

3 Trabalhos Relacionados

3.1. Introdução

Neste capítulo são analisados algumas aplicações e sistemas de hipermídia adaptativa. Para tanto, é utilizado o Metamodelo de Referência para Sistemas Hipermídia Adaptativos apresentado em Assis (2004). Os modelos que o compõem representam a essência de AHSs no que diz respeito às principais questões de adaptação identificadas. Cada aplicação é instanciada em relação ao metamodelo. Desta forma, é possível analisar, além dos pontos em comum, as particularidades de cada aplicação tanto na interpretação do metamodelo, quanto nas características extras que elas apresentam.

3.2. Metamodelo de referência

A representação do metamodelo de referência (Figura 7) utiliza a notação de diagrama de classe da UML⁹.

⁹A notação UML foi estendida com uma relação entre classe e pacote para representar a existência de relacionamento entre a referida classe com qualquer classe integrante do pacote relacionado.

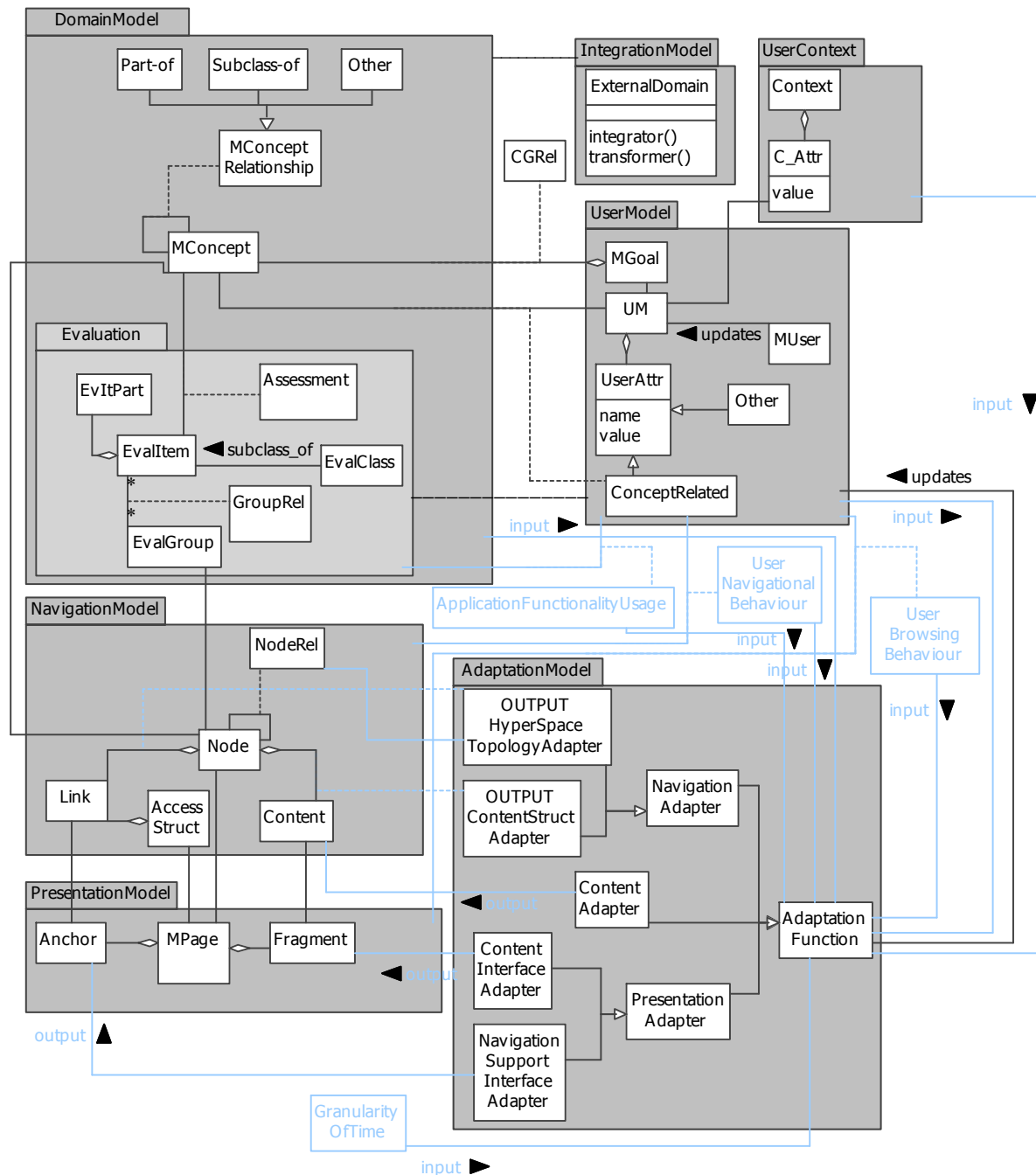


Figura 7 – Metamodelo de referência para AHSs

O metamodelo baseia-se na proposta do AHAM (*Adaptive Hypermedia Application Model*) (De Bra et al., 1999), segundo a qual uma adaptação efetiva em uma aplicação depende de três fatores (Wu et al., 2001):

- A aplicação precisa ser baseada em um Modelo de Domínio que descreva, usando conceitos e relações entre eles, como o conteúdo da aplicação é estruturado;
- É necessário um Modelo de Usuário que represente, essencialmente, o conhecimento, objetivos e preferências do usuário;

- O sistema deve ser capaz de realizar a adaptação baseado nos Modelos do Domínio e do Usuário, utilizando um Modelo de Adaptação.

Além disto, o ideal é que haja uma separação clara entre esses três modelos. A identificação de cada um deles permite a sua análise individual bem como dos seus inter-relacionamentos.

De acordo com os princípios do OOHDm que, como já visto, preconiza a separação entre os aspectos conceituais, navegacionais e de apresentação, foram acrescentados os seguintes modelos:

- Modelo de Navegação para permitir identificar as diferentes visões navegacionais em relação ao domínio, englobando os nós, seus conteúdos e elos e as estruturas de acesso às páginas;
- Modelo de Apresentação para separar da interface as características mais concretas de implementação.

Outros modelos foram identificados, a saber:

- Modelo de Integração que contempla os sistemas abertos, cujo domínio se estende para além do próprio sistema, integrando domínios externos ao Modelo do Domínio;
- Modelo do Contexto (Ambiente) de Utilização, referente às condições de utilização do sistema tais como localização, dispositivo e velocidade de acesso.

Cada um dos modelos identificados está encapsulado em um pacote UML (caixa cinza na Figura 7) e é descrito nas próximas seções. A funcionalidade da adaptação aparece com uma tonalidade diferente e pode ser melhor compreendida na seção 3.2.5, onde é detalhada.

Na instanciação dos AHSs, os nomes das classes são precedidos pela metaclassa correspondente. Uma vez que a modelagem foi construída sob a ótica do metamodelo de referência, ela pode ser diferente da descrição original feita pelos respectivos autores sem, entretanto, alterar a essência dos sistemas¹⁰. Exemplos disso são os Modelos de Navegação e Apresentação que não são definidos explicitamente na descrição de nenhum dos sistemas instanciados, mas

¹⁰ Os metamodelos descritos foram apresentados e validados pelos seus respectivos proponentes, através de interações pessoais e via comunicação eletrônica..

são “criados” nas instanciações de acordo com o metamodelo de referência. O Modelo do Contexto (Ambiente)¹¹ não aparece porque nenhum dos sistemas abordados considera a situação de uso para a adaptação. A questão do “quando” não é abordada explicitamente pelos sistemas, não sendo, por isto, modelada.

3.2.1. Modelo do Domínio

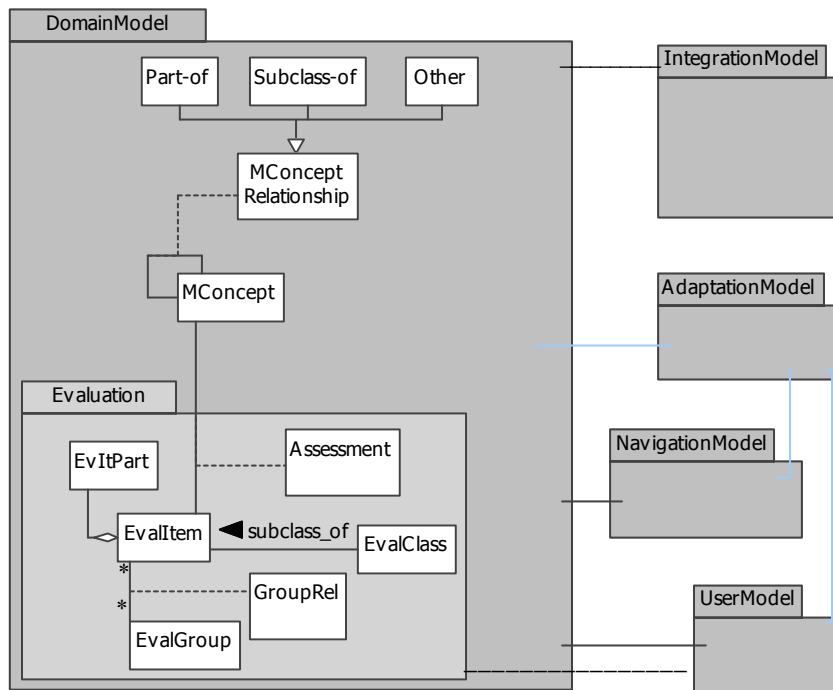


Figura 8 – Modelo do Domínio

O DM (Figura 8) consiste, basicamente, de um conjunto de conceitos (*MConcept*¹²), relacionados entre si (*MConceptRelationship*). As relações são, tipicamente¹³: subparte (*part-of*), subclasse (*subclass-of*) e outras (*other*).

Em geral, existe um esquema de avaliação (pacote *Evaluation*) onde alguns itens (*EvalItem*) são usados para avaliar o conhecimento do usuário sobre certos conceitos, de acordo com um determinado peso (*Assessment*). Estes itens podem ser compostos (por *EvItPart*), podem estar agrupados – em *EvalGroup*, com o critério definido em *GroupRel* – e podem ser de diversos tipos (*EvalClass*). Tal esquema pode ser mais bem entendido pela instanciação do NetCoach (seção 3.6).

