PAPI, 2002) e o IMS LIP (*Learner Information Package*) (IMS, 2001). Ambos sugerem categorias de informações sobre o usuário, mas enquanto o primeiro reflete idéias de Sistemas de Tutoria Inteligentes, priorizando informações sobre o desempenho e ressalta a importâncias das relações interpessoais, o segundo se baseia na noção de currículo e não considera tais relações (Dolog & Nejd, 2003).

O CC/PP (*Composite Capabilities/Preferences Profile*) (CC/PP, 2004) é um outro padrão, que se propõe a representar as capacidades dos dispositivos e as preferências dos usuários.

Uma idéia geral de cada um dos padrões citados é apresentada a seguir. Vale observar que, embora esses padrões existam como especificações completas de implementações, são destacados apenas os aspectos relativos às informações representadas, relevantes ao escopo deste trabalho.

2.3.2. PAPI

O objetivo maior deste padrão é servir como uma especificação para troca de dados, ou seja, comunicação entre sistemas cooperativos, determinando a sintaxe e a semântica de um “Modelo do Aluno” (*Learner Model*) que caracterizam o aluno e seus conhecimentos e aptidões, podendo ser facilmente aplicado a um Modelo do Usuário em geral.

Este padrão considera seis tipos de informações sobre o aluno (usuário) (PAPI, 2002; Dolog & Nejd, 2003):

1. Contato: fundamentalmente relacionado à administração;
2. Relacionamento: referente à relação do usuário com outros usuários do sistema;
3. Segurança: relativo às credenciais de segurança do usuário;
4. Preferências: descreve preferências que podem melhorar a interação humano-computador, indicando tipos de dispositivos e objetos que o aluno pode reconhecer;

5. Desempenho (*performance*): relacionado ao histórico do usuário, trabalho atual ou objetivos futuros; armazena informações sobre a *performance* do aluno pelo material de ensino, ou seja, o que o aluno sabe. É usado para melhorar ou otimizar as experiências de aprendizagem (utilização);
6. *Portfolio*: é uma coleção representativa do trabalho do usuário ou referências a eles. O objetivo é ilustrar ou justificar as habilidades e realizações (experiência prévia) dos usuários.

Esses tipos, exemplificados na Tabela 3, são considerados informações sobre o perfil do usuário (*Profile Information*).

Tipo de Informação	Alguns Exemplos
Contato	Nome, endereço, <i>e-mail</i> , CPF.
Relacionamento	Professor, aluno, colega, vendedor, cliente, paciente.
Segurança	Senhas, direitos de acesso.
Preferências	Dispositivos E/S, estilos de aprendizagem, deficiências físicas.
Desempenho	Notas (graus), registro das interações com o sistema (<i>log books</i>).
<i>Portfolio</i>	Trabalhos realizados.

Tabela 3 – Tipos de informação do PAPI

O padrão PAPI permite não só que as informações sejam estendidas, como também que novos tipos sejam acrescentados.

2.3.3. IMS LIP

Este pacote descreve as características dos alunos necessárias para manipular o histórico, objetivo e realizações relacionados ao aprendizado deles; envolvê-los na experiência de aprendizagem e descobrir oportunidades de aprendizagem para eles. A especificação dá suporte à troca de informações entre sistemas.

O modelo contém dados e metadados sobre estas informações, definindo os campos dos dados e o tipo de informação que eles contêm. Exemplos de dados são: nome, curso completado, objetivo de aprendizagem, preferência por tipo de tecnologia. Metadados podem incluir informações de tempo; identificação e indexação; privacidade e proteção dos dados. Estes metadados estão disponíveis para cada campo do modelo de informação, diretamente ou via herança.

