

# Tvorba obsahu v adaptívnych hypermédiách pre e-vzdelávanie

Mária Bielíková\*

bielik@fiit.stuba.sk

Jaroslav Kuruc\*

kuruc@fiit.stuba.sk

**Abstrakt:** V príspevku sa zaoberáme aspektmi návrhu modelov adaptívneho výučbového systému a podporou ich naplňania obsahom. Z tohto pohľadu je najviac diskutovaný doménový model, keďže tento je špecifický pre každý systém. Ide vlastne o priestor vedomostí, ktoré sa prezentujú používateľovi (študentovi). Zaoberáme sa možnosťami štruktúrovania priestoru vedomostí a definovaním vzťahov medzi konceptmi v tomto priestore. Druhou oblasťou je definovanie a naplnenie samotného informačného priestoru, ktorý reprezentuje obsah konceptov. Uvedieme príklady návrhu modelov, ktoré sa používajú v kontexte adaptívnych systémov pre vzdelávanie. V príspevku sa podrobnejšie venujeme systému AHA!, verzia 2.0 – adaptívnemu webovému systému. Ide o otvorený systém vytvorený s cieľom širšie sprístupniť adaptívne hypermédie.

**Kľúčové slova:** adaptívne hypermédie, e-vzdelávanie, model domény, model používateľa, prispôsobovanie, AHA!

## 1 Úvod

Adaptívne hypermédie (AH) umožňujú prispôsobenie prezentácie (obsahu aj výzoru) a tiež navigácie jednotlivým používateľom, resp. skupinám používateľov. Vhodnosť použitia adaptívnych hypermédií v aplikáciách pre výučbu je daná tým, že efektívnosť učenia sa významne závisí od možnosti prispôsobenia študijného materiálu jednotlivým študentom. Nie náhodou medzi prvými AH systémami nájdeme práve systémy pre vzdelávanie [6]. Použitie AH systémov v spojení s prostredím Internetu, najmä služby World Wide Web, v určitom zmysle ohraničuje použité technológie, ale na druhej strane umožňuje prenosnosť a podporuje distribuovanosť vzdelávania [1].

Prispôsobovanie v AH systéme je založené na vedomostiach o obsahu jednotlivých stránok, väzbách medzi nimi a predpokladoch o vedomostiach, preferenciách a ďalších charakteristikách študenta. Podstatnou činnosťou pri tvorbe AH systémov je práve vytvorenie obsahu a tiež získanie a reprezentácia vedomostí o prispôsobovaní. V súčasnosti sa využívajú najmä prístupy založené na explicitnej reprezentácii vedomostí prostredníctvom pravidiel. Vedomosti sú uložené aj implicitne v štruktúre informačného priestoru (napr. definovaním vzťahu prerekvizít medzi konceptmi). Ďalšou kategóriou vstupov, ktoré sa využívajú pri prispôsobovaní sú informácie zozbierané počas používania AH systému (reprezentujú správanie používateľa a možno z nich odvodiť hypotézy o úrovni vedomostí používateľa).

V tomto príspevku sa zameriame na podporu tvorby modelov adaptívneho hypermediálneho systému. Diskutujeme možnosti štruktúrovania informačného priestoru AH systému a podporu jeho naplňania. Tvorbu obsahu AH systémov vysvetlíme na príklade systému AHA!, ktorý bol vyvinutý na univerzite v Eindhovene s cieľom širšie sprístupniť adaptívne hypermédie.

---

\* Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave, Ilkovičova 3, 842 16 Bratislava

## 2 Návrh obsahu adaptívneho sídla pre e-vzdelávanie

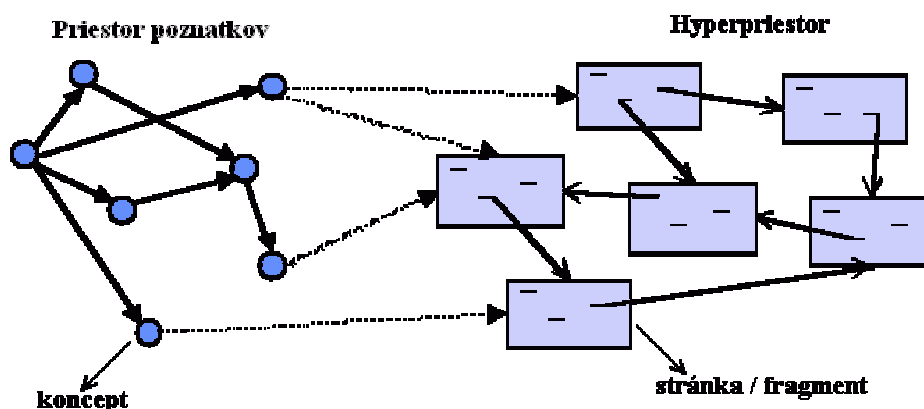
### 2.1 Návrh modelov

AH systém zahŕňa tieto modely [2]:

- *model aplikačnej domény*, ktorý opisuje štruktúru obsahu hypermediálneho systému,
- *informačná báza*, ktorá obsahuje materiál prezentovaný používateľovi,
- *model prispôsobovania*, kde sú vyjadrené vedomosti o prispôsobovaní,
- *model používateľa* a/alebo *model prostredia*, ktoré predstavujú zdroj prispôsobovania.