¹¹ A denominação *UserContext*, como aparece no modelo é obsoleta, uma vez que o contexto é o ambiente de uso, independentemente do usuário.

¹² A letra M é usada para indicar conceitos do metamodelo, distinguindo-os dos conceitos dos sistemas, conforme será visto nas instanciações. Analogamente para *MConceptRelationship*, *MUser*, *MGoal* e *MPage*.

¹³ Relações de elo entre conceitos, consideradas por alguns autores, fazem parte do nosso

Por ser a avaliação feita em relação aos conceitos, ela foi incluída no DM. Entretanto, esta é uma questão em aberto, uma vez que a avaliação pode ser considerada de uma maneira mais ampla, como um mecanismo de observação das relações entre o DM e o UM (seção 3.2.2).

3.2.2.

Modelo do Usuário e do Contexto (Ambiente) de Utilização

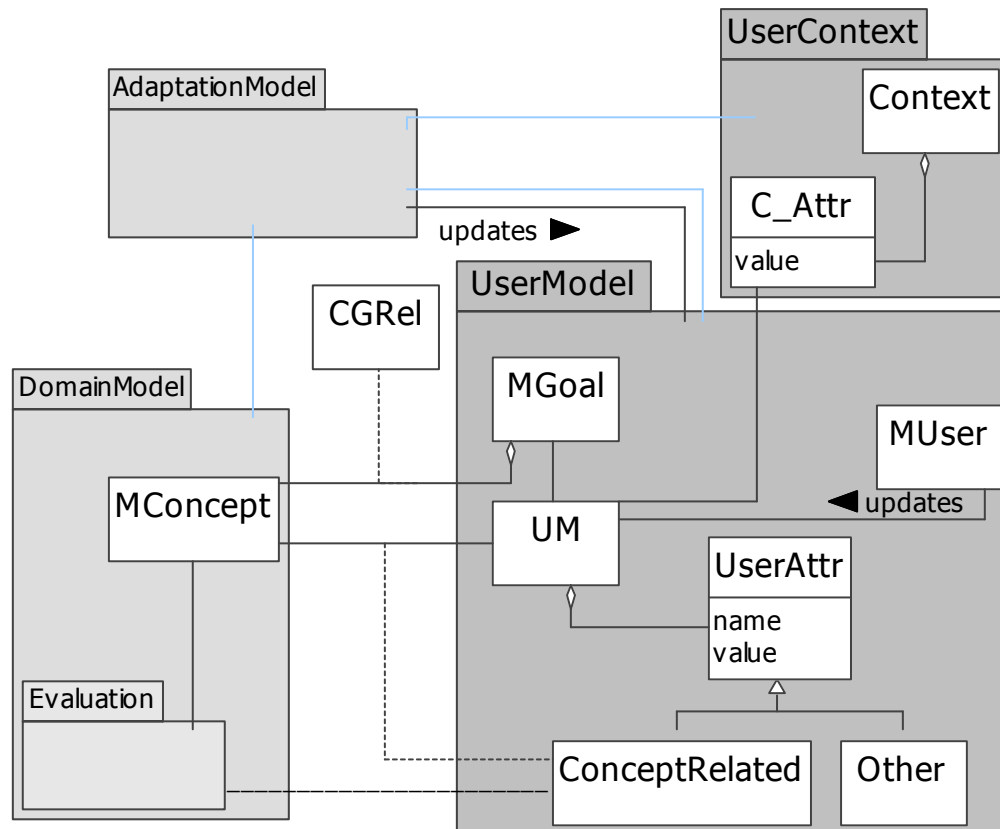


Figura 9 – Modelo do Usuário e Contexto (Ambiente) de Utilização

A Figura 9 mostra o UM e como o Contexto (Ambiente) de Utilização é modelado.

O UM, em geral, possui atributos relacionados aos conceitos do domínio (*ConceptRelated*) e inclui um conjunto de objetivos (*MGoal*) a serem alcançados pelo usuário. Em especial nos sistemas educacionais, aspectos modelados importantes são o nível de conhecimento (incluído no perfil do usuário) que o usuário tem sobre um conceito e a definição de quais os seus objetivos, no caso, quais conceitos ele deseja conhecer. Em um domínio de comércio eletrônico, por exemplo, os atributos poderiam indicar quais produtos o usuário já comprou e os

objetivos poderiam representar se o usuário deseja comprar ou presentear um produto.

Pode existir uma relação (*CGRel*) entre objetivos e conceitos representando, por exemplo, no caso de objetivos de aprendizagem, em que ordem os conceitos devem ser aprendidos ou se algum conceito é consequência de outro.

Alguns sistemas permitem que o próprio usuário (*MUser*) atualize (*updates*) diretamente o modelo sobre ele armazenado pelo sistema. Um mecanismo de observação das relações entre o DM e o UM também pode gerar uma atualização deste último. Tal mecanismo é representado por avaliações (*Evaluation*, seção 3.2.1), que tanto podem ser o resultado de testes existentes em alguns sistemas, como podem ser generalizadas para outros tipos de uso, como um *feedback* de relevância dado por usuários em outros domínios (não-educacionais). O UM também pode ser atualizado como resultado da Função de Adaptação (seção 3.2.5).

Além dos relacionados aos conceitos, outros atributos do usuário (*Other*) – independentes do domínio – podem ser levados em consideração para a adaptação, tais como preferências do usuário.

A situação de utilização do sistema (*Context*) é modelada por atributos (*C_Attr*) que também são relacionados ao usuário e independentes do domínio. A diferença é que estes atributos de contexto não estão relacionados à individualidade do usuário. Por exemplo: no caso do tipo de acesso (discado ou banda larga), o foco é a velocidade de acesso do usuário, independentemente deste usuário ser o Daniel ou a Patricia.

3.2.3. Modelos de Navegação e de Apresentação

Os Modelos de Navegação e de Apresentação aparecem representados na Figura 10.

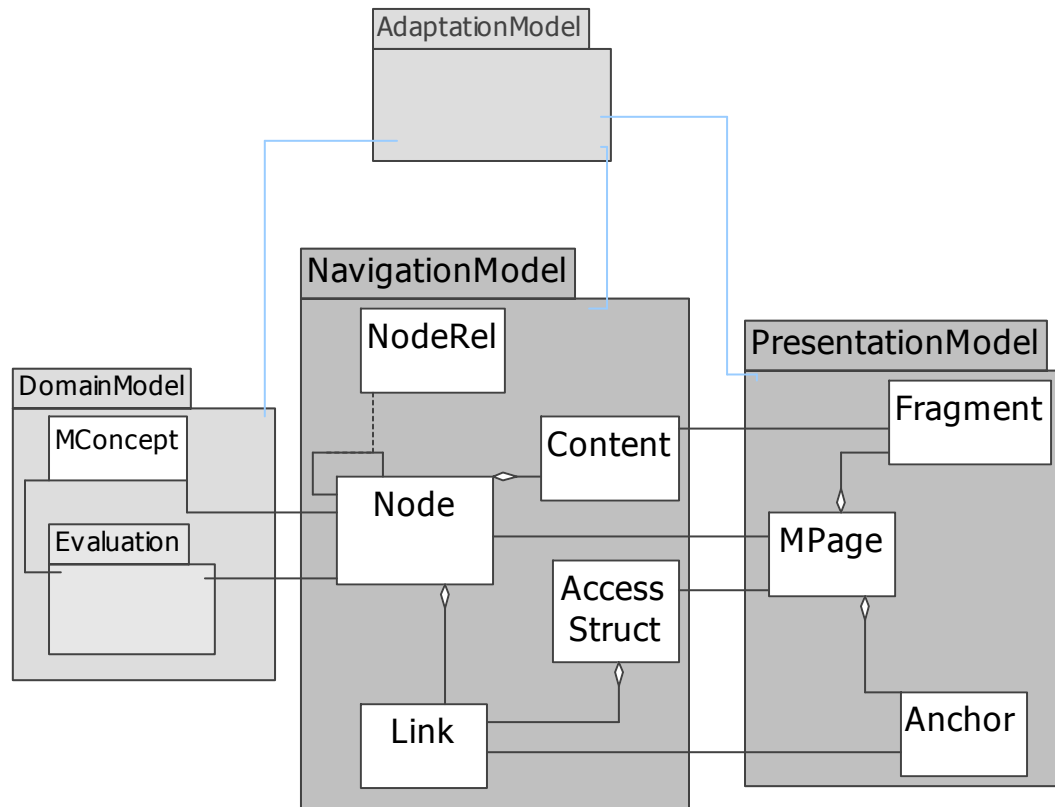


Figura 10 – Modelos de Navegação e de Apresentação

Os nós (*Node*) representam visões navegacionais dos conceitos (*MConcept*) e podem estar relacionados entre si (*NodeRel*) através de relações arbitrárias ou padronizadas como no caso de “próximo”, “anterior”, “primeiro”, “último”.

Embora em desacordo com a proposta do SHDM, que diferencia nós de navegação das páginas físicas de uma aplicação, nesta representação, nós correspondem a páginas (*MPage*) do Modelo de Apresentação, de modo a permitir a instanciação das aplicações analisadas. O modelo foi construído desta forma pois reflete mais fielmente como os sistemas atuais são construídos. Como será visto posteriormente, esta correspondência não é mantida na proposta do metamodelo semântico.

O conteúdo (*Content*) é mapeado para os fragmentos (*Fragment*) e os elos (*Link*) são mapeados para as âncoras (*Anchor*). Estruturas de Acesso (*AccessStruct*) contêm elos (*Link*) e funcionam como índices.

3.2.4. Modelo de Integração

Este modelo (Figura 11), ainda incipiente, utiliza funções de integração responsáveis por buscar dados na *Web* (*integrator*) e convertê-los (*transformer*) para a representação utilizada

pelo DM (Vdovjak et al., 2003).

