

Adaptívna prezentácia hypermédií na webe

Mária BIELIKOVÁ

*Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
Ilkovičova 3, 812 19 Bratislava
bielik@elf.stuba.sk*

Abstrakt. Popularita adaptívnych hypermédií, špeciálne adaptívnej prezentácie informácií vo forme hypermediálnych aplikácií, v poslednom období výrazne stúpa aj vďaka rozširovaniu objemu informácií prístupných na webe. Príspevok sa sústreďuje na niektoré aspekty adaptívnych hypermédií najmä v súvislosti s ich prezentáciou. Analyzujú sa metódy a techniky prispôbovania na úrovni obsahu, prezentácie a navigácie. Berie sa pri tom do úvahy nielen model používateľa (napr. preferencie používateľa, úroveň znalostí, ciele), ale aj model prostredia (lokalizácia, jazyk, použité zariadenie, čas), v ktorom sa používateľ nachádza. Prispôbovanie prezentujeme na príklade aplikácie pre oblasť výučby. Pozornosť venujeme aj možnostiam modelovania takýchto systémov v UML. V príspevku sa riešia špecifické problémy a úlohy vyplývajúce z povahy prezentovaných médií.

Kľúčové slová: adaptívne hypermédiá, model navigácie, doménový model, model používateľa, model prostredia.

1 Úvod

Prístup k informáciám sa vďaka súčasným technológiám neustále rozširuje: sieť Internet poskytuje všeobecný prístup k informáciám; prostredníctvom celosvetovej siete WWW (angl. World Wide Web, ďalej len web) máme prístup k ľubovoľným informačným fragmentom prostredníctvom odkazov (v zmysle ohraničení podľa stanovených prístupových práv). Mnohé dnešné prezentácie na webe sú dynamické v tom zmysle, že umožňujú používateľovi aktívnu úlohu pri prezeraní, či štúdiu informačných zdrojov. Napriek významnému pokroku v tejto oblasti stále ostávajú viaceré problémy súvisiace s prezentáciou informácií. V tomto príspevku sa budeme zaoberať riešením týchto problémov:

- zablúdenie v informačnom priestore, dezorientácia (ide o pomerne bežný jav vzhľadom na skutočnosť, že web predstavuje otvorený priestor navzájom rôzne prepojených uzlov),
- kognitívne preťaženie (používateľ je zaplavený, preťažený a zmätený rôznymi možnosťami v navigácii),
- nesúvislý tok informácií (pri neriadenej navigácii sa používateľ často nevyhne návšteve rôznych, navzájom nie veľmi súvisiacich častí informačného priestoru, dôsledkom je napr. strata koncentrácie).

Lubos Popelínský (ed.), DATAKON 2003, Brno, 18.-21. 10. 2003, pp. 1-19.

Pri riešení uvedených problémov treba uvažovať so špecifickými potrebami každého jednotlivca. A tieto potreby sa navyše časom menia. Vlastnosťou dobrého webového sídla by teda mala byť taká prezentácia informácií, aby používateľ ľahko a prirodzene našiel požadované informácie podľa *aktuálnych potrieb*. Toto možno dosiahnuť adaptívnou prezentáciou informácií.

Hlavnou motiváciou pre prispôsobovanie v hypermediálnych systémoch je zvyšovanie efektívnosti práce s informáciami. Dôležitosť prispôsobovania narastá najmä v súvislosti s nárastom rozsahu informačných priestorov existujúcich hypermediálnych aplikácií. Pri rozsahu informačného priestoru neuvažujeme iba v pojmoch kvantitatívnych (počet riadkov textu, obrázkov a pod.), ale aj o možnostiach intelektuálneho zvládnutia obsahu, ktoré často priamo nezávisí od kvantity prezentovaného materiálu. Metódy a techniky adaptívnych hypermédií (AH) poskytujú prostriedky na dynamické prispôsobovanie prezentácie, obsahu alebo navigácie vzhľadom na aktuálny stav charakteristík používateľa alebo prostredia, v ktorom sa informácie poskytujú.