Aktívnym prvkom systému je *mechanizmus prispôsobovania*, ktorý zabezpečuje prispôsobovanie.

V prípade AH systému pre e-vzdelávanie model aplikačnej domény predstavuje priestor vedomostí (poznatkov), ktoré systém prezentuje používateľovi – študentovi. Najčastejšie sa reprezentuje prostredníctvom konceptov. Informačná báza (hyperpriestor) obsahuje študijný materiál a často sa rozčleňuje na informačné fragmenty (pozri obr. 1). V prípade webových AH systémov pre e-vzdelávanie informačný fragment predstavuje stránku alebo časť stránky.



Obr. 1. Štruktúra informačného priestoru AH systému pre e-vzdelávanie [7].

V prvých AH systémoch pre e-vzdelávanie vzťah medzi konceptmi a informačnými fragmentmi bol jednoduchý v tom zmysle, že každému konceptu zodpovedal jeden informačný fragment reprezentovaný web stránkou. Na efektívne použitie AH systému sa ukazuje takáto granularita ako nepostačujúca. V súčasnosti existuje niekoľko systémov, ktoré používajú indexovanie fragmentov stránok, t.j. stránky sú rozdelené do fragmentov a definujú sa vzťahy s konceptmi v modeli aplikačnej domény (vedomosť reprezentovaná konceptom sa realizuje príslušným informačným fragmentom). Takýto spôsob využíva napr. systém WHURLE [4] alebo AHA! [10].

Na štruktúrovanie doménového modelu (priestor vedomostí reprezentovaný konceptmi) sa často používa relácia *jeRodič*, pomocou ktorej možno definovať hierarchiu konceptov (podobne ako v knihe máme kapitoly, podkapitoly a pod).

### 2.2 Naplnenie modelov obsahom

Naplnenie informačného priestoru spočíva vo vytvorení učebných textov prislúchajúcich jednotlivým konceptom. V súčasných systémoch sa používajú dva prístupy: prístup založený na značkách (angl. markup) a prístup založený na využití grafického rozhrania (GUI) [7]. Napĺňanie sa vo všeobecnosti nelíši od napĺňania iných (hypermediálnych, webových)

aplikácií. Špecifické je získanie a reprezentácia znalostí o prispôsobovaní, ktoré sú často „prepletené“ s modelom aplikačnej domény (napr. v systémoch AHA! [10] a TIM [3] sa pravidlá pre prispôsobovanie definujú ako hodnoty atribútov konceptov).

Prístupy založené na značkách umožňujú vo všeobecnosti využitie všetkých navrhnutých vlastností reprezentácie. Je možné vytvoriť niekoľko rôznych reprezentácií (na rôznych úrovniach abstrakcie), ktoré sa transformáciami prevedú do vnútornej formy, s ktorou pracuje mechanizmus prispôsobovania. Prvým systémom, ktorý využíval prístup založený na značkách bol InterBook systém, kde autor vytváral obsah v MS Word [5]. Súčasný systém využívajú jazyk XML. Výhodou je veľká výrazová sila, nevýhodou možnosť chýb a najmä nevyhnutnosť vedomostí súvisiacich so syntaxou a sémantikou zápisu.

GUI prístupy riešia uvedený problém. Vo väčšine prípadov využívajú grafické rozhranie prostredníctvom formulárov. Vzhľadom na zložitosť informačného priestoru väčšiny AH systémov je návrh a aj napĺňanie obsahu neprehľadné a vyžaduje, aby si autor udržoval informácie o zložitých väzbách medzi jednotlivými elementami.

### **3 Príklad – tvorba obsahu v systéme AHA!**

V súčasnosti existuje viacero adaptívnych systémov pre e-vzdelávanie. Jedným z nich je aj systém AHA! [10], ktorý sa vyvíja s cieľom vytvorenia univerzálneho prostriedku pre tvorbu adaptívnych systémov, a to nielen pre oblasť e-vzdelávania. Tento cieľ je zrejmý aj z jeho názvu, kde AHA! znamená „Adaptive Hypermedia for All“.

Ide o otvorený webový systém. Je založený na princípe prispôsobovania webového obsahu, ktorý poskytuje. Samotný obsah sa definuje vo forme adaptívnych prezentácií. Systém AHA! vychádza z modelu AHAM [8] (Adaptive Hypermedia Application Model), t.j. definuje sa doménový model, model prispôsobovania a model používateľa.