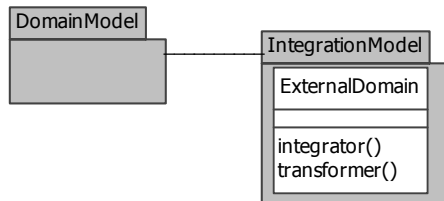


Figura 11 – Modelo de Integração

3.2.5. Modelo de Adaptação

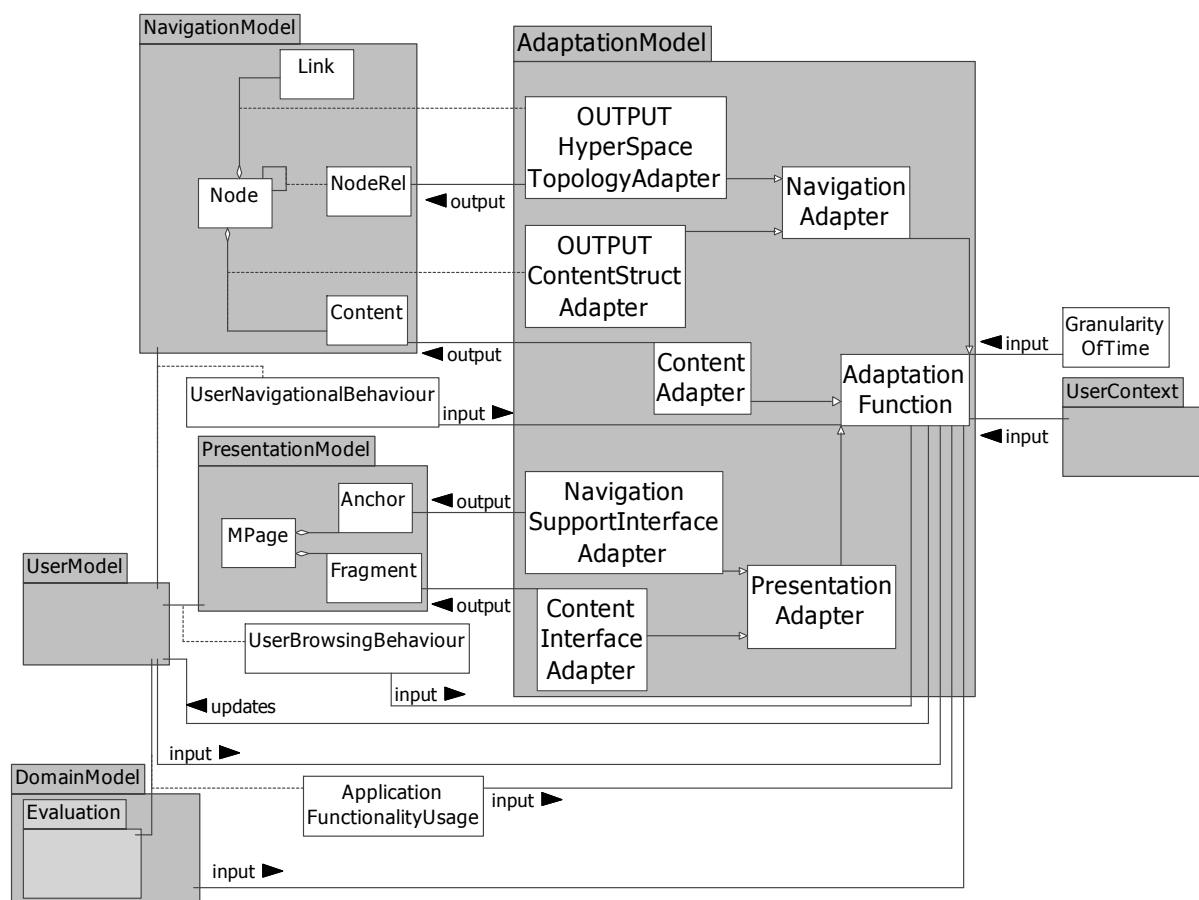


Figura 12 – Funcionalidade da Aplicação

Este modelo (Figura 12) responde às principais questões de adaptação, representando a funcionalidade da aplicação e propiciando uma maneira de expressar os diversos tipos de adaptação. A classe “Função de Adaptação” (*AdaptationFunction*) é usada para encapsular o mecanismo de adaptação usado no sistema (por exemplo, regras no caso do AHA!), representando “como” a adaptação é feita (seção 2.1.3.3). Esta função produz como saída (*output*) “o quê” o sistema adapta (seção 2.1.3.1). Observa-se, desta forma, que a adaptação da

topologia do hiperespaço (*HyperspaceTopologyAdapter*) é um tipo de adaptação de navegação (*NavigationAdapter*) que ocorre na relação entre nós (*Node*) e elos (*Link*) – como no caso da inclusão condicional de elos – ou na relação entre nós (*NodeRel*), por exemplo, na orientação direta. A inclusão condicional de fragmentos pode ser representada pela subfunção de adaptação da estrutura do conteúdo (*ContentStructAdapter*) que ocorre na relação entre nós (*Node*) e conteúdo (*Content*). A subfunção de adaptação do conteúdo (*ContentAdapter*) atua diretamente no conteúdo dos nós. Já a adaptação da apresentação (*PresentationAdapter*) tem reflexos no Modelo de Apresentação, mais especificamente nas âncoras (*Anchor*) – adaptação da interface do suporte à navegação (*NavigationSupportInterfaceAdapter*) – e nos fragmentos (*Fragment*) – adaptação da interface do conteúdo (*ContentInterfaceAdapter*).

As entradas (*inputs*) da função representam “em função de quê” a adaptação ocorre (seção 2.1.3.2: usuário, estruturação do domínio, granularidade do tempo, etc.). A relação da granularidade de tempo com outros modelos, tais como domínio e contexto (exemplo: adaptar quando muda o contexto) é uma questão a ser avaliada, uma vez que não foi observada em nenhum dos sistemas estudados.

3.3. AHAM

O AHAM (*Adaptive Hypermedia Application Model*) (De Bra et al., 1999) é o modelo por trás de diversos sistemas hipermídia adaptativos e defende a separação entre os Modelos do Domínio, do Usuário e de Adaptação. Segundo Wu (2002), o Modelo do Domínio é formado pelos conceitos atômicos, páginas, conceitos compostos e as relações entre os conceitos. No nível mais baixo, estão os fragmentos (conceitos atômicos). Em um nível intermediário, estão as páginas, que são unidades de informação apresentadas ao usuário a cada interação. Estas compõem os conceitos abstratos que, por sua vez, podem compor conceitos abstratos de mais alto nível. Páginas são compostas por fragmentos, possibilitando a representação abstrata de diversas técnicas de adaptação, tais como, inclusão / remoção de fragmentos; alteração de fragmentos; *stretch text* e ordenação de fragmentos. Fragmentos podem ser condicionalmente incluídos, mas não podem ser alterados pelos AHSs.

Nossa proposta é que a instanciação, apresentada na Figura 13, fique de

acordo com o metamodelo de referência que propõe a separação das visões conceitual, navegacional e de apresentação. Assim, os conceitos (compostos) – que representam itens de informação abstratos (Wu et al., 2001) – e as relações entre eles são instanciados no Modelo do Domínio (seção 3.3.1). Já relações de elo, conceitos atômicos e páginas correspondem a classes dos Modelos de Navegação e de Apresentação (seção 3.3.3). Interações com o sistema são descritas através de eventos, modelados como atualizações nos atributos que disparam regras (Wu & De Kort, 2002) (seção 3.3.5).

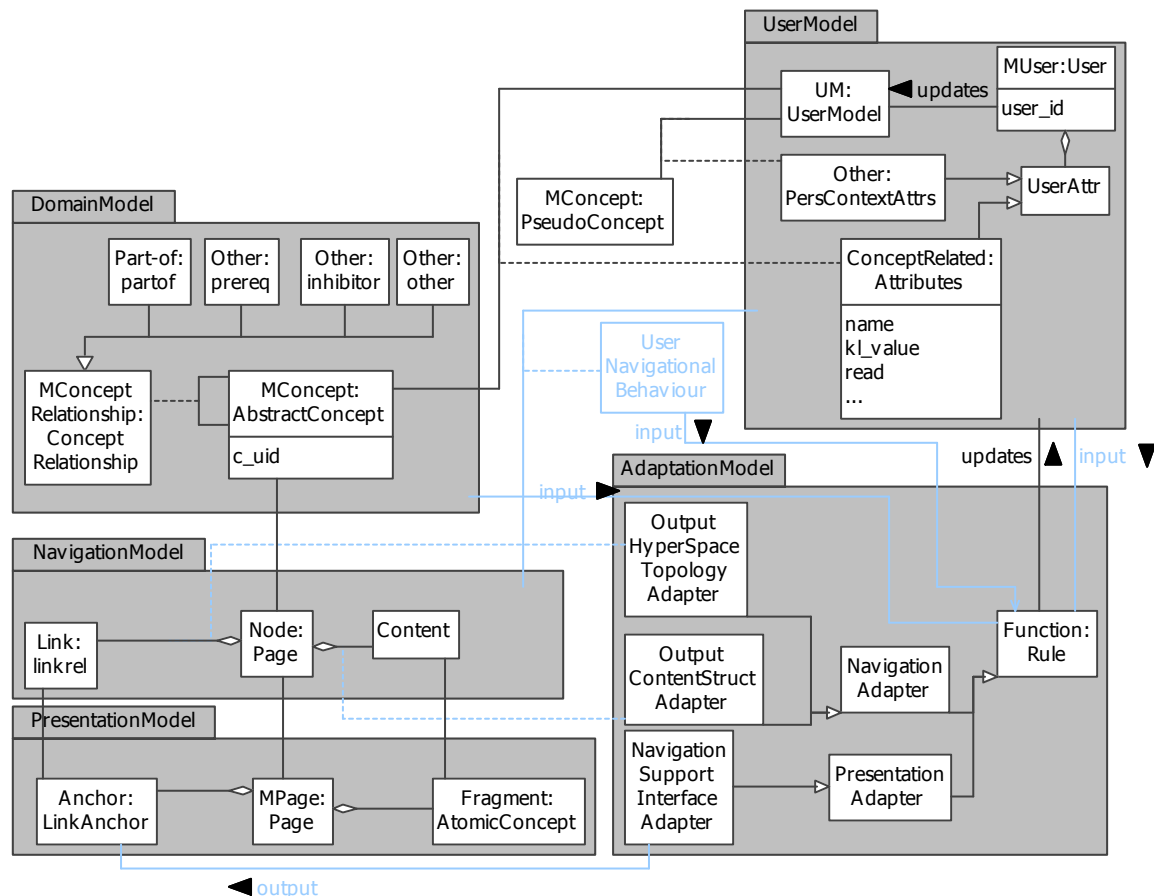


Figura 13 – Instanciamento do metamodelo do AHAM

3.3.1. Modelo do Domínio

O domínio é composto por conceitos abstratos (Wu et al., 2001) e as relações entre eles. A relação “subparte” (*part-of*) pode ser usada para representar a hierarquia entre os conceitos. Outros tipos de relações (abstratas) relevantes para a adaptação – como, por exemplo, pré-requisito (*prereq*) e inibidor (*inhibitor*) – são considerados. Teoricamente, relações podem ser definidas arbitrariamente (*Other: other*).