Výskum v oblasti adaptívnych hypermédií trvá relatívne krátko (cca 10 rokov, za prelomový rok sa považuje rok 1996, dokedy išlo iba o aktivity niekoľkých oddelených skupín). Prispôsobovanie sa v začiatkoch využívalo najmä v hypermediálnych systémoch určených pre vzdelávanie. Inteligentné systémy pre vzdelávanie (ITS, Intelligent Teaching Systems), vyvíjané využitím poznatkov z umelej inteligencie už pomerne dlhé obdobie, obsahujú model používateľa, na základe ktorého sa prispôsobuje obsah pre konkrétneho študenta, či skupinu študentov. Zmenou technológií a využitím hypermédií pre výučbové systémy vznikajú nové problémy (najmä v súvislosti s navigáciou pomocou odkazov), ale možno stavať na dosiahnutých výsledkoch v oblasti ITS. S rozvojom Internetu sa posúvajú aj možnosti aplikácií. Významnou aplikačnou oblasťou pre AH systémy sú napr. aj sídla rôznych predajcov, bánk, ktoré sa prispôsobovaním snažia čo najviac priblížiť k svojmu zákazníkovi. AH systémy sa začínajú využívať v systémoch pre vyhľadávanie informácií. Jedným z prvých systémov v tejto oblasti je WebWatcher [28]. Ďalšími príkladmi aplikácií adaptívnych hypermédií sú systémy na inzerovanie, adaptívne elektronické noviny, adaptívne zoznamy reštaurácií, adaptívne rozhranie pre prístup k informáciám prostredníctvom mobilných zariadení [5].

Základný prehľad systémov, metód a techník adaptívnych hypermédií možno nájsť v [6], neskôr doplnený v [7]. Ďalšie informačné zdroje možno nájsť najmä na stránke venovanej adaptívnym hypertextom a hypermediám, ktorú udržiava Technická univerzita v Eindhovene (<http://www.wis.win.tue.nl/ah>). Nachádzajú sa tu napr. zborníky z pracovných stretnutí (Workshops on Adaptive Hypertext and Hypermedia), ktoré sa uskutočňujú od roku 1997. V roku 2000 bola usporiadaná prvá medzinárodná konferencia AH 2000 venovaná adaptívnym hypermediám s plánom uskutočňovania medzinárodných AH konferencií každé dva roky.

V tomto príspevku sa sústredíme na niektoré aspekty adaptívnej prezentácie v hypermediálnych aplikáciách. Pod adaptívnou prezentáciou budeme rozumieť prispôsobovanie na úrovni prezentovaného obsahu, štýlu prezentácie a navigácie. Berie sa pri tom do úvahy nielen model používateľa (napr. preferencie používateľa, úroveň znalostí, ciele), ale aj model prostredia (lokalizácia, jazyk, použité zariadenie, čas), v ktorom sa používateľ nachádza. Otázky spojené s efektívnou prezentáciou a výberom rôznych typov médií sú mimo rozsah tohto príspevku.

Použitie AH systémov ukážeme na systéme ALEA (Adaptive LEArning) pre výučbu programovania pomocou príkladov. ALEA [26, 27] slúži na podporu výučby programovania pomocou riešených príkladov. ALEA obsahuje jednoduché príklady v programovacích jazykoch lisp a prolog spolu s návodmi na ich riešenie a riešeniami (riešenie nie je obmedzené uvedenými programovacími jazykmi, ale systém je naplnený takýmito príkladmi). ALEA poskytuje študentovi prostredie, v ktorom sa môže efektívne zdokonaľovať v programátorských zručnostiach. Na základe cieľov používateľa a sledovania práce mu odporúča príklady na riešenie. Pedagógovi zasa poskytuje spätnú väzbu pri výučbe napr. poznaním problematických príkladov.

2 Model adaptívneho hypermediálneho systému

Prispôsobovanie v hypermediálnom systéme sa môže diať na základe manuálneho zásahu používateľa (vyplnením formulárov, nastavením vlastností a pod.) alebo automaticky systémom (na základe daných charakteristík používateľa či prostredia a sledovaním činnosti používateľa). Prvý typ systémov sa označuje aj ako *adaptabilný* (angl. adaptable) a druhý ako *adaptívny* (angl. adaptive). Často sa využíva kombinácia oboch prístupov. Ak sa systém vyznačuje aspoň nejakými črtami automatického prispôsobovania, hovoríme o adaptívnych hypermediách.