A informação é separada em onze categorias principais, identificadas como as estruturas de dados básicas necessárias para a representação de informações sobre o aluno (IMS, 2001; Dolog & Nejd, 2003):

1. Identificação: informações pessoais – tais como nome, endereço, informações para contato, agentes (quem pode agir em nome do aluno) –, além de dados demográficos (exemplos: local de nascimento, número de documento), de um indivíduo ou organização;
2. Objetivo: de aprendizado, objetivos profissionais e outros. Pode também incluir informações de monitoramento do progresso em atingir os objetivos. Pode ser definido em termos de subobjetivos;
3. QCL (*Qualifications, Certifications and Licenses*): identificação de qualificações, certificações e licenças reconhecidas. Pode conter cópias eletrônicas de documentos;
4. Atividade: qualquer atividade de aprendizagem em qualquer estágio de realização. Pode ser auto-reportada. Inclui educação formal e informal, treinamento, experiência profissional, serviço militar ou civil. Referente às atividades para as quais a informação do aluno é relevante;
5. Transcrito (*Transcript*): registro utilizado para prover um resumo institucional das realizações acadêmicas em um nível arbitrário de detalhe. Em geral, fornecido pelos responsáveis por avaliar o desempenho da pessoa;
6. Interesse: qualquer informação que descreva *hobbies* e atividades recreativas. Pode incluir prêmios formais (descritos no “QCL” associado). Pode conter versões eletrônicas dos produtos gerados pelos interesses;
7. Competência: experiência, conhecimento e habilidades adquiridas nos campos cognitivo, emocional e/ou psicomotor. Pode estar associada com treinamento formal ou informal ou histórico de trabalho (descrito em “atividade”) e prêmios (descritos no “QCL”);
8. Filiação: organizações a que o aluno pertence;
9. Acessibilidade: acesso geral a informações do usuário em termos de linguagem, deficiências, aptidões, preferências de aprendizagem

incluindo preferências cognitivas (exemplo: estilo de aprendizagem), físicas (como preferência por fonte grande) e tecnológicas (tal como preferência por determinada plataforma); Descreve a capacidade do aluno de interagir com o ambiente de aprendizagem;

10. Chave de segurança: armazena a descrição das senhas, chaves de criptografia e chaves de autenticação atribuídas ao aluno para transações com os sistemas e serviços;
11. Relacionamento: o conjunto de relacionamentos entre os outros componentes de dados principais. Pode também ser usada para mapear relacionamentos a serem usados nos sistemas que se comunicam.

Essas categorias foram escolhidas para atender aos requisitos de uma grande variedade de casos de uso e para facilitar o mapeamento entre IMS e outras especificações relevantes.

Como se pode observar, neste padrão a informação é um conjunto de informações sobre o aluno (indivíduos ou grupos) ou sobre produtores do conteúdo de aprendizagem (autores; provedores: departamento de treinamento, universidade, tutor; ou fornecedores: entidade que oferece produtos como ferramentas de software) cujo objetivo é a troca de informações entre sistemas.

2.3.4. CC/PP

Um perfil⁶ CC/PP consiste na descrição das capacidades dos dispositivos e das preferências do usuário, freqüentemente denominada de contexto de envio para dispositivos (*device's delivery context*). Ou seja, é uma tecnologia voltada para a solução de problemas relacionados ao envio de conteúdo para dispositivos diversos que pode ser usada para guiar a adaptação da apresentação do conteúdo em determinado dispositivo. Assim, por exemplo, pode-se especificar que mesmo que um navegador suporte milhões de cores, determinado usuário só consiga distinguir certas cores. Ou ainda que um usuário só use teclado (CC/PP, 2004; Wasp, 2004). A Figura 5 mostra o exemplo de um perfil CC/PP.

⁶ Neste caso, perfil (*profile*) diz respeito ao(s) documento(s) trocado(s) entre dispositivos, com a descrição das capacidades de um dispositivo.

```

[ex:MyProfile]
|
+--ccpp:component-->[ex:TerminalHardware]
|
|           +--rdf:type----> [ex:HardwarePlatform]
|           +--ex:displayWidth--> "320"
|           +--ex:displayHeight--> "200"
|
+--ccpp:component-->[ex:TerminalSoftware]
|
|           +--rdf:type----> [ex:SoftwarePlatform]
|           +--ex:name-----> "EPOC"
|           +--ex:version--> "2.0"
|           +--ex:vendor---> "Symbian"
|
+--ccpp:component-->[ex:TerminalBrowser]
|
|           +--rdf:type----> [ex:BrowserUA]
|           +--ex:name-----> "Mozilla"
|           +--ex:version--> "5.0"
|           +--ex:vendor---> "Symbian"
|           +--ex:htmlVersionsSupported--> [ ]
|
|           -----
|           |
|           +--rdf:type---> [rdf:Bag]
|           +--rdf:_1-----> "3.2"
|           +--rdf:_2-----> "4.0"