#### **3.1 Doménový model a model prispôsobovania**

Definícia doménového modelu spočíva vo vytvorení grafu konceptov. Tento graf vyjadruje závislosti medzi jednotlivými konceptmi a zvyčajne tak určuje spôsob, v akom poradí by sa mali jednotlivé koncepty prezentovať používateľovi. AHA! spája špecifikáciu tohto modelu so špecifikáciou modelu prispôsobovania. Autor pre každý koncept definuje tieto vlastnosti:

- jedinečné meno konceptu, na základe ktorého sa tento koncept používa v pravidlách prispôsobovania,
- opis konceptu, ktorý slúži ako pomocný text pre autora,
- zdroj prislúchajúci danému konceptu v informačnom priestore,
- výraz vyjadrujúci vhodnosť daného konceptu, na základe ktorého sa neskôr prispôbujú odkazy na tento koncept,
- atribúty, ktoré definujú typy informácií uchovávané pre jednotlivé koncepty v modeli používateľa,
- pravidlá prispôsobovania, na základe ktorých sa pri zobrazení daného konceptu modifikuje model používateľa.

Vhodnosť konceptu sa vyjadruje výrazom, ktorý sa vyhodnocuje v prípade odkazovania sa na tento koncept. Na základe výsledku vyhodnotenia dochádza k prispôsobovaniu spomínaného odkazu. V prípade, že koncept sa vyhodnotí ako nevhodný, dochádza k skrývaniu odkazu, t.j. odkaz sa zobrazí ako normálny text (bez podčiarknutia, ktoré signalizuje odkaz). V prípade vhodnosti konceptu je výzor odkazu prispôbovaný formou zobrazenia v rôznych farbách na základe toho, či koncept, na ktorý daný odkaz smeruje, už bol v minulosti zobrazený.

Definícia konceptu tiež obsahuje definíciu atribútov, ktorých hodnoty sa pre každého používateľa ukladajú v modeli používateľa. Na základe hodnôt týchto atribútov dochádza k prispôsobovaniu prezentácie pre toho ktorého používateľa. Definícia atribútu konceptu obsahuje tieto vlastnosti:

- meno atribútu jedinečné v rámci daného konceptu, na základe ktorého sa tento atribút konceptu adresuje v pravidlách prispôsobovania,
- opis atribútu, ktorý slúži ako pomocný text pre autora,
- typ atribútu (boolovská hodnota, celé číslo alebo reťazec),
- perzistencia, teda či je hodnota tohto atribútu len dočasná počas vyhodnocovania pravidiel prispôsobovania alebo sa ukladá,
- príznak, či ide o systémový atribút,
- príznak, či si sám používateľ môže hodnotu tohto atribútu meniť,
- prednastavená (štandardná) hodnota.

Každý atribút môže navyše obsahovať ľubovoľný počet pravidiel prispôsobovania, ktoré určujú, akým spôsobom sa mení model používateľa (hodnota niektorých z atribútov konceptov) pri zmene hodnoty daného atribútu. AHA! definuje pre každý koncept povinný atribút boolovského typu s názvom *access*, ktorého hodnota sa v prípade zobrazenia daného konceptu dočasne nastaví na *true*. Je tak možné definovať pravidlá prispôsobovania, ktoré sa vykonajú pri zobrazení konceptu. Definícia pravidla prispôsobovania obsahuje tieto vlastnosti:

- príznak, či sa pri zmene hodnoty niektorého z atribútov konceptu počas vyhodnocovania tohto pravidla majú vyhodnotiť aj pravidlá asociované s menenými atribútmi,
- podmienka, na základe ktorej je vyhodnotené pravidlo,
- akcie, ktoré sa majú vykonať v prípade splnenia predchádzajúcej podmienky,
- akcie, ktoré sa majú vykonať v prípade nesplnenia predchádzajúcej podmienky.

Okrem pravidiel prispôsobovania definovaných pre jednotlivé atribúty obsahuje model prispôsobovania taktiež mechanizmy na prispôsobovanie informačného priestoru, ktorý prislúcha jednotlivým konceptom. Poskytuje mechanizmus skrývania fragmentov, kedy sa blok textu zobrazí iba v prípade splnenia predpísaného pravidla, resp. v prípade jeho nesplnenia sa zobrazí alternatívny blok textu. Ďalším už spomínaným mechanizmom prispôsobovania informačného priestoru je prispôsobovanie odkazov. Odkazy môžu byť nepodmienené, kedy sa zobrazujú vždy rovnako (a teda sa neprispôbujú) alebo môžu byť podmienené, čo znamená, že spôsob ich zobrazenia závisí od vhodnosti konceptu, na ktorý odkazujú. Počas prispôsobovania takýchto odkazov dochádza buď k skrývaniu odkazov alebo k ich anotovaniu.