Adaptívny hypermediálny (AH) systém je v prvom rade hypermediálny systém. To znamená, že pri jeho opise a návrhu treba vychádzať z poznatkov v oblasti tvorby hypermediálnych systémov a tieto rozšíriť o prispôsobovanie. Hypermediálne systémy zvyčajne zdieľajú určité prvky architektúry. Abstrakciu architektúry možno opísať referenčným modelom. Existuje niekoľko referenčných modelov hypermediálnych (hypertextových) systémov. Na špecifikáciu využívajú formálne (napr. Z, VDM, Petriho siete, UML-OCL), semiformálne (napr. diagramy), či neformálne (prirodzený jazyk) techniky opisu alebo ich kombinácie. Za prvý referenčný model sa považuje HAM (Hypertext Abstract Machine), ktorý bol navrhnutý a opísaný neformálne v roku 1988 [8].

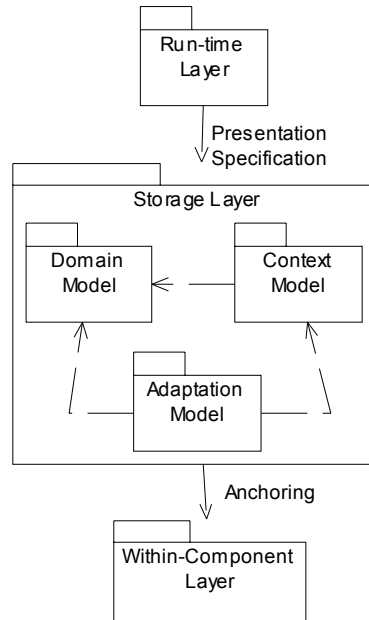
Najznámejším referenčným modelom hypermediálnych systémov je Dexterovský model (Dexter Hypertext Reference Model, nazvaný podľa miesta jeho vzniku – vytvorený na pracovnom stretnutí v Dexter Inn, Sunapee, New Hampshire [18, 19]). Cieľom bolo vytvoriť spoločný jazyk pre ľudí, ktorí sa angažujú vo vývoji hypermedií a vytvoriť abstraktný pohľad na hypermediá na základe existujúcich aplikácií. Existuje aj formalizácia tohto modelu v jazyku Z [18], Object-Z a UML-OCL [24].

Z Dexterovského modelu vychádzajú viaceré ďalšie modely, ktoré ho rôznym spôsobom rozširujú. Vychádzajú z neho aj dva známe modely adaptívnych hypermediálnych systémov: AHAM (Adaptive Hypermedia Architecture Model, [12]) a Mnichovský model AH systémov (Munich Reference Model for Adaptive Hypermedia Systems, [25]). AHAM opisuje AH systém najmä z pohľadu dát a Mnichovský model vychádza z objektovo-orientovaného prístupu a na opis využíva UML – vizuálnu reprezentáciu diagramami ako aj formálnu špecifikáciu v OCL (Object Constraint Language). Model AHAM bol navrhnutý najmä pre výučbové adaptívne hypermediá (súčasťou prvej verzie je model vyučovania spolu s pedagogickými pravidlami, ktorý bol neskôr zovšeobecnený na model prispôsobovania).

Dexterovský model definuje tri vrstvy (pozri obr. 1): *Run-Time Layer* (obsahuje opis prezentácie uzlov a odkazov; zodpovedá za interakciu s používateľom, získanie odozvy od používateľa a manažment sedení), *Storage Layer* (uchováva a sprístupňuje informácie o štruktúre hypermediálneho obsahu) a *Within-Component Layer* (obsahuje informačný obsah spolu so štruktúrou hypermediálnych uzlov). Dexterovský model formalizuje prvé dve vrstvy spolu s rozhraniami (*Presentation Specification*, *Anchoring*). Tretia vrstva je špecifická pre každú hypermediálnu aplikáciu.

Hlavným cieľom Dexterovského modelu je opis siete uzlov a odkazov medzi nimi, ktorý sa definuje v strednej vrstve. Súziami sa v tejto vrstve pracuje ako so všeobecnými skladmi dát. Stredná vrstva v Dexterovskom modeli predstavuje

- *model aplikačnej domény* (angl. domain model), ktorý opisuje štruktúru obsahu hypermediálneho systému a vzťahy medzi jednotlivými komponentami (v zmysle navigácie).



Obr. 1. Štruktúra AH systému.