```

Figura 5 – Exemplo de um perfil CC/PP (CC/PP, 2004)

A maioria das soluções existentes para o gerenciamento de dispositivos variados é baseada na seleção de conteúdo – que é disponibilizado de diversas formas, escolhidas de acordo com os recursos – ou tem mecanismos que definem comportamentos alternativos. Tais procedimentos são relativamente simples considerando-se os artefatos definidos para *Web*, tais como os cabeçalhos (*headers*) “*accept*” do HTTP e os atributos “*alt=*”, além dos elementos “*object*” e “*link*”, do HTML que provêm mecanismos para a definição de comportamentos alternativos. Outros exemplos são a linguagem multimídia para conteúdo audiovisual, SMIL, que através do elemento “*switch*” pode definir alternativas de escolha como, por exemplo, o tipo de conteúdo de acordo com a largura da banda; e o mecanismo “*Media Queries*” da CSS, que seleciona as folhas de estilo apropriadas.

Ao expressar as capacidades dos dispositivos, o CC/PP provê flexibilidade. Além de simplesmente definir um conjunto fixo de preferências usado para construir o perfil dos dispositivos, este *framework* baseado em RDF também permite a criação de vocabulários completos, tornando a representação das capacidades dos dispositivos, bem como das preferências dos usuários, extensível.

Usando CC/PP os produtores de dispositivos da *Web* e os agentes de usuários⁷ podem facilmente definir perfis precisos para seus produtos. Assim, os servidores *Web* e *proxies* podem usar estes perfis para adaptar o conteúdo que eles estão disponibilizando de acordo com as necessidades do dispositivo.

O RDF é uma escolha natural, uma vez que os perfis de agente do usuário⁸ são metadados com o objetivo primeiro de comunicação entre tais agentes e provedores de dados dos recursos. Assim, um perfil CC/PP contém nomes de atributos e valores associados usados por um servidor para determinar a forma mais apropriada de recurso a enviar para o cliente. Ele é estruturado de modo a permitir a um cliente descrever suas capacidades, com relação a um perfil padrão – acessível a um servidor origem ou a outro remetente dos dados do recurso –, e um conjunto menor de características adicionais ao, ou diferentes do, perfil padrão. O vocabulário CC/PP é constituído, então, pelos nomes de atributo, valores permitidos e significados associados. Os vocabulários serão diferentes para aplicações diversas, caso se deseje representar propriedades específicas das mesmas. Para que elas trabalhem juntas, é necessário um vocabulário comum ou um método de conversão entre os vocabulários.

O CC/PP define um perfil para o formato dos dados do cliente e um *framework* para incorporar características específicas de aplicações e do ambiente operacional, tendo sido projetado para ser usado como parte de um *framework* mais amplo da aplicação. Sozinho, ele não é capaz de resolver o problema de autoria para qualquer dispositivo. Entretanto, as transformações disparadas pelo CC/PP são promissoras. Provavelmente, linguagens como SMIL e CSS serão utilizadas em conjunto com os perfis e preferências do CC/PP.

2.3.5. GUMO

A ontologia GUMO baseia-se na proposta do UserML (Heckmann & Krüger, 2003 apud Heckmann et al., 2005), cuja idéia central é a divisão das Dimensões do UM (*User Model Dimension*) em três partes: auxiliar, predicado e

⁷ *User agent*: um programa como, por exemplo, um navegador (*browser*) rodando no dispositivo de acordo com os interesses do usuário. Usuários podem usar agentes diferentes em momentos diferentes.

⁸ *User agent profile*: informação sobre as capacidades e preferências relativas às capacidades do dispositivo, ambiente operacional e de rede e preferências pessoais do usuário para receber conteúdo e/ou recursos.

intervalo (respectivamente, *auxiliary*, *predicate* e *range*), conforme especificado abaixo (Heckmann et al., 2005):

sujeito { Dimensões_do_UM } objeto ↓ sujeito { auxiliar, predicado, intervalo } objeto
--

Assim, para representar o interesse por futebol, por exemplo, pode-se utilizar o auxiliar “TemInteresse” (*auxiliary* = *hasInterest*), o predicado “futebol” (*predicate* = *football*) e o intervalo “baixo, médio, alto” (*range* = *low-medium-high*). Uma vez que praticamente tudo pode ser predicado para os auxiliares “TemInteresse” ou “TemConhecimento” (*hasKnowledge*), é necessário identificar as dimensões básicas do UM por um lado, deixando o conhecimento mais geral do mundo aberto para outras ontologias já existentes, como SUMO (Pease, 2002 apud Heckmann et al., 2005) ou UbisWorld (<http://www.ubisworld.org>), por outro lado. Desta forma, obtém-se uma arquitetura modular, característica básica do GUMO.