### 3.2 Model používateľa

Úlohou modelu používateľa v systéme AHA! je udržiavať hodnoty atribútov konceptov pre jednotlivých používateľov. Hodnoty atribútov sa počas používania systému menia, najmä z dôvodu (predpokladanej) zmeny stavu vedomostí používateľa. Tieto hodnoty sú dôležité pre riadenie samotného prispôsobovania. Požiadavka na zobrazenie časti informačného priestoru prislúchajúcej niektorému z konceptov sa v systéme AHA! spracuje takto:

1. Klient zašle serveru požiadavku na zobrazenie stránky.

2. Server identifikuje prislúchajúci koncept a na základe pravidiel definovaných pre atribút *access* tohto konceptu vykoná akcie, ktoré zabezpečia zmeny v modeli používateľa.
3. Server vyhodnotí prípadné podmienky určujúce zobrazenie/skrytie podmienených blokov zobrazovanej stránky a na základe výsledku ich do zobrazovaného textu buď zahrnie alebo nezahrnie. Taktiež sa prispôbia podmienené odkazy.
4. Výslednú stránku pošle server späť klientovi.

Atribúty konceptov vo svojej definícii obsahujú informáciu o tom, či je možné ich hodnoty v modeli používateľa zo strany samotného používateľa meniť. Adaptívna prezentácia tak môže poskytovať rozhranie, cez ktoré možno realizovať zmenu hodnôt v modeli používateľa.

Systém AHA! definuje pre každú prezentáciu povinný koncept s názvom *personal*, ktorý slúži ako prostriedok na ukladanie preferencií používateľa, ktoré tiež možno brať do úvahy v modeli prispôsobovania. Každá takáto preferencia sa definuje ako atribút tohto konceptu.

### 3.3 Príklad definície adaptívnej prezentácie

Definíciu doménového modelu a modelu prispôsobovania si ukážeme na príklade (príklad je prevzatý z [9]). Na ich reprezentáciu používa AHA! alternatívne formát XML, resp. relačnú databázu MySQL. Príklad, ktorý uvádzame, ukazuje reprezentáciu pomocou formátu XML (táto reprezentácia je mierne upravená z dôvodu prezentovateľnosti).

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE conceptList SYSTEM 'concept.dtd'>

<conceptList>
  <name>Belgian Beer Example</name>
  <concept>
    <name>de_koninck</name>
    <desc>first author's favorite beer</desc>
    <resource>de_koninck.xhtml</resource>
    <req>beer.interest > 50</req>

    <attribute name="access" type="bool"
      isPersistent="false" isSystem="true"
      isChangeable="false">
      <default>>false</default>

    <generateListItem isPropagating="true">
      <req>beer.interest < 100</req>
      <>trueActions>
        <action>
          <concept>beer</concept>
          <attribute>interest</attribute>
          <expr>beer.interest + 10</expr>
        </action>
      </trueActions>
    </generateListItem>

    <generateListItem isPropagating="true">
      <req>chocolate.interest < 50 and
        chocolate.interest > 4</req>
      <>trueActions>
        <action>
          <concept>chocolate</concept>
          <attribute>interest</attribute>
          <expr>chocolate.interest - 5</expr>
        </action>
      </trueActions>
    </generateListItem>

    <generateListItem isPropagating="true">
      <req>beer.interest > 50</req>
      <>trueActions>
        <action>
          <concept>de_koninck</concept>
          <attribute>knowledge</attribute>
          <expr>100</expr>
        </action>
      </trueActions>
      <falseActions>
        <action>
          <concept>de_koninck</concept>
          <attribute>knowledge</attribute>
          <expr>35</expr>
        </action>
      </falseActions>
    </generateListItem>