V prípade AH systémov treba rozšíriť strednú vrstvu o (pozri obr. 1):

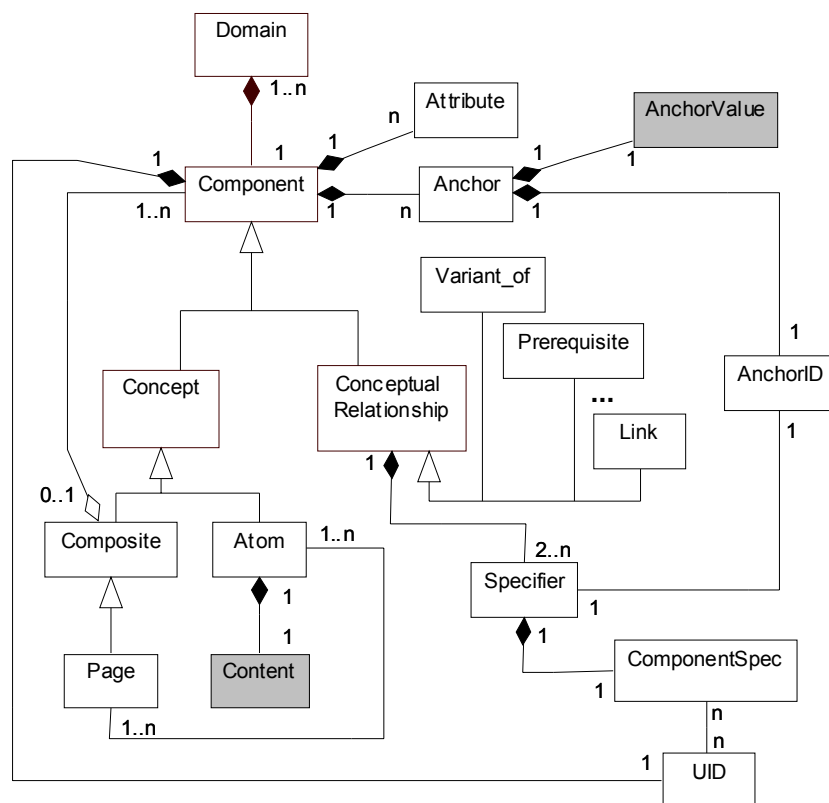
- model používateľa, resp. prostredia; používateľ, resp. prostredie predstavujú kontext prezentácie, preto budeme túto časť označovať ako *model kontextu* (angl. context model), ktorý predstavuje zdroj prispôsobovania a
- *model prispôsobovania* (angl. adaptation model), ktorý definuje prispôsobovanie systému.

Model prispôsobovania zahŕňa:

- *špecifikáciu prispôsobovania*, ktorou sú vyjadrené znalosti prispôsobovania (často s využitím pravidiel ako formalizmu reprezentácie znalostí) a
- *mechanizmus prispôsobovania*, ktorý predstavuje aktívny prvok systému zabezpečujúci prispôsobovanie. Mechanizmus prispôsobovania na základe špecifikácie prispôsobovania odvodzuje kotvenie odkazov (*anchoring*) a generuje špecifikáciu prezentácie (*presentation specification*).

2.1 Model aplikačnej domény

Model aplikačnej domény sa skladá z *konceptov*, ktoré reprezentujú ucelené časti prezentovanej oblasti a vzťahov medzi konceptami. Samotný obsah sa nachádza v informačnej báze (vo forme *informačných fragmentov*, ktoré sa zoskupujú do prezentateľných jednotiek – *stránok*). Oddelenie modelu aplikačnej domény od samotného obsahu (vrstva *Storage Layer* a vrstva *Within-Component Layer*) má výhodu v zjednodušení modelu. Napr. rôzne varianty obsahu tohto konceptu sa



Obr. 2. Meta-model aplikačnej domény (upravené z [25]).

reprezentujú oddelene v informačnej báze a teda možno využiť usudzovanie zvlášť pre výber konceptu a pre výber fragmentov.