GUMO procura abranger todas as classes, predicados e instâncias referentes às condições e modelos do usuário, aos dispositivos dos sistemas e aos ambientes. A Figura 6 apresenta a ontologia GUMO com as informações de contato expandidas. A ontologia completa pode ser consultada em www.gumo.org.

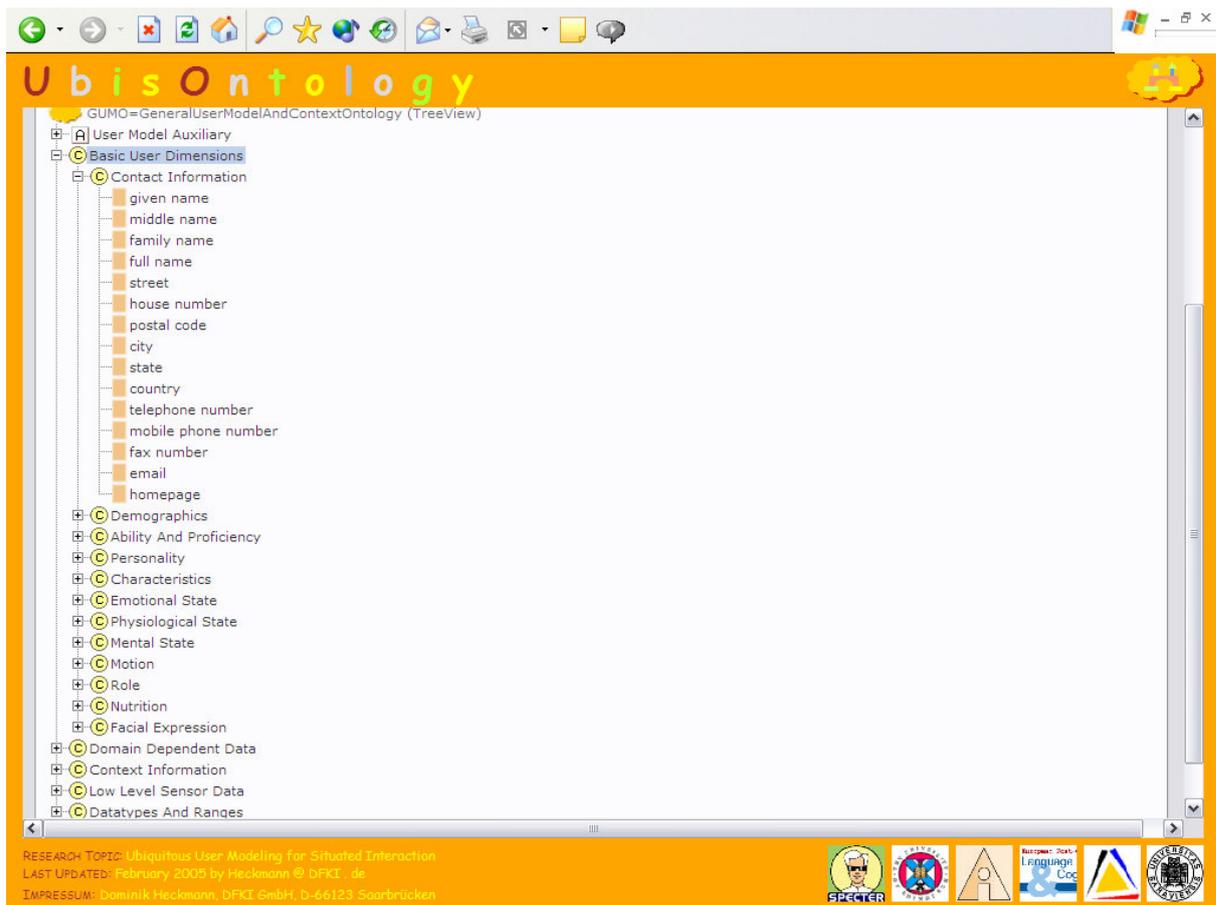


Figura 6 – Ontologia GUMO

As classes relativas ao usuário que estão no topo da ontologia são:

- Dimensões Básicas do Usuário;
- Dados Dependentes do Domínio;
- Informações de Contexto.