    <attribute name="knowledge" type="bool"
      isPersistent="true" isSystem="false"
      isChangeable="true">
      <default>0</default>
    <generateListItem isPropagating="true">
      <req>>true</req>
      <>trueActions>
        <action>
          <concept>beer</concept>
          <attribute>knowledge</attribute>
          <expr>beer.knowledge +
            0.2*_de_koninck.knowledge</expr>
        </action>
      </trueActions>
    </generateListItem>
  </concept>
  ...
</conceptList>

```

Príklad obsahuje definíciu konceptu *de\_koninck*, ktorému v informačnom priestore prislúcha zdroj (stránka) *de\_koninck.xhtml*. V prípade odkazovania sa na stránku prislúchajúcu tomuto konceptu je príslušný odkaz prispôsobený na základe splnenia podmienky *beer.interest > 50*. Táto podmienka vyjadruje záujem daného používateľa o informácie prislúchajúce konceptu *beer* vyjadrený hodnotou jeho atribútu *interest*. V prípade nesplnenia tejto podmienky (hodnota atribútu je menšia alebo rovná hodnote 50), dochádza k skrytiu odkazu (prezentuje sa ako bežný text bez podčiarknutia). Naopak, v prípade splnenia tejto podmienky (hodnota atribútu je väčšia ako hodnota 50), dochádza k anotovaniu odkazu (odkaz systém prezentuje farebne, v závislosti od toho, či už v minulosti bol zobrazený alebo nie).

Pri požiadavke zobrazenia stránky prislúchajúcej konceptu *de\_koninck* systém dočasne nastaví hodnotu atribútu *access* tohto konceptu na hodnotu *true*. Keďže tento atribút nie je perzistentný, jeho hodnota sa neukladá v modeli používateľa a po vyhodnotení pravidiel prispôsobovania asociovaných so zmenou hodnoty tohto atribútu sa jeho hodnota opäť nastaví na prednastavenú hodnotu.

Spomínaný atribút má asociované tri pravidlá prispôsobovania. Prvé z nich zabezpečuje zvýšenie záujmu o informácie prislúchajúce konceptu *beer* (atribút *interest*), avšak iba dotedy, kým jeho hodnota nedosiahne hodnotu 100. Druhé z nich zabezpečuje zníženie záujmu o informácie prislúchajúce konceptu *chocolate*, avšak iba v prípade, že tento nie je príliš vysoký (jeho hodnota je menšia ako hodnota 50), pričom k zníženiu dôjde iba vtedy, ak sa hodnota nezníži pod hranicu 0. Tretie pravidlo zabezpečí, že používateľom, ktorí sa zaujímajú o informácie prislúchajúce konceptu *beer*, sa stanoví znalosť konceptu *de\_koninck* (atribút *knowledge*) na hodnotu 100, kým v prípade nezájmu sa jeho znalosť stanoví na hodnotu 35.

Koncept *de\_koninck* má tiež definovaný atribút *knowledge*, ktorý vyjadruje znalosť tohto konceptu. Keďže ide o perzistentný atribút, jeho hodnota sa ukladá v modeli používateľa. Prednastavená hodnota znalosti tohto konceptu je stanovená na hodnotu 0. Pravidlo asociované s týmto atribútom vyjadruje, že akákoľvek zmena znalosti konceptu *de\_koninck* má za následok aj zmenu znalosti konceptu *beer*, pričom k znalosti konceptu *beer* sa pripočíta hodnota, o ktorú sa mení znalosť konceptu *de\_koninck* (vyjadrené predpisom *\_de\_koninck.knowledge*), vynásobená koeficientom 0,2.

Ako je z uvedeného príkladu zrejmé, pri zobrazení stránky *de\_koninck.xhtml* dochádza k zmene záujmu o informácie prislúchajúce konceptom *beer* a *chocolate* a taktiež dochádza k zmene znalosti konceptu *de\_koninck*. Pri zmene znalosti tohto konceptu sa vyhodnotí príslušné pravidlo, čo zase spôsobuje zmenu znalosti konceptu *beer*. Zmena záujmu o informácie prislúchajúce konceptom *beer* a *chocolate*, ako aj zmena znalosti konceptu *beer* spôsobuje vyhodnotenie pravidiel asociovaných s príslušnými atribútmi týchto konceptov, tie však už uvedený príklad neobsahuje.

Definovaním pravidla prispôsobovania môže nastať prípad, kedy zmena hodnoty atribútu má za následok zmenu hodnoty iného atribútu, ktorý zase mení hodnotu prvého atribútu. Inými slovami, môže dôjsť k zacykleniu pri vyhodnocovaní pravidiel prispôsobovania. Súčasťou definície pravidla je preto aj príznak *isPropagating*, ktorý vyjadruje, či sa pri jeho vyhodnocovaní majú vyhodnocovať aj prípadné ďalšie pravidlá vyplývajúce z dôsledkov pôvodného pravidla, mnohokrát vyvolané zmenou hodnoty niektorého ďalšieho atribútu.

Model používateľa, v ktorom sa pre jednotlivých používateľov ukladajú hodnoty atribútov konceptov, systém AHA! reprezentuje taktiež pomocou formátu XML. Týmto spôsobom sa rieši potreba perzistencie niektorých atribútov konceptov.

Nasledujúci príklad znázorňuje formát ukladania modelu používateľa v jazyku XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE profile SYSTEM 'profile.dtd'>
<profile>
  <record>
    <key>personal.id</key>
    <type>string</type>
    <persistent>true</persistent>
    <value>debra</value>
  </record>
  <record>
    <key>personal.email</key>
    <type>string</type>
    <persistent>true</persistent>
    <value>debra@win.tue.nl</value>
  </record>
  <record>
    <key>personal.goodlink</key>
    <type>string</type>
    <persistent>true</persistent>
    <value>0000ff</value>
  </record>
  <record>
    <key>personal.badlink</key>
    <type>string</type>
    <persistent>true</persistent>
    <value>000000</value>
  </record>
  ...
  <record>
    <key>de_koninck.knowledge</key>
    <type>int</type>
    <persistent>true</persistent>
    <value>35</value>
  </record>
  <record>
    <key>beer.knowledge</key>
    <type>int</type>
    <persistent>true</persistent>
    <value>20</value>
  </record>
  <record>
    <key>beer.interest</key>
    <type>int</type>
    <persistent>true</persistent>
    <value>40</value>
  </record>
  <record>
    <key>chocolate.knowledge</key>
    <type>int</type>
    <persistent>true</persistent>
    <value>15</value>
  </record>
  <record>
    <key>chocolate.interest</key>
    <type>int</type>
    <persistent>true</persistent>
    <value>20</value>
  </record>
  ...
</profile>
```

Ako sme už uviedli vyššie, model používateľa obsahuje hodnoty atribútov preddefinovaného konceptu *personal*. Okrem toho obsahuje aj atribúty autorom definovaných konceptov, v tomto prípade tak, ako boli definované v predchádzajúcom príklade. Ide o atribút *knowledge* pre koncepty *de\_koninck*, *beer* a *chocolate*, ako aj atribút *interest* pre koncepty *beer* a *chocolate*.