Hierarchia *koncept – stránka – informačný fragment* zodpovedá referenčným modelom AH systémov [12, 25]. Na obr. 2 je časť meta-modelu aplikačnej domény, ktorý opisuje štruktúru modelu aplikačnej domény. Každý komponent (*Component*) má jednoznačný identifikátor (*UID*). Komponent môže byť koncept (*Concept*) alebo konceptuálna väzba (*Conceptual Relationship*). Koncept ďalej môže byť zložený (*Composite*) alebo elementárny (*Atom*). Konceptuálnymi väzbami modelujeme napr. odkazy v hypermediálnom systéme (*Link*) a ďalšie vzťahy (ako napr. prerekvizita). Konceptuálne väzby sa definujú nepriamo pomocou kotiev. Kotva (*Anchor*) pozostáva z dvoch častí identifikátor (*AnchorID*) a hodnota (*AnchorValue*). Hodnota kotvy, podobne ako informačný obsah elementárneho komponentu sú súčasťou vrstvy *Within-Component Layer*. K jednotlivým komponentom možno definovať premenlivý počet atribútov (trieda *Attribute*). Atribúty majú uplatnenie pri adaptívnej prezentácii, ale aj pri vyhľadávaní informácií, klasifikácii komponentov a pod. V [25] každý komponent povinne obsahuje aj špecifikáciu prezentácie komponentu. Táto sa však zvyčajne modeluje zvlášť v modeli prezentácie.

Ako možno vidieť z obr. 2, model podporuje oddelenú reprezentáciu konceptov, vzťahov medzi konceptami a informačných fragmentov (obsahu elementárnych

konceptov). Uvedený model neumožňuje explicitnú reprezentáciu variantov (na úrovni konceptov alebo na úrovni informačných fragmentov). Varianty možno reprezentovať pomocou vzťahu *VariantOf* na úrovni elementárnych konceptov, ale aj zložených konceptov. Formálny opis modelu sa definuje v jazyku OCL (OMG: <http://www.omg.org>).

Existujú viaceré ďalšie modely aplikačnej domény, ktoré sa odlišujú od načrtnutého prístupu najmä doplnením podrobností v súvislosti s použitými technológiami na realizáciu hypermediálneho systému a v tejto súvislosti aj rozdielnou terminológiou. Príkladom môže byť XAHM (XML-based Adaptive Hypermedia Model) [9], ktorý definuje tri abstraktné úrovne logickej štruktúry modelu aplikačnej domény:

1. informačné fragmenty alebo elementárne koncepty,
2. špecifikácie prezentácií (vychádzajú z koncepcie stránky, ktoré sa definujú ako jednotky prezentácie),
3. elementárne abstraktné koncepty, ktoré reprezentujú zoskupenia prezentačných jednotiek.

Aplikačná doména sa skladá z elementárnych abstraktných konceptov, ktoré sú organizované v digrafe (hrany reprezentujú vzťahy medzi elementárnymi abstraktnými konceptami, ktoré sú ohodnotené váhami vzhľadom na správanie používateľa). Model je reprezentovaný v XML, čo sa predpokladá aj o aplikáciách vytvorených na základe tohto modelu. Na opis meta-údajov sa využívajú „čisté“ XML dokumenty (namiesto napr. RDF a RDF Schema), čo zrejme spôsobí nemožnosť ďalšieho vývoja smerom k otvoreným systémom.

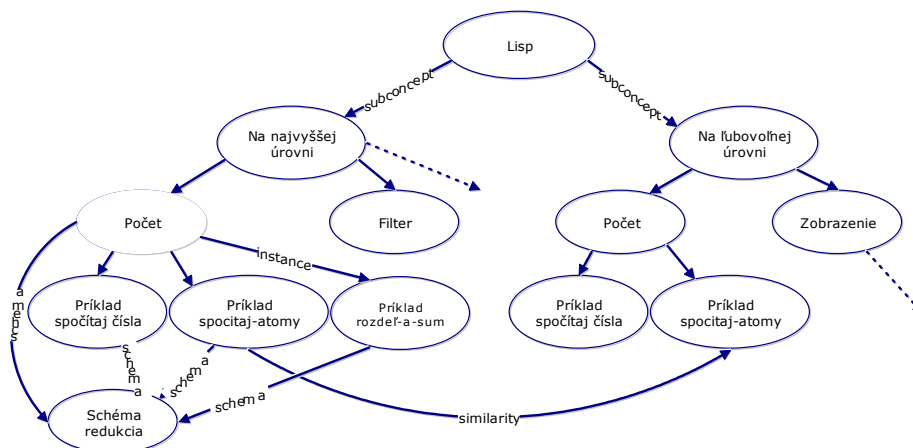
V systéme ALEA sú aplikačnou doménou znalosti z oblasti programovania reprezentované vo forme príkladov v programovacích jazykoch lisp a prolog. Model domény definuje tri typy konceptov organizované v hierarchickej štruktúre:

text	vysvetľujúce texty (ekvivalent textu v učebnici),
schema	šablóny riešenia určitých typov úloh, resp. zápisu riešenia v programovacom jazyku),
example	príklad programu; jeden príklad môže byť zviazaný s viacerými témami.