São modeladas, também, as dimensões auxiliares (*hasProperty*, *hasInterest*, *hasBelief*, *hasKnowledge*, *hasPlan*, *hasGoal*, *hasPreference*, *hasRegularity*, *hasDone*, *hasLocation*), além dos dados obtidos por sensores de baixo nível (*Low Level Sensor Data*) e dos tipos de dados e intervalos (*Datatypes and Ranges*, como *default*, texto, número, etc. e conjuntos de elementos: “yes no”, “low medium high”, “from -3 to 3”, etc.).

A ontologia GUMO pode ser alterada através do *site* www.u2m.org, bastando que o usuário se registre informando nome, senha e, opcionalmente, o endereço eletrônico. Assim, os exemplos a seguir refletem a ontologia em seu estado atual, com contribuições de usuários.

As Dimensões Básicas do Usuário são consideradas como sendo os predicados adequados ao auxiliar “TemPropriedade” (*hasProperty*). Algumas

dessas dimensões são: Informações para Contato (nome, endereço, telefone, etc.); Demográficas (gênero, faixa etária, idade, naturalidade, nível educacional, salário, etc.); Capacidade e Proficiência (capacidade de: enxergar, cheirar, escutar, andar, subir escada, nadar, etc.; capacidade de leitura, escrita, digitação); Personalidade (extrovertido, introvertido, sensível, intuitivo, otimista, etc.); Características (falante, dominante, tímido, prestativo, eficiente, frívolo, ansioso, calmo, imaginativo, cooperador, etc.).

Os Dados Dependentes do Domínio incluem: (a) Categorias de Interesse, entre as quais, Filmes (de ação, de televisão, outros, etc.); Música (clássica, popular, concertos, etc.); Meio-Ambiente (poluição do ar, aquecimento global, políticas públicas, energia nuclear, etc.); (b) Conhecimento; (c) Crença; (d) Preferências como Preferências de Privacidade; de Interface (*layout, modality*) e outras; (e) Regularidade; (f) Planos e (g) Objetivos.

As classes consideradas como Informações de Contexto são: (a) Localização (espacial, virtual, coordenadas); (b) Ambiente Físico (nível de ruído, temperatura, luminosidade); (c) Ambiente Social como Família e Colegas (classes) e amigos e (d) Informações sobre Produtos (preço, fornecedor, tamanho da tela, capacidade da bateria).

A linguagem OWL é utilizada para representar os termos do UM e suas inter-relações.

2.3.6. Similaridades entre os Padrões

À exceção do GUMO, as ontologias apresentadas foram desenvolvidas com objetivos específicos. O PAPI e o IMS LIP são voltados para a área educacional, embora o primeiro seja mais facilmente portátil para um sistema de propósito geral. O CC/PP é o mais restrito, voltado essencialmente para adaptar o ambiente de utilização. A Tabela 4 sumariza as principais categorias de informações sobre o usuário modeladas por cada um destes padrões. A compreensão dessas ontologias ajuda a identificar o tipo de informação desejável para um Modelo de Usuário.

PAPI	IMS LIP	CC / PP	GUMO
Contato	Identificação (biográficas e demográficas)		Contato
Relacionamento	Relacionamento (entre categorias)		Papel
Segurança	Chave de Segurança		(*)
Preferências	Acessibilidade	Preferências	Preferências (Dependente do Domínio)
Desempenho	QCL Competência		(*)
<i>Portfolio</i>	Atividade Transcrito		(*)
	Objetivo		Objetivo (Dependente do domínio)
	Interesse		Interesse (Dependente do Domínio)
	Filiação		(*)
		Capacidade do dispositivo	Informações do Produto (Contexto)
	Demográficas (dentro de identificação)		Demográficas
			Personalidade
			...

(*) não existem atualmente, mas classes ou instâncias podem ser criadas.

Tabela 4 – Principais categorias de informações sobre usuário modeladas por vários padrões