Stránky informačného priestoru sú v systéme AHA! reprezentované pomocou formátu XHTML, ktorý je rozšírený o špeciálne značky umožňujúce definovanie podmienených blokov. Nižšie je príklad definície takéhoto bloku:

```
<if expr="beer.knowledge > 50">
  <block>text</block>
</if>
<if expr="beer.knowledge > 50">
  <block>text A</block>
  <block>text B</block>
</if>
```

Ako je z príkladu zrejmé, existujú dve alternatívy definície. Prvá z nich (vľavo) v prípade splnenia podmienky *beer.knowledge > 50* zabezpečí zobrazenie textu v bloku. Druhý spôsob definície v prípade splnenia rovnakej podmienky zabezpečí zobrazenie textu A, v opačnom prípade zobrazenie textu B.

Okrem podmienených blokov je možné v stránkach informačného priestoru prispôbovať aj odkazy. V nasledujúcom príklade uvádzame ako prvú definíciu podmieneného odkazu, za ktorou nasleduje definícia nepodmieneného odkazu.

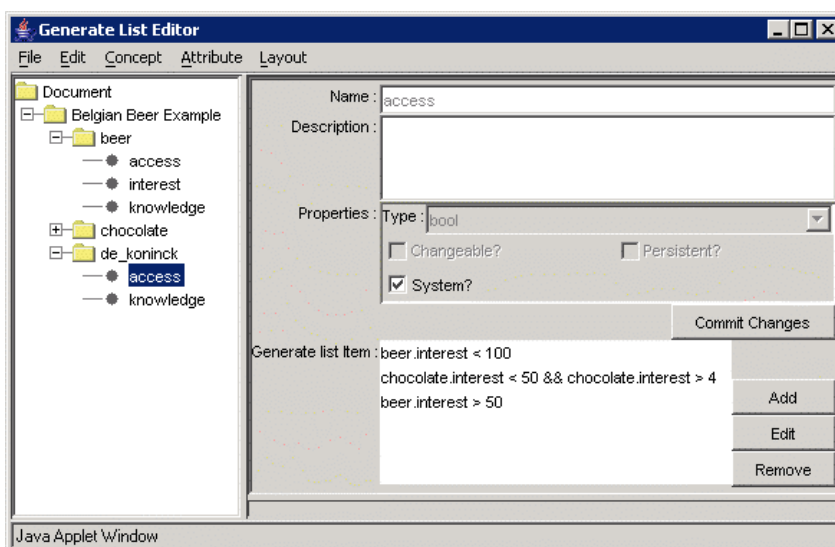
```
<a href="de_koninck.xhtml" class="conditional">link</a>
<a href="de_koninck.xhtml" class="unconditional">link</a>
```

### 3.4 Podporné prostriedky na definíciu adaptívnej prezentácie

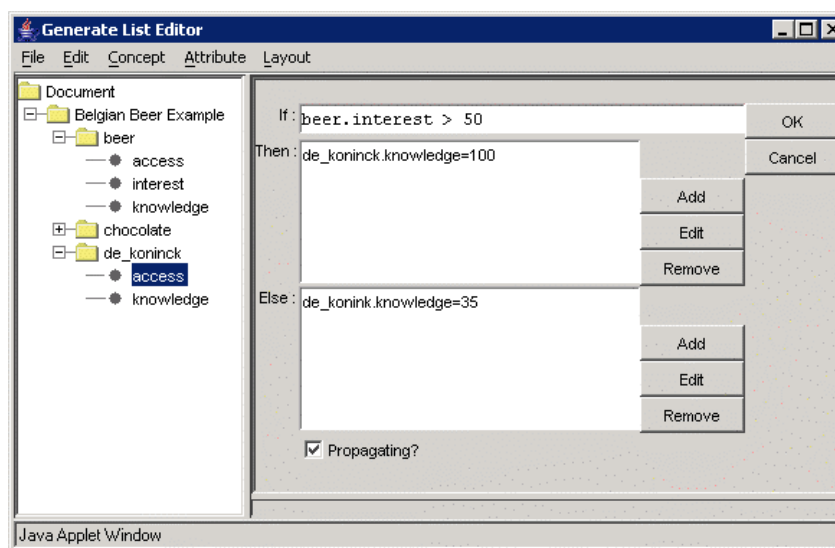
Definovanie doménového modelu a modelu prispôbovania vo formáte XML môže byť pre autora adaptívnej prezentácie mnohokrát príliš zložitý a neprehľadný. Navyše autorom obsahu systému pre e-vzdelávanie môže byť (a často aj je) pedagóg, ktorý nemusí byť dostatočne technicky zdatný. Aj z tohto dôvodu AHA! poskytuje dva podporné prostriedky, ktorých úlohou je uľahčiť a spríjemniť proces navrhovania tejto prezentácie. Sú to prostriedky *Concept Editor* a *Graph Author*.