Dôležitú súčasť modelu aplikačnej domény predstavujú vzťahy medzi konceptami. V systéme ALEA sa používajú tieto typy relácií medzi konceptmi (meta-model ako aj realizácia systému ALEA umožňuje doplniť ďalšie relácie):

subconcept	označuje väzbu na bližšie nešpecifikovaný koncept, ktorý rozpracúva koncept vyššej úrovne,
schema	predstavuje väzbu na koncept, ktorý reprezentuje schému pre skupinu príkladov danú aktuálnym konceptom,
instance	reprezentuje väzbu na inštanciu – príklad,
prerequisite	väzba na koncept, ktorý je potrebné preštudovať pred daným konceptom,
similarity	väzba na koncept, ktorý je príbuzný aktuálnemu konceptu resp. je jeho analógiou v inej časti aplikačnej domény.

Koncepty typu `example` môžu mať priradený atribút `level`, ktorý definuje náročnosť daného príkladu. Atribút môže nadobúdať hodnoty 1, 2 alebo 3, pričom 1 znamená najľahší a 3 najnáročnejší príklad. Na obr. 3 sa nachádza príklad časti



Obr. 3. Príklad zobrazenia modelu aplikačnej domény ALEA ako grafu [26].

aplikačnej domény systému ALEA. Všetky modely sú v systéme ALEA zapísané v jazyku XML. Na špecifikáciu prezentácie konceptov sa využíva XSLT.

Dôležité je nielen vytvorenie modelu aplikačnej domény v zmysle jeho štruktúry, ale aj interpretácia definovaných konceptuálnych väzieb a funkcie umožňujúce prácu s modelom aplikačnej domény. Keďže reláciu prerekvizity (vyjadruje vhodnosť prezentácie konceptu v závislosti od toho, či a ako sa prezentovali iné koncepty) nájdeme vo väčšine AH systémoch, uvedieme si jednu možnosť jej interpretácie (prevzaté zo systému TIM – Time-Aware Adaptive Hypertext System [22]). Definuje sa parametrizovaná relácia *isPrerequisite*. Parameter určuje akciu, ktorú má AH systém vykonať v prípade, že ešte neboli prezentované všetky prerekvizity daného konceptu. Tento parameter môže mať štyri hodnoty (zoradené vzostupne podľa priority):

- *hide*: odporúčaná akcia je odkaz na koncept vôbec nezobraziť;
- *asText*: odporúčaná akcia je zobraziť odkaz iba ako neaktívny text;
- *dissuade*: odporúčaná akcia je zobraziť odkaz v štýle „neodporúčaný koncept“; týmto spôsobom máme možnosť zobraziť odkazovaný koncept, ale zároveň oznámiť používateľovi, že navštívenie tohto konceptu ešte nie je odporúčané;
- *recommended*: odporúčaná akcia je zobraziť odkaz na koncept v štýle „odporúčaný koncept“.

Ak napr. používateľ navštívil už všetky koncepty, ktoré sú prerekvizitami cieľového konceptu, AH systém zobrazí odkaz na koncept v štýle „odporúčaný koncept“. Ak ešte navštívené neboli, systém prejde všetky doteraz navštívené koncepty a určí parameter s najvyššou prioritou a podľa tohto parametra zvolí akciu.

Pri tvorbe modelu rozsiahlej aplikačnej domény je často problémom dosiahnuť úplnú explicitnú reprezentáciu vzťahov v modeli aplikačnej domény. Už pri stredne rozsiahlych systémoch (niekoľko desiatok konceptov) je veľmi náročné pre experta (tvorca informačného obsahu) explicitne definovať vzťahy medzi konceptami a podmienky pre prezentáciu príslušného informačného fragmentu, či prechodu na ďalší koncept. Preto sa využíva hierarchia konceptov (spolu s dedičnosťou), doplnená

o niekoľko ďalších vzťahov. Dôraz sa kladie na získanie a reprezentáciu znalostí o prispôbovaní v modeli prispôbovania.