3.3.2. Modelo do Usuário

O Modelo do Usuário é uma sobreposição (*overlay*) do Modelo do Domínio: para cada conceito são armazenados valores de atributo sobre o conceito. Estes atributos (*Attributes*) representam as relações entre o usuário e os conceitos do DM como, por exemplo, conhecimento do usuário sobre o conceito (*kl_value*) e interesse do usuário pelo conceito. Outro atributo usual é “lido” (*read*), indicando se o usuário leu alguma coisa sobre o conceito. Através destes atributos, o UM rastreia o quanto o usuário sabe sobre os conceitos do domínio. Além destes, o UM pode armazenar conceitos independentes do domínio como, por exemplo, características pessoais e de contexto (*PersContextAttrs*), tais como: “experiência” (*background*); propriedades do dispositivo de acesso; idade. Embora semanticamente bem diferente, este tipo de informação pode ser armazenado e implementado da mesma forma que os conceitos do domínio, pela utilização de “pseudoconceitos” (*PseudoConcept*).

O AHAM assume que o UM é atualizado em função de eventos. Assim, testes, importação de informações sobre o usuário (de sistemas externos) e indicação explícita das preferências do usuário (através de formulários) podem ser eventos utilizados para atualizar o UM (Wu, 2002), sendo modelados pelo esquema geral de adaptação (seção 3.3.5). A possibilidade de o usuário alterar seu próprio modelo é mostrada pela relação “*updates*” entre *User* e *UserModel*.

3.3.3. Modelos de Navegação e de Apresentação

O AHAM não separa a navegação e a apresentação. Seguindo o metamodelo de referência, conceitos atômicos, páginas e âncoras dos elos (*LinkAnchor*) são instâncias, respectivamente, das classes *Fragment*, *MPage* e *Anchor* do Modelo de Apresentação. Por outro lado, de acordo com Wu (2002), conteúdo é um atributo dos fragmentos e a informação é acessada como páginas. Assim, no Modelo de Navegação, página é também instância de *Node* e a relação de elo entre conceitos é mapeada para a classe *Link*.

3.3.4. Modelo de Integração

Uma vez que eventos são utilizados para descrever interações com o

sistema, pode-se considerar que a integração com domínios externos poderia ser implementada como um evento, ou seja, como atualizações em atributos que disparam regras.

3.3.5. Modelo de Adaptação

Para descrever o comportamento de um Sistema Hiperídia Adaptativo, o AHAM exemplifica a Função de Adaptação como sendo implementada por regras, executadas por um mecanismo de adaptação (*Adaptation Engine*), responsável pelos aspectos dependentes da implementação. A descrição detalhada de tal mecanismo foge ao escopo deste trabalho. De uma maneira geral, atualizações nos atributos disparam as (condições das) regras. As (ações das) regras podem atualizar atributos. Assim, regras podem disparar outras regras. A partir de informações do DM, do UM e da navegação do usuário, as regras geram a especificação da apresentação e atualizam o UM. Tipos de adaptação exemplificados são: a escolha de quais fragmentos e quais elos serão incluídos na página e em que ordem (estrutura do conteúdo e topologia do hiperespaço) e especificações das âncoras (interface de suporte à navegação). O autor pode definir se e quais conceitos do UM podem ser atualizados pelo próprio usuário.

3.4. AHA!

AHA! (*Adaptive Hypermedia Architecture*) (De Bra & Ruiters, 2001; De Bra & Stash, 2002; De Bra et al., 2002a, 2002b) é uma ferramenta de propósito geral, desenvolvida para acrescentar adaptação a um *website*. A Figura 14 apresenta a instanciação do seu metamodelo, baseado na versão 2.0. As maiores diferenças em relação à versão 1.0 estão na utilização de um Modelo de Usuário mais bem estruturado e de regras de requisito e geração mais avançadas. Além disto, aspectos independentes do domínio – informados quando o usuário se registra – podem ser utilizados para a adaptação, uma vez que são representados por um tipo especial de conceito (*PersonalConcept*) e seus atributos.

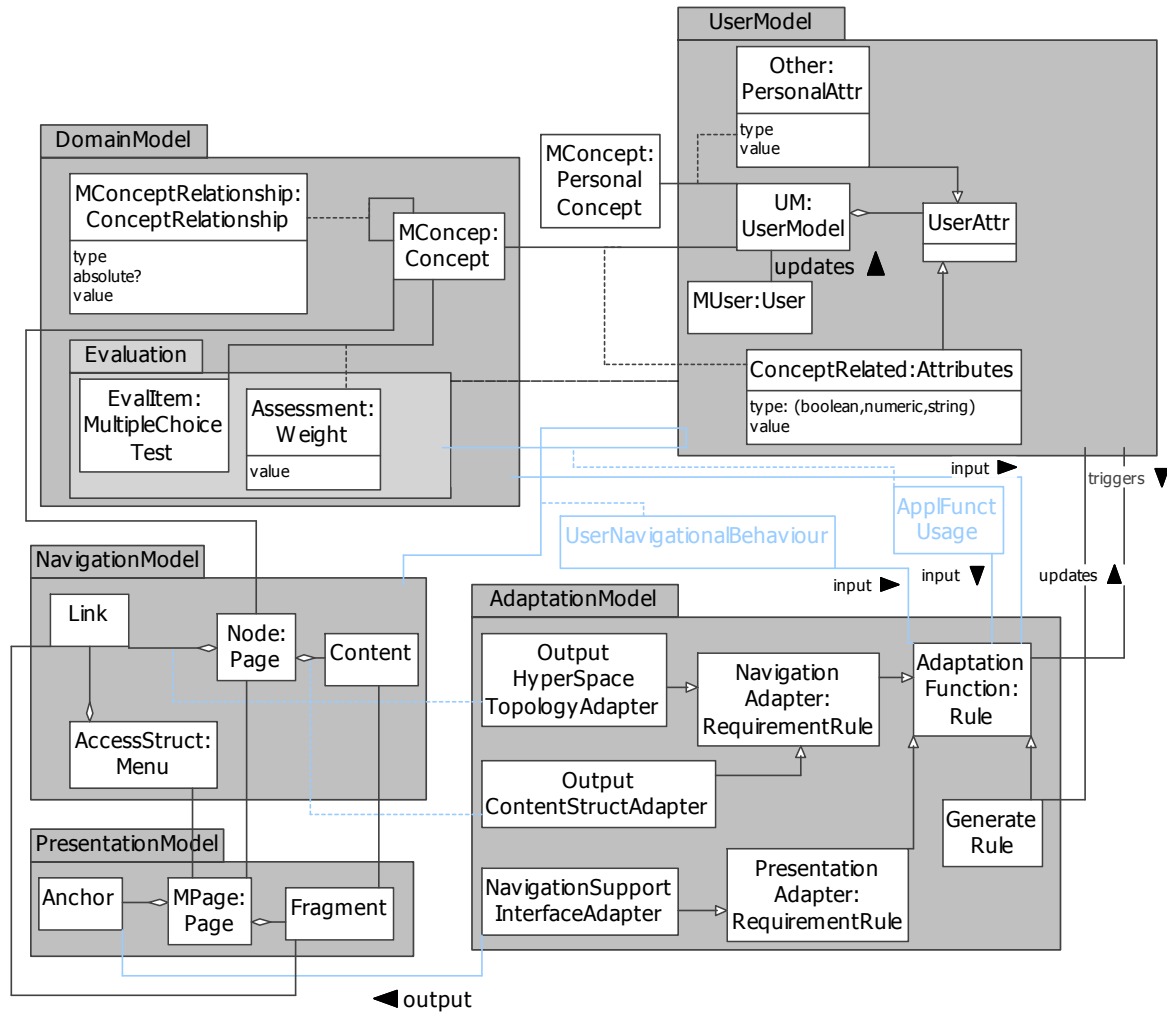


Figura 14 – Instanciação do metamodelo do AHA!

De acordo com De Bra et al. (2002b), o AHA! se desvia do AHAM por não separar o Modelo do Domínio do Modelo de Adaptação. Os conceitos são definidos junto com (regras de) requisitos (*requirement rules*) – que determinam sob quais condições o usuário está “pronto” para acessar o conceito – e com regras de geração (*generate rules*) que especificam como o comportamento de navegação do usuário é utilizado para atualizar o Modelo do Usuário. As regras são do tipo condição-ação e são disparadas por atualizações nos atributos dos conceitos. O acesso às páginas também dispara regras, por ser considerado como uma atualização de um (pseudo) atributo – “acesso” – associado às páginas. Conforme será visto a seguir, a separação entre os Modelos do Domínio e de Adaptação foi mantida na instanciação e as relações entre conceitos e regras e entre páginas e atributos ficam implícitas, podendo ser deduzidas a partir dos relacionamentos existentes no modelo.

Além do acesso às páginas, outros eventos podem disparar as regras.

Considera-se que estes eventos são implementados através de módulos adicionados ao AHA!, tornando o mesmo extensível. A versão atual do AHA! já inclui um módulo que implementa testes de múltipla escolha (seção 3.4.1) e outro para permitir a atualização do UM baseada em formulários (seção 3.4.2).

O AHA! usa cores diferentes para as âncoras, de acordo com o tipo de elo. Por *default*, o AHA! utiliza também a técnica de ocultação de elos (usando a cor preta para elos inoportunos). Entretanto, como o esquema de cores fica armazenado no Modelo do Usuário e pode ser alterado pelo usuário¹⁴, o AHA! pode ser configurado para usar apenas a técnica mais geral de anotação de elos (com a utilização de cores diferentes e visíveis para todos os tipos de elo).

3.4.1. Modelo do Domínio

A relação (*ConceptRelationship*) entre os conceitos (*Concept*) pode ser de diversos tipos (*type*) e pode ter um valor (*value*) associado. Este valor pode ser um percentual ou um valor absoluto e é utilizado para atualizar o valor do conceito relacionado no UM (seção 3.4.2). Na maioria das aplicações, o DM é uma hierarquia de conceitos: alguns conceitos representam páginas enquanto outros correspondem a conceitos abstratos de mais alto nível. As relações entre os conceitos dão uma idéia da ordem em que os mesmos devem ser apresentados.

O autor pode elaborar testes de múltipla escolha (*MultipleChoiceTest*), cujos resultados são utilizados para atualizar os valores dos conceitos associados, de acordo com o peso (*Weight*) atribuído.

O AHA! representa os atributos independentes do domínio da mesma forma que os dependentes. Assim, a representação foi feita de maneira análoga – tratando-os como conceitos, mas fora do DM (instanciação *MConcept: PersonalConcept*).

3.4.2. Modelo do Usuário

Os conceitos formam a base do UM. Para cada conceito, o autor define um conjunto de atributos que relacionam o usuário ao conceito. Estes podem ou não ser dependentes do domínio (*ConceptRelated: Attributes* ou *Other: PersonalAttr*).

¹⁴ A representação do esquema de cores precisa ser modelada como uma extensão ao metamodelo de referência.

Um formulário de registro simples é utilizado para identificar cada usuário individualmente, embora o *login* anônimo também seja permitido. O UM é inicializado a partir das informações obtidas e de valores *default* definidos pelo autor e persiste a cada sessão. No caso de usuários anônimos, o UM refletirá o uso feito pelo último acesso no mesmo navegador e computador. O autor também pode utilizar formulários para permitir que o usuário atualize diretamente o valor de (alguns) conceitos do UM (relação “*User updates UserModel*”). A implementação dos formulários não é modelada.

3.4.3. Modelos de Navegação e de Apresentação

Uma vez que o AHA! não define um Modelo de Apresentação nem um de Navegação, estes foram representados de acordo com o metamodelo de referência, sendo que a página (*Page*) do AHA! foi instanciada para ambos, de modo a possibilitar a representação dos tipos de adaptação que ocorrem. No AHA!, o próprio autor pode implementar estruturas de acesso (*Menu*). Uma vez que o AHA! considera elos (*Link*) como fragmentos (*Fragment*) (ver seção 3.4.5), uma relação entre eles precisou ser criada.

3.4.4. Modelo de Integração

O AHA! não faz uma referência explícita a um Domínio Externo, talvez porque o acesso a URLs externas seja permitido, mas sem adaptação. Por outro lado, um módulo para implementar a integração de domínios poderia ser adicionado.

3.4.5. Modelo de Adaptação

A Função de Adaptação (*AdaptationFunction*) é implementada por regras, que podem ser de geração (*GenerateRule*, atualizam o UM) ou de requisito (*RequirementRule*: adaptação da navegação e da apresentação).

Para o AHA!, as regras de requisitos provêm dois tipos de adaptação: adaptação da apresentação, feita pela inclusão condicional de fragmentos, e adaptação da navegação, pela anotação dos elos. Entretanto, o metamodelo de referência classifica a inclusão condicional de fragmentos como uma adaptação da estrutura do conteúdo (*ContentStruct*) e a anotação (das âncoras) dos elos como

adaptação da apresentação ou, mais especificamente, adaptação da interface de suporte à navegação (*Navigation Support Interface*). Além disso, como elos (*links*) são considerados fragmentos pelo AHA!, o mecanismo de inclusão condicional de fragmentos possibilita, também, a inclusão ou não de elos, implementando a adaptação da topologia do hiperespaço (*Hyperspace Topology*).

As regras implementam a adaptação e atualizam o UM a partir de informações sobre o histórico de navegação do usuário (acessos às páginas), perfil do usuário (deduzido do histórico de navegação), resultado dos testes (*ApplicationFunctionalityUsage*) e DM.

3.5. Interbook

O InterBook (Eklund et al., 1997; Brusilovsky et al., 1998), cujo metamodelo aparece instanciado na Figura 15, é um sistema de autoria para o desenvolvimento de livros-texto eletrônicos (*Electronic Textbooks, ETs*).

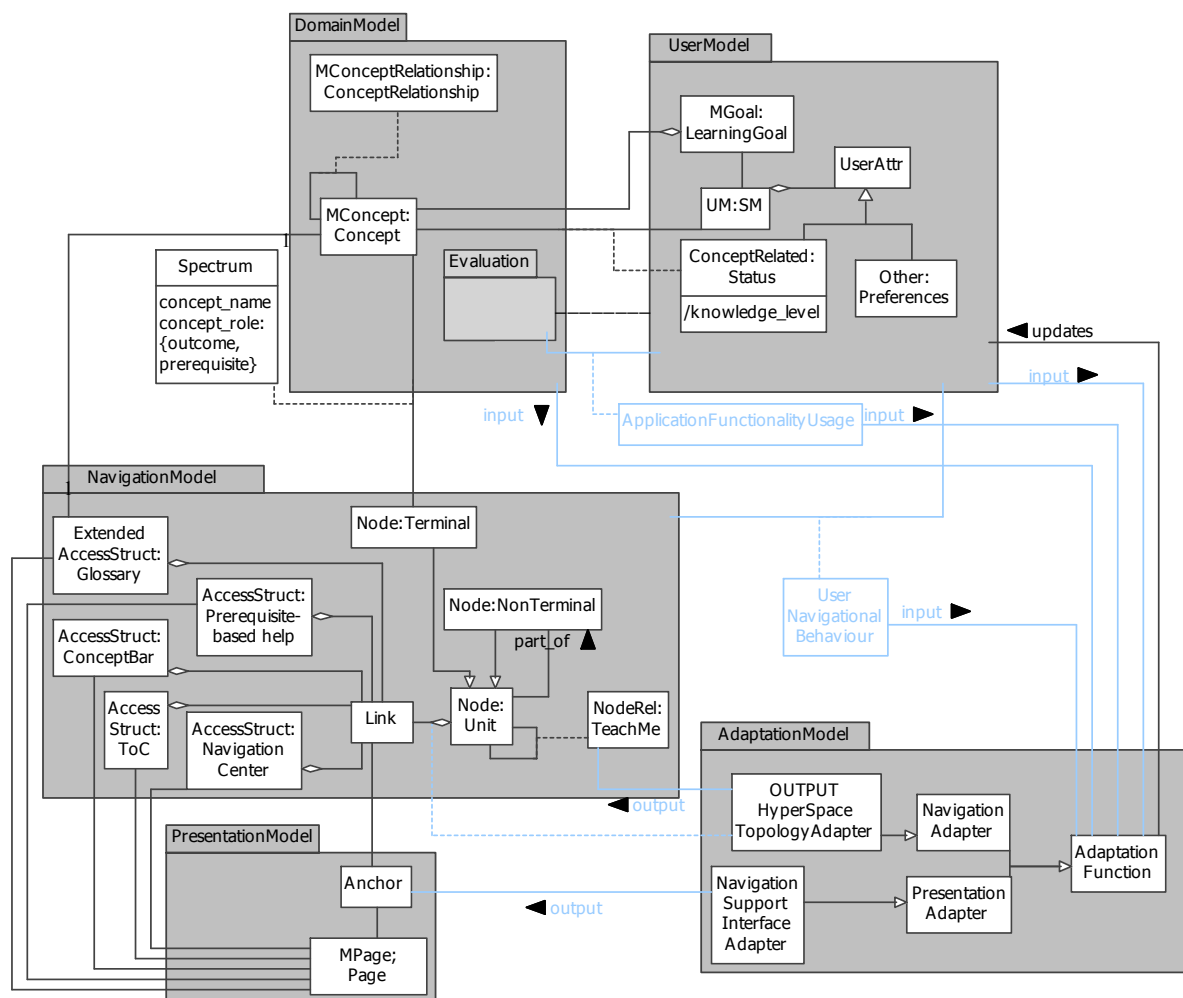


Figura 15 – Instanciação do metamodelo do Interbook

3.5.1. Modelo do Domínio

O Modelo do Domínio é formado pelos conceitos e relações entre eles. Problemas e *quizzes* podem ser usados para atualizar o conhecimento do usuário (Brusilovsky et al., 1998). Tal possibilidade é representada pelo pacote *Evaluation*.

3.5.2. Modelo do Usuário

O Modelo do Usuário é representado pelo Modelo do Estudante (*Student Model*, SM), usado para armazenar a crença (calculada) do sistema sobre o nível de conhecimento do estudante (*knowledge-level*) sobre cada conceito (*Status*). O SM é inicializado através de estereótipos e informações obtidas pela página de registro. Todas as ações do estudante (acesso às páginas, solução de problemas ou *quizzes*) são rastreadas e utilizadas para aumentar ou diminuir os níveis de conhecimento sobre os conceitos envolvidos (Brusilovsky et al., 1998). Os objetivos de aprendizagem (*Learning Goals*), ou seja, o conjunto de conceitos a serem aprendidos, também fazem parte do SM. As visitas às páginas são armazenadas no SM, de uma forma independente dos conceitos. Consideramos que estas visitas estão representadas pela classe *UserNavigationalBehavior*. Atributos independentes do domínio (*Other*) podem ser usados para armazenar preferências (*Preferences*) relativas a características da interface (se deseja marcar páginas já visitadas, se deseja que o texto contenha elos para outras páginas, etc.).

3.5.3. Modelos de Navegação e de Apresentação

No Interbook, a estruturação do conteúdo ocorre no DM, que representa o conhecimento sobre o domínio sendo ensinado através de uma rede, cujos nós correspondem aos conceitos do domínio e cujos elos refletem os vários tipos de relacionamento entre estes conceitos. De acordo com o metamodelo de referência, nós e elos são representados no Modelo de Navegação.

Um ET pode ser definido como qualquer material hipermídia, hierarquicamente estruturado em unidades (*Units*) de diferentes níveis (capítulos, seções e subseções). Cada unidade no nível final da hierarquia (*Terminal*) pode ser uma unidade de apresentação (*atomic presentation*), um exemplo, um

problema ou um teste. Esta classificação não é modelada. Uma barra de navegação (*NavigationCenter*) permite o acesso às unidades do mesmo nível ou de níveis superiores. Um sumário (*Table of Contents, ToC*), onde todas as entradas são elos, também pode ser gerado.

Conceitos são usados como índices para o ET. Para implementar isto, uma lista de conceitos (*Spectrum*) (extensão ao metamodelo de referência) é associada a cada unidade terminal contendo, para cada conceito envolvido, o nome dele (*concept_name*) e o papel (*concept_role*) que ele representa para a unidade: pré-requisito (*prerequisite*) ou consequência (*outcome*, ou seja, a unidade introduz o conceito). Isto possibilita ao sistema saber quais conceitos estão presentes em cada página e, conseqüentemente, quais conceitos precisam ser aprendidos antes que a página possa ser acessada. Uma Barra de Conceitos (*ConceptBar*) permite o acesso aos conceitos organizados de acordo com seus papéis.

O glossário é considerado como uma estrutura de acesso por constituir uma visualização da rede do domínio e exercer a função de indexador: o glossário não só descreve o conceito (quando tal descrição é fornecida pelo autor), como também indexa todas as unidades que o introduzem ou que dele necessitam. Para tanto, o glossário está diretamente associado aos conceitos¹⁵ (cada entrada do glossário corresponde a um dos conceitos do domínio e cada conceito é representado por uma entrada do glossário).

Outra estrutura de acesso presente no InterBook é a ajuda baseada em pré-requisitos que consiste na geração de uma lista ordenada de elos para todas as seções que apresentam informações sobre conceitos relevantes para o entendimento da seção atual (*background concepts*).

Uma vez que a página no Interbook não é composta por fragmentos, estes, bem como o conteúdo dos nós, não foram modelados.

Apesar do Interbook trabalhar com duas janelas principais: uma para o conteúdo (*Textbook*) e outra para o glossário, consideramos isto como uma questão de implementação. Portanto, em nossa modelagem apenas a página (*Page*) e as âncoras (*Anchor*) nela incluídas estão representadas.

¹⁵ Esta associação entre glossário e conceito é uma extensão ao metamodelo de referência.

3.5.4. Modelo de Integração

O Interbook não prevê integração com domínios externos, embora possua o conceito de conjunto (*bookshelf*) constituído por vários ETs sobre um mesmo assunto.

3.5.5. Modelo de Adaptação

A Função de Adaptação: (a) provê orientação direta (*direct guidance*), através do botão *TeachMe*; (b) calcula a lista ordenada de elos que é apresentada pela ajuda baseada em pré-requisitos, sendo esta ajuda requisitada através de um botão (*button*) especial e (c) determina a anotação dos elos, usando ícones, fontes e cores diferentes de acordo com o nível de aprendizado de cada unidade. O metamodelo de referência considera que os dois primeiros tipos são adaptações da topologia do hiperespaço (*Hyperspace Topology*) e estão modelados, respectivamente, como alterações na relação entre os nós (*NodeRel: TeachMe*) e na relação entre nós (*Node*) e elos (*Link*). Como já visto, a anotação (das âncoras) dos elos é considerada adaptação da apresentação, mais especificamente, da interface de suporte à navegação (*Navigation Support Interface*).

Para realizar a adaptação, o InterBook leva em consideração o perfil do estudante – conhecimento sobre o domínio e objetivos – e as ações do estudante, ou seja, o histórico da navegação (páginas acessadas) e a utilização da funcionalidade da aplicação (solução de problemas e *quizzes*). O Modelo de Adaptação caracteriza-se pela indexação do Modelo de Domínio.

3.6. NetCoach

O NetCoach (NetCoach Manual, 2003; Weber & Specht 1997; Weber & Brusilovsky, 2001) (anteriormente denominado ART-Web) é um sistema de autoria para cursos de ensino a distância baseados na *Web*. Ele foi desenvolvido após os resultados positivos obtidos com o sistema ELM-ART para ensino da linguagem de programação LISP. A instanciação do metamodelo do NetCoach pode ser vista na Figura 16.

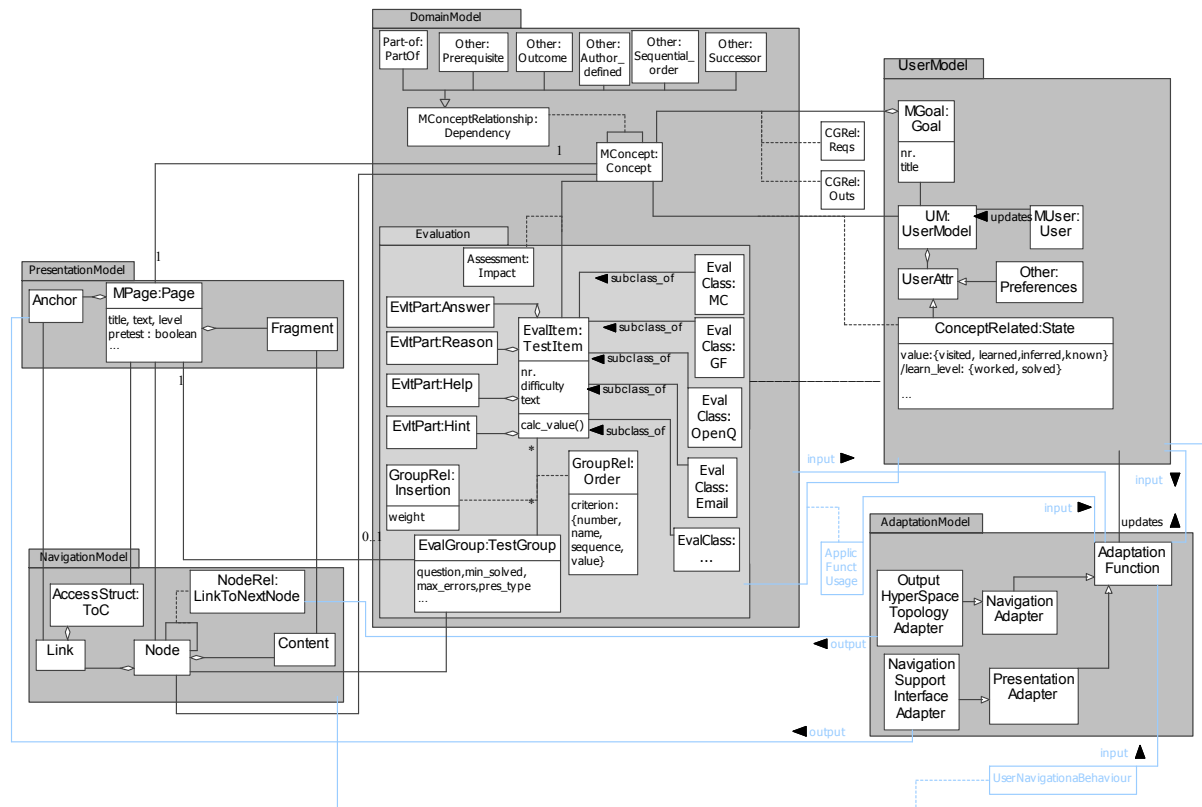


Figura 16 – Instanciação do metamodelo do NetCoach

3.6.1. Modelo do Domínio

O domínio, considerado como sendo uma Base de Conhecimento (*Knowledge Base*), é formado pelos conceitos e pelas relações entre eles. Relações podem ser de pré-requisito (*Prerequisite*) ou inferência (*Outcome*) e devem ser cuidadosamente definidas de acordo com a estrutura do domínio, de modo a propiciar uma adaptação eficiente. Relações de ordenação (*Sequential_order*), sucessão (*Successor*) e parte (*Part-of*) são deduzidas a partir da estrutura hierárquica das páginas. Teoricamente, há a possibilidade da definição de relações por parte dos autores (*Author_defined*).

No NetCoach, os testes desempenham um papel fundamental na avaliação do conhecimento do usuário e conseqüente adaptação do sistema. O esquema de avaliação (*Evaluation*) no NetCoach consiste de Grupos de Testes (*Test Groups*, TGs) associados a páginas (do Modelo de Apresentação)¹⁶. Um TG pertence a exatamente uma página e é composto por Itens de Teste (*Test Items*, TIs). Estes podem pertencer a mais de um Grupo de Teste, sendo ordenados (*Order*) por

¹⁶ Esta relação é implícita no metamodelo de referência.

número, nome ou grau de dificuldade. Um mesmo TI pode ter peso (*weight*) diferente, de acordo com o grupo a que pertence (*GroupRel: Insertion*). Existem diferentes tipos de TIs: múltipla escolha (*multiple choice*, MC), lacuna (*gap filling*, GF), discursivos (*open question*, OP), e-mail e outros. Os TIs podem estar associados a respostas (*Answer*), explicações (*Reason*), textos de ajuda (*Help*) e dicas (*Hint*).

Os resultados obtidos pelo usuário nos testes são utilizados para atualizar o UM e influenciam não só os conceitos aos quais os testes estão diretamente associados, como também, embora em menor grau, os conceitos relacionados a estes. A influência dos testes nos conceitos é representada pela classe de relacionamento *Assessment: Impact*.

3.6.2. Modelo do Usuário

Além do nível de conhecimento (*learn level*) calculado (se os testes associados foram realizados ou resolvidos), o UM também armazena, para cada usuário, informações sobre o estado (*State*) das páginas, isto é, quais páginas foram visitadas, estudadas, inferidas ou quais páginas são conhecidas (respectivamente, *visited*, *learned*, *inferred*, *known*). Estas últimas podem ser definidas pelo próprio usuário. A forma de implementação do UM por camadas (*layer*) possibilita que informações de diferentes fontes não se sobrescrevam (exemplo: informações obtidas com os testes permanecem, mesmo que o usuário altere seu próprio modelo) (Weber et al., 2001).

Os usuários (estudantes) também podem selecionar objetivos (*Goals*) no NetCoach, escolhendo quais conceitos querem aprender. Nesse tipo de aprendizado, alguns conceitos serão indicados como necessários (*Reqs*) para que o estudante possa atingir o objetivo, enquanto outros serão marcados como conhecidos (*Outs*), à medida que o estudante for avançando no seu aprendizado.

Alguns sistemas desenvolvidos com o NetCoach permitem que os usuários indiquem suas preferências (*Preferences*. Exemplos: uso ou não de *frames* no *browser*, opção por *feedback* do sistema para respostas corretas).

3.6.3. Modelos de Navegação e de Apresentação

De acordo com Weber et al. (2001), os conceitos são representações internas

das páginas que serão apresentadas ao estudante e estão relacionados entre si. Assim, uma vez que o NetCoach não faz distinção entre aspectos conceituais (*MConcept*) e de implementação (*MPage*), é necessária a criação (inexistente no metamodelo de referência) de uma relação 1:1 entre *MConcept* e *MPage*. Esse isomorfismo possibilita que a estrutura hierárquica das páginas seja modelada pelas relações entre os conceitos no DM.

O sumário (*Table of Contents, ToC*), por permitir o acesso a qualquer página, a qualquer momento, é considerado como Estrutura de Acesso (*AccessStruct*).

3.6.4. Modelo de Integração

O NetCoach não provê integração com domínios externos. Autores podem importar o conteúdo de um arquivo HTML existente.

3.6.5. Modelo de Adaptação

Nos sistemas desenvolvidos com o NetCoach, os elos são anotados segundo a metáfora do “sinal de trânsito” (adaptação da apresentação). A adaptação da topologia do hiperespaço, refletida na relação entre nós (*LinkToNextNode*), ocorre na visita guiada: o usuário pode solicitar ao sistema uma orientação sobre o melhor caminho de navegação. O sistema, então, usa informações sobre o estado do UM e sobre o objetivo geral de aprendizagem para calcular dinamicamente a melhor página a ser visitada em seguida.

No NetCoach, o conhecimento, preferências e objetivos do usuário; os resultados dos testes (*ApplicationFunctionalityUsage*) e o caminho de navegação escolhido (histórico de navegação) são alguns aspectos levados em consideração para a adaptação. A partir dos parâmetros definidos no TG e das informações do UM, o Modelo de Adaptação executa os cálculos: da avaliação do conhecimento do usuário; da escolha dos testes a serem apresentados e da indicação do próximo passo a ser seguido. A estrutura do domínio, refletida na Base de Conhecimento, também determina o modo como a adaptação ocorre.

3.6.6. Extensões

Além de prover um bloco de notas, onde os usuários podem fazer anotações

sobre o que estão aprendendo, o NetCoach armazena estatísticas sobre a utilização do sistema que permitem aos tutores observar como os usuários estão navegando e resolvendo os testes. Página com referências, glossário – acessíveis através de elos no texto ou de *buttons* – e ferramenta de busca também podem ser implementados. O NetCoach disponibiliza, ainda, ferramentas de comunicação – síncronas e assíncronas – que permitem a interação entre os estudantes, sem atrapalhar o ritmo de aprendizagem de cada um.

3.7. KBS

O KBS *Hyperbook System* (KBS) (Fröhlich et al., 1998; Henze & Nejd 1999a; Henze et al., 1999; Nejd & Wolpers 1999) é uma ferramenta voltada para sistemas hipermídia adaptativos abertos. Sua instanciação é mostrada na Figura 17.

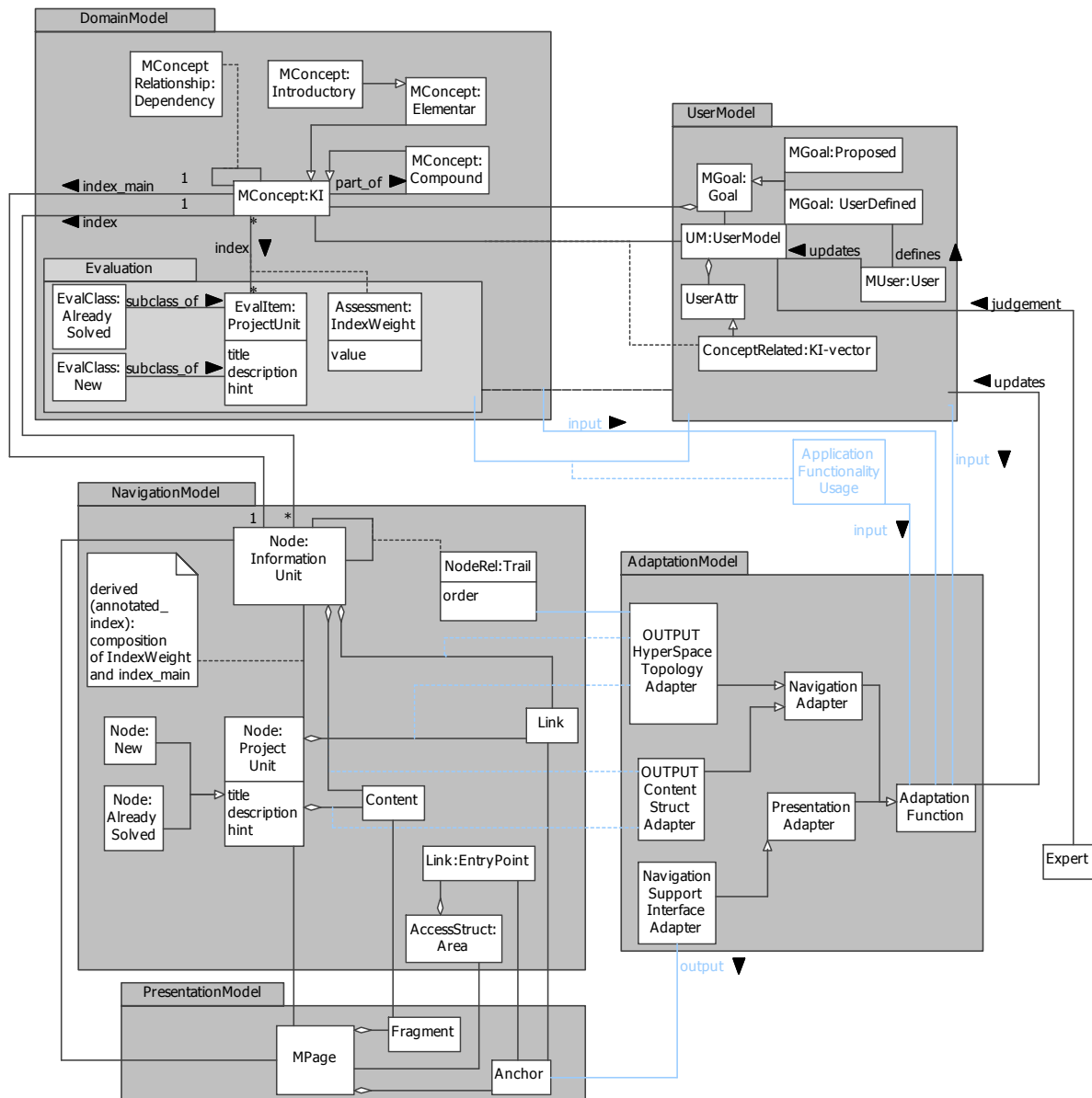


Figura 17 – Instanciação do metamodelo do KBS

3.7.1. Modelo do Domínio

Os conceitos do domínio são representados por Itens de Conhecimento (*Knowledge Items*, KIs), utilizados como índices. Esta indexação pode ser melhor entendida na descrição do Modelo de Adaptação (seção 3.7.5). KIs podem ser elementares (*Elementary*) ou compostos (*Compound*). Conceitos introdutórios (*Introductory*, subtipo de conceitos elementares) são utilizados na implementação. O KBS considera o DM como um Modelo de Conhecimento (*Knowledge Model*, KM) que agrupa os KIs e as relações de dependência (*Dependency*) entre eles.

Unidades de Projeto (*Project Units*, PUs) formam a base do KBS, sendo

representadas no DM por *EvalItem*. PUs podem ser “novas” (*New*) ou “já resolvidas” (*Already Solved*, funcionando como exemplos). Os KIs também são usados como índices ponderados (*IndexWeight*) para as PUs. Tais valores indicam a relevância dos KIs para a execução dos projetos.

3.7.2.

Modelo do Usuário

O UM armazena a estimativa sobre o conhecimento do usuário em relação a cada conceito do KM sob a forma de um vetor de KIs (*KI-vector*), implementado por redes Bayesianas (Henze & Nejd, 1999b). A atualização do UM, baseada no desempenho do usuário (estudante) em algum projeto, é feita a partir de um *feedback* dado pelo próprio usuário (representado pela relação *updates*) ou de uma avaliação feita por especialistas (*expert*, classe que estende o metamodelo de referência).

A aprendizagem por objetivos também é contemplada pelo KBS, sendo que o objetivo (*Goal*) – formado pelos KIs a serem aprendidos – pode ser escolhido pelo usuário (*UserDefined*) ou proposto pelo sistema (*Proposed*).

3.7.3.