*Concept Editor* je softvérový prostriedok, ktorý umožňuje definovať koncepty, ich atribúty, pravidlá vhodnosti ako aj pravidlá prispôbovania. Cieľom je odbremeniť autora od potreby znalosti syntaxe XML, v ktorej je samotná definícia zapísaná. Poskytuje GUI rozhranie, v ktorom je možné zapisovať spomínané definície prostredníctvom formulárov. Tento prostriedok umožňuje vytvorenie ľubovoľnej sady konceptov, atribútov a pravidiel. Ukážka GUI rozhrania je na obr. 2 a 3. Príklad uvádza definíciu konceptov uvedenú v časti 3.3, kde obr. 2 znázorňuje rozhranie pre definíciu atribútu konceptu a obr. 3 rozhranie pre definíciu pravidla prispôbovania.

*Graph Author* umožňuje definíciu konceptov pomocou grafickej notácie. Cieľom je umožniť autorovi abstrakciu spôsobu definovania modelov (doménového aj modelu prispôbovania). *Graph Author* je založený na princípe preddefinovania štandardných vzťahov medzi konceptmi, ktoré sa najčastejšie vyskytujú pri tvorbe týchto modelov.



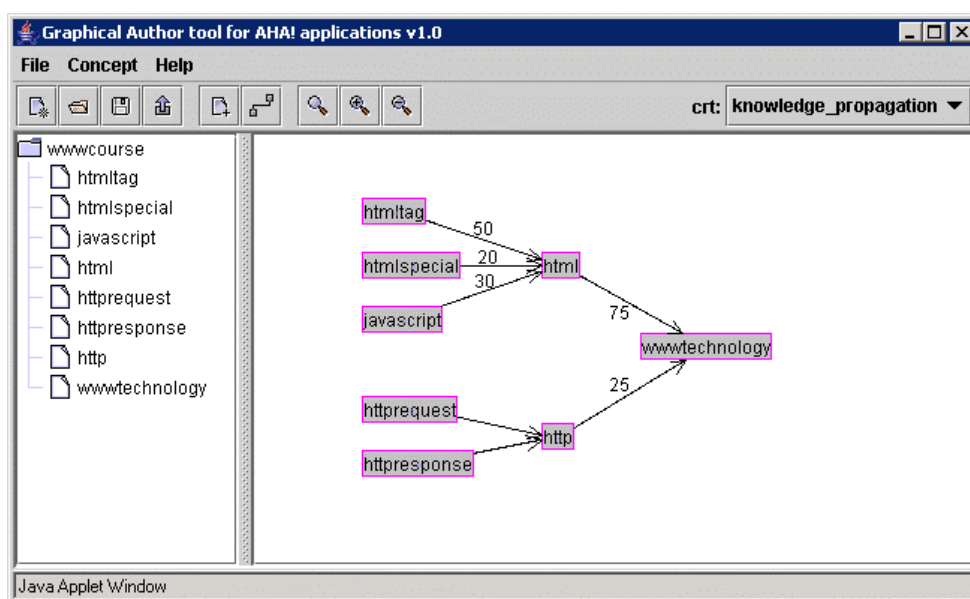
Obr. 2. AHA! Concept Editor.



Obr. 3. AHA! Concept Editor.



Klasickým príkladom je vzťah *prerekvizita*, ktorý určuje nevyhnutnosť znalosti jedného konceptu pre znalosť iného konceptu. Ďalšími príkladmi takýchto vzťahov sú napr. vzťahy *záujem*, *šírenie znalosti* a iné. Tieto vzťahy sa vo výsledných modeloch premietnu ako sada konkrétnych atribútov a pravidiel prispôsobovania. Ukážku definície grafu konceptov v tomto prostredí znázorňuje obr. 4 [11]. Príklad uvádza definíciu konceptov, medzi ktorými sú definované vzťahy typu *šírenie znalosti*. Uvedená definícia stanovuje, akým spôsobom vplýva znalosť čiastkových WWW technológií na celkovú znalosť tejto technológie. Napríklad, znalosť značiek jazyka HTML vplýva päťdesiatimi percentami na celkovú znalosť tohto jazyka, kým v prípade JavaScriptu je to iba 30%. Obdobne je to v prípade protokolu HTTP, kde však rovnakým spôsobom vplýva znalosť formátu požiadaviek na server ako aj formátu odpovedí zo servera. Podľa uvedeného príkladu na celkovú znalosť technológie WWW vplýva tromi dielmi znalosť jazyka HTML a jedným dielom znalosť protokolu HTTP.