2.2 Model kontextu

Základným zdrojom pre prispôbovanie sú charakteristiky používateľa a/alebo prostredia. Bez niektorých z uvedených charakteristík by sme o prispôbovaní nemohli hovoriť. Viaceré modely uvažujú iba jednu dimenziu prispôbovania – charakteristiky používateľa ([12, 25]), ktorú ďalej rozpracúvajú. XAHM [9] rozlišuje tri dimenzie prispôbovania: používateľ, externé prostredie a použité technológie. Rôzne dimenzie prispôbovania sa spoločne označujú ako *kontext* [1] a vyjadrujú sa modelom kontextu. Kontext reprezentuje stav, v závislosti od ktorého sa vykonávajú špecifické akcie (prechod na ďalší koncept, či zobrazenie obsahu konceptu určitým spôsobom).

Model kontextu a jeho rozpracovanie závisí v jednotlivých AH aplikáciách od aplikačnej domény. AH systémy pre vzdelávanie vyžadujú dobre rozpracovaný model používateľa najmä vzhľadom na znalosti a ciele používateľa-študenta. Adaptívny turistický sprievodca, ktorý nevyužíva informácie o polohe používateľa-turistu, zrejme nebude plniť očakávania. Vzhľadom na tieto skutočnosti nie je možné (a ani vhodné) definovať jednotný meta-model pre model kontextu.

Z pohľadu života AH systému rozpoznávame tri základné činnosti v súvislosti s modelom používateľa alebo prostredia:

1. získavanie údajov o používateli/prostredí,
2. spracovanie údajov a vytvorenie, či modifikácia modelu používateľa/prostredia a
3. použitie modelu pri prispôbovaní.

Poslednú etapu vždy vykonáva AH systém. Prvé dve sa realizujú v súčasných systémoch kombinovane (AH systém sleduje kroky používateľa, používateľ vyplní formulár a nastavuje vlastnosti a pod.).

Model používateľa

Typický adaptívny systém obsahuje model používateľa. Tento model reprezentuje rôzne charakteristiky používateľa ako napr. ciele, úlohy, úroveň vedomostí o prezentovaných informáciách, skúsenosti, preferencie. Zdrojom prispôbovania, najmä v oblasti výučbových AH systémov je využitie vlastností pamäti človeka a zabúdania [2].

Jednoduché AH systémy zahŕňajú iba charakteristiky používateľa súvisiace s aplikačnou doménou. Napr. model používateľa v AHAM [12] pozostáva z pomenovaných entít, ktoré uchovávajú dvojice atribút-hodnota. AH systém pre každého používateľa udržiava tabuľku, v ktorej sú pre každý koncept z aplikačnej domény uložené hodnoty stanovených atribútov. Takýmito atribútmi je napr. úroveň znalostí (typicky pre výučbové AH systémy), ale aj počet návštev alebo vhodnosť daného konceptu. V mnohých prípadoch je však okrem charakteristík používateľa súvisiacich s doménou užitočné modelovať *doménovo nezávislé* charakteristiky ako napr. všeobecné znalosti alebo kognitívne charakteristiky používateľa.

Model používateľa sa najčastejšie realizuje dvoma spôsobmi [6]: pomocou stereotypov alebo prekryvným modelom. *Stereotypy* predstavujú jednoduchšiu

metódu, ktorá neumožňuje personalizáciu (prispôsobovanie sa jednotlivým používateľom). Používatelia sa rozdelia do skupín a systém prispôsobuje obsah, jeho prezentáciu alebo navigáciu jednotlivým skupinám. Tento spôsob je výhodný najmä vtedy, ak nemáme dostatok informácií o jednotlivých používateľoch, teda na začiatku práce s AH systémom.

Prekryvný model umožňuje reprezentáciu doménovo orientovaných charakteristík používateľa. Predpokladá kópiu doménového modelu pre každého používateľa (pozri obr. 4, väzba triedy *DomainDependent* a triedy *Component* z modelu aplikačnej domény). Pre každý koncept aplikačnej domény AH systém získava a uchováva rôzne atribúty súvisiace s charakteristikami používateľa (napr. úroveň pochopenia daného konceptu, čas strávený štúdiom daného konceptu, počet návštev stránok prezentujúcich daný koncept). Prekryvný model môže reprezentovať znalosti vzťahujúce sa k úrovni pochopenia jednotlivých konceptov. Môže však reprezentovať aj odchýlky od očakávaného správania sa. Vtedy hovoríme o tzv. modeli porúch alebo chýb [30].

Zrejmé nároky na realizáciu prekryvného modelu vyvažuje možnosť prispôsobovania individuálne každému používateľovi. Často sa používa kombinácia oboch prístupov, keďže nevýhodu prekryvného modelu spočívajúcu v potrebe inicializácie modelu možno vyriešiť použitím stereotypov. Doménovo nezávislé charakteristiky používateľa sa reprezentujú nezávisle od modelu aplikačnej domény.

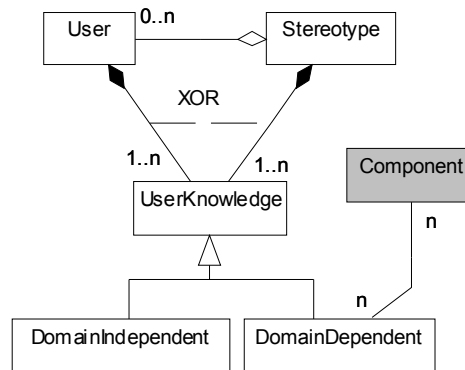
Vzhľadom na to, že model používateľa reprezentuje znalosti AH systému o používateľovi, na jeho realizáciu je vhodné využiť techniky pre reprezentáciu znalostí známe z umelej inteligencie vrátane neurčitosti. Podrobnejšie k modelovaniu používateľa možno nájsť v [23].

Systém ALEA využíva prekryvný model. Model používateľa sa uchováva v databáze, ktorá obsahuje iba údaje vzťahujúce sa k navštíveným konceptom a informačným fragmentom (príslušným používateľom). Zoznam navštívených fragmentov zahŕňa informácie o počte návštev stránky, ktorá zobrazuje fragment a dátum posledného prístupu. Zoznam navštívených konceptov zahŕňa odhadovanú úroveň znalostí používateľa o každom navštívenom koncepte. Uchovávajú sa tiež preferencie používateľa a jeho správanie (postupnosť akcií s doplnujúcimi údajmi, ktoré umožňujú rekonštrukciu práce používateľa so systémom).

Systém ALEA využíva prekryvný model. Model používateľa sa uchováva v databáze, ktorá obsahuje iba údaje vzťahujúce sa k navštíveným konceptom a informačným fragmentom (príslušným používateľom). Zoznam navštívených fragmentov zahŕňa informácie o počte návštev stránky, ktorá zobrazuje fragment a dátum posledného prístupu. Zoznam navštívených konceptov zahŕňa odhadovanú úroveň znalostí používateľa o každom navštívenom koncepte. Uchovávajú sa tiež preferencie používateľa a jeho správanie (postupnosť akcií s doplnujúcimi údajmi, ktoré umožňujú rekonštrukciu práce používateľa so systémom).

Model prostredia

Modelovanie prostredia nie je ešte bežné v súčasných AH systémoch. Prostredie predstavuje kontext, v ktorom používateľ pracuje (niekedy sa označuje aj ako model kontextu). Zahŕňa charakteristiky ako časové, priestorové umiestnenie, jazyk, sociálne aspekty, ale aj charakteristiky súvisiace s použitými technológiami ako napr. prehliadač, kvalita pripojenia, výstupné zariadenie používateľa.



Obr. 4. Časť meta-modelu používateľa.

Napr. v systéme turistického sprievodcu GUIDE [10] prostredie reprezentuje informácia o type zariadení, ktoré návštevník práve používa, o jeho polohe a o počasi (prší, fúka vietor a pod.). Model používateľa v tomto systéme obsahuje záujmy návštevníka, preferovaný jazyk a množinu atrakcií, ktoré už v danom dni (alebo inom časovom úseku návštevník videl). Model kontextu (charakteristiky používateľa a prostredia) sa využijú pri navigácii návštevníka po meste.

Niektoré špecifické údaje o kontexte sa reprezentujú v absolútnych alebo relatívnych hodnotách, napr. poloha (pomocou súradníc alebo „v blízkosti univerzity“) alebo čas (dátum alebo prvý deň tretieho týždňa semestra).