Modelos de Navegação e de Apresentação

Unidades de Informação (*Information Units*, IUs) são consideradas nós de navegação. Elas podem estar relacionadas entre si formando uma trilha (*Trail*, seção 3.7.5). O *Hyperbook* pode ser dividido em áreas (*AccessStruct: Area*) que agrupem as IUs, estabelecendo pontos de entrada (*EntryPoints*) que podem ser seguidos pelos estudantes.

Como há uma mistura entre o conceitual e o navegacional, PUs – além de itens de avaliação (seção 3.7.1) – são consideradas também como nós.

As IUs relevantes para os projetos podem ser deduzidas (*derived*), uma vez que são indexadas pelos KIs (vide seção 3.7.5).

O KBS não especifica a apresentação que é, portanto, modelada segundo o metamodelo de referência.

3.7.4.

Modelo de Integração

A abordagem de indexação (seção 3.7.5) utilizada pelo KBS permite que as IUs sejam tratadas de maneira equivalente, independentemente de suas origens,

possibilitando a integração de páginas HTML existentes na *Web* como se elas fossem documentos da própria biblioteca do *Hyperbook*.

3.7.5.

Modelo de Adaptação

O KBS é baseado em problemas / projetos e suas relações com as IUs. A conexão entre o *Hyperbook* e o UM é feita pela indexação de qualquer tipo de recurso de informação (páginas HTML do *Hyperbook*, projetos, exemplos, páginas da *Web*). Assim, KIs indexam IUs – sendo que um único KI é o índice principal (*index_main*) de uma determinada IU – e KIs associados a valores indexam PUs. Desta forma, as IUs relevantes para os projetos podem ser deduzidas (*derived*),

As estratégias de adaptação implementadas pelo KBS são baseadas nos KIs. De acordo com o objetivo e o conhecimento do usuário, o sistema pode:

- a) Sugerir um projeto para o estudante;
- b) Apresentar as IUs relevantes para a tarefa sendo executada (objetivo ou projeto). Estas IUs são apresentadas como um índice ordenado, onde cada elo é anotado (adaptado), usando a metáfora do “sinal de trânsito”, de acordo com o conhecimento do estudante;
- c) Propor uma trilha (*trail*: seqüência ordenada de IUs) a ser seguida. As trilhas podem ser propostas para ajudar o estudante a atingir seu objetivo, resolver um projeto ou completar o curso baseado no KBS.

De acordo com o conhecimento do estudante, o sistema pode propor os objetivos e, a partir destes, sugerir projetos, IUs ou trilhas.

A geração de trilhas, por ser um tipo de orientação direta, é considerada adaptação da topologia do hiperespaço. A escolha dos projetos a serem executados pelo estudante e a determinação de objetivos de aprendizagem podem ser vistas como adaptação da navegação: tanto da estrutura do conteúdo, como da topologia do hiperespaço, uma vez que a navegação muda dependendo dos projetos e objetivos escolhidos. Vale observar que a atribuição de objetivos está representada apenas de forma generalizada pela atualização do UM (relação *updates*), uma vez que os objetivos são somente “disparadores” para as outras formas de adaptação.

A adaptação da apresentação, mais especificamente, da interface de suporte

à navegação ocorre pela mudança na aparência dos elos para indicar a relevância dos mesmos nos pontos de entrada (*entry points*), na indicação do que explorar em seguida (estrutura de navegação) e, como já visto, na sugestão das IUs adequadas a um objetivo / projeto.

Para a adaptação, o KBS leva em consideração o conhecimento do usuário e seus objetivos de aprendizagem (perfil). Por julgar difícil avaliar o conhecimento obtido em visitas às páginas, o KBS não considera o histórico de navegação para a atualização do sistema. Para motivar o estudante a explorar o material de aprendizagem realizando projetos, a atualização só ocorre pela avaliação do desempenho do estudante em algum projeto (funcionalidade da aplicação). Toda a estratégia de adaptação do KBS é baseada nos KIs.

3.7.6. Extensões

No KBS, os estudantes podem “marcar” (*bookmark*) as Unidades de Informação que julgarem interessantes, assim como podem fazer anotações sobre elas. A integração dos projetos dos estudantes ao *Hyperbook* é baseada em *portfolios*, usados para demonstrar os conceitos utilizados nos projetos. As *homepages* destes estudantes também são integradas ao *Hyperbook*. O KBS gera um glossário a partir do KM.

3.8. Comparação entre AHSs

Como visto, o metamodelo de referência provê um vocabulário comum que permite comparar os sistemas. A comparação é feita a partir de cada metamodelo em relação aos aspectos fundamentais da adaptação: “o quê”, “em função de quê” e “como”. A Tabela 5 apresenta o quadro comparativo dos sistemas estudados. O AHAM não é considerado, pois é um modelo e não um sistema implementado. A granularidade de tempo, por ser ainda um objeto de estudo, não aparece. Observa-se, entretanto, que em todos os sistemas o UM persiste entre as sessões e os sistemas são adaptados continuamente, de acordo com os acessos feitos (visita às páginas) e após a resolução de testes / problemas, caso haja. O Interbook pode customizar a adaptação previamente, de acordo com o estereótipo do usuário e alguns sistemas desenvolvidos com o NetCoach aplicam um teste inicial para medir o conhecimento prévio.

		AHA!	InterBook	NetCoach	KBS
O Q U Ê	Apresentação: conteúdo				
	Apresentação: navegação	anotação de elos			
	Navegação: Estrutura do Conteúdo	fragmentos condicionais			atribuição projetos e objetivos; indicação Unidades de Informação (IUs).
	Navegação: Topologia do Hiperespaço	elos condicionais	orientação direta (botão "TeachMe"); ajuda baseada em pré-requisitos (<i>special button</i>).	orientação direta (botão "Next")	atribuição projetos e objetivos; indicação IUs; geração trilhas (<i>trails</i>).
	Conteúdo				
E M F U N Ç Ã O D E	Modelo do Domínio (DM)	conceitos e relações; misturado com AM.	rede: nós (conceitos) e elos (relações)	Base de Conhecimento: conceitos + relações	Modelo de Conhecimento (<i>Knowledge Model</i>): Itens de Conhecimento (KIs) + relações
	Perfil do usuário (<i>profile</i>)	conhecimento	conhecimento, objetivos, preferências.	conhecimento, objetivos, preferências.	conhecimento; objetivo escolhido ou proposto.
	Contexto (Situação de uso)	teoricamente, pode ser implementado da mesma forma que o perfil do usuário.			
	Histórico de Navegação	páginas visitadas			NÃO CONSIDERADO
Q U Ê	Utilização funcionalidade aplicação	solução de testes	solução problemas e <i>quizzes</i>	solução de testes / problemas	desempenho em projetos
	Histórico de Interação				
C O M O	Modelo do Usuário (UM)	pares atributo-valor implementados por tabela	pares conceito-valor	relação entre conceito e usuário armazenada em camadas (independência da atualização)	vetor de KI; rede Bayesiana.
	Modelo de Adaptação (AM)	regras de geração / requisitos	indexação baseada no Modelo do Domínio	computações baseadas nos resultados dos testes e na estruturação do domínio	baseada nos Itens de Conhecimento (KIs)
	Agente de entrada (<i>input</i>)	formulário registro + valores <i>default</i>	estereótipo (página de registro)	itens de teste / problemas	desempenho nos projetos julgado pelos estudantes ou por especialistas
	Agente de atualização (<i>update</i>)	sistema / usuário	Sistema	sistema / usuário	

Tabela 5 – Comparação de alguns AHSs

Nenhum dos sistemas estudados adapta a apresentação do conteúdo, nem o conteúdo propriamente dito. Por outro lado, todos se valem da anotação de elos para orientar a navegação, embora o tipo de anotação varie entre os sistemas. Para

a adaptação da navegação, diferentes técnicas são utilizadas.

A definição do Modelo do Domínio varia entre os sistemas, principalmente porque não há uma distinção clara entre os aspectos conceituais, navegacionais e de implementação. Em essência, o domínio é composto por conceitos estruturados hierarquicamente (relação de subparte) e as relações entre eles.

O AHA! considera que a adaptação ocorre através de eventos que podem ser implementados por módulos. Além disto, o mecanismo de adaptação se baseia em atributos que podem não ser dependentes do domínio. Assim, haveria a possibilidade de se realizar a adaptação em função da situação de uso do sistema. A interação entre os usuários e os sistemas é, basicamente, a navegação. Outras ações não são consideradas para a adaptação. O KBS se distingue por não levar em conta a navegação, se baseando no desempenho em projetos.

Para todos os sistemas, o Modelo do Usuário (ou Modelo do Estudante, quando uma terminologia educacional é utilizada) representa o relacionamento entre o usuário e o domínio, sendo usado, essencialmente, para armazenar a crença que o sistema tem sobre o conhecimento do usuário a respeito dos conceitos do domínio. O KBS se diferencia por usar uma rede Bayesiana para a modelagem. O NetCoach implementa o UM em camadas (*layer*), com atualização independente de cada uma delas (Weber et al., 2001).

O mecanismo de adaptação é diferente para cada um dos sistemas estudados. Uma característica marcante do KBS é a separação entre o conhecimento e a informação que permite aos professores determinarem a estrutura navegacional ou conceitual desejada para o material didático, personalizando-o, e torna o sistema robusto a mudanças. O KBS possui, ainda, um esquema peculiar de indexação que permite sua abertura (*openness*). Outra particularidade é a utilização apenas dos projetos para a atualização do sistema.

O InterBook usa estereótipos para a configuração inicial da aplicação. O NetCoach oferece a opção de se fazer uma avaliação preliminar do conhecimento do usuário. Sistemas desenvolvidos com o AHA! e com o NetCoach podem permitir que usuários atualizem o próprio modelo. O KBS permite indiretamente, através da auto-avaliação nos projetos. O NetCoach e o InterBook podem possibilitar ao usuário a alteração de funcionalidades de interface que influenciam no nível de adaptação realizado. O usuário do AHA! pode mexer na adaptação, em menor escala, pela escolha do sistema de cores para a anotação de elos.