Obr. 4. Graph Author.

Keďže sú stránky informačného priestoru reprezentované štandardným formátom XHTML, pri ich tvorbe je možné použiť ľubovoľný editor webových stránok. Aj z tohto dôvodu systém AHA! neposkytuje žiadny ďalší podporný prostriedok na ich definíciu.

#### 4 Záver

Adaptívne hypermédiá predstavujú v súčasnosti prirodzený posun v systémoch pre e-vzdelávanie. Výskum v tejto oblasti je úzko prepojený s výskumom v oblasti psychológie, keďže nástroje pre adaptívne hypermédiá majú podporiť proces učenia. Viac ako desaťročný výskum v tejto oblasti vyústil do stavu, kedy pôvodne ad-hoc prístupy v súčasnosti nahrádzajú prístupy na základe všeobecne akceptovaných metód a techník pre prispôsobovanie. Dôsledkom určitej „stabilizácie“ a nájdenia vhodných postupov sa prirodzene výskum posúva aj do oblasti metód a techník návrhu modelov a ich naplnenia.

Cieľom tohto príspevku bolo podať pohľad na stav v oblasti návrhu a naplňania modelov adaptívnych hypermediálnych systémov pre e-vzdelávanie prostredníctvom príkladu takéhoto systému – systému AHA!. AHA! vznikol na základe zovšeobecnenia skúseností s tvorbou adaptívnych učebníc. Zovšeobecnenie sa týka dvoch dimenzií – e-vzdelávania, t.j. možnosti

vytorenia ďalších adaptívnych učebníc a tiež adaptívnych systémov všeobecne, t.j. ambíciou AHA! je využitie nielen v adaptívnych systémoch pre vzdelávanie (podobne ako vznikali aj prázdne expertné systémy).

Súčasný systém len veľmi slabo podporujú vytváranie modelov na rôznych úrovniach abstrakcie. Vychádzajú z reprezentácie a umožňujú priamo alebo prostredníctvom formulárov naplniť jednotlivé štruktúry. Výskum v tejto oblasti je otvorený, priestor je najmä v skúmaní vhodných modelov a abstrakcií informačného priestoru a tiež v nových prístupoch k používateľskému rozhraniu, ktoré umožnia efektívne vytváranie a znovupoužitie materiálov pre výučbu.

*Táto práca bola čiastočne podporená grantom Slovenskej grantovej agentúry č. VGI/0162/03 „Kolaboratívne sprístupňovanie, analýza a prezentácia dokumentov v internetovom prostredí pomocou moderných softvérových nástrojov“.*

## Literatúra

1. Bieliková, M.: Adaptívna prezentácia hypermédií na webe. In *Proc. of DATAKON 2003*, L. Popelínský (Ed.), Brno, 2003, pp. 72-91.
2. Bieliková, M.: Využitie adaptívnych hypermédií pre e-vzdelávanie In *Proc. of Technologie pro e-vzdělávání*. B. Mannová, P. Šaloun (Eds.), Praha, 2003, pp. 15-21.
3. Bieliková, M., Habala, R.: Time-Based Extensions to Adaptation Techniques. In *Proc. of Int. Conference on Adaptive Hypermedia – AH 2004*. Springer Verlag, LNCS, Eindhoven, The Netherlands, August 2004, Accepted.
4. Brailsford, T.J.; Stewart, C.D.; Zakaria, M.R. & Moore, A.: Autonavagation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment. In *Proc. of 11th Int. World Wide Web Conference*, Honolulu, Hawaii, May 2002.
5. Brusilovsky, P., Schwarz, E., Weber, G.: A tool for developing adaptive electronic textbooks on WWW. In *Proc. of WebNet'96, World Conference on the Web Society*, H. Maurer (Ed.), San Francisco CA, AACE, October 1996, pp. 64-69.
6. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, Special issue on adaptive hypertext and hypermedia. Kluwer academic publishers, 2001, vol. 6, no. 2-3, 87-129.
7. Brusilovsky, P.: Developing adaptive educational hypermedia systems: From design models to authoring tools. In S. Blessing, T. Murray, S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments*. Kluwer, 2003.
8. De Bra, P., Houben, G.J., Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proc. of the ACM Conference on Hypertext and Hypermedia*, Darmstadt, Germany, 1999, pp.147-156.
9. De Bra, P., Aerts, A., Smits, D., Stash, N.: AHA! Version 2.0, More Adaptation Flexibility for Authors. In *Proc. of the AACE ELearn'2002 Conference*, October 2002, pp. 240-246.
10. De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N.: AHA! The Adaptive Hypermedia Architecture. In *Proc. of the ACM Hypertext Conference*, Nottingham, UK, August 2003, pp. 81-84.
11. Stash, N., De Bra, P.: Building Adaptive Presentations with AHA! 2.0. In *Proc. of the PEG Conference*, Saint Petersburg, Russia, June 2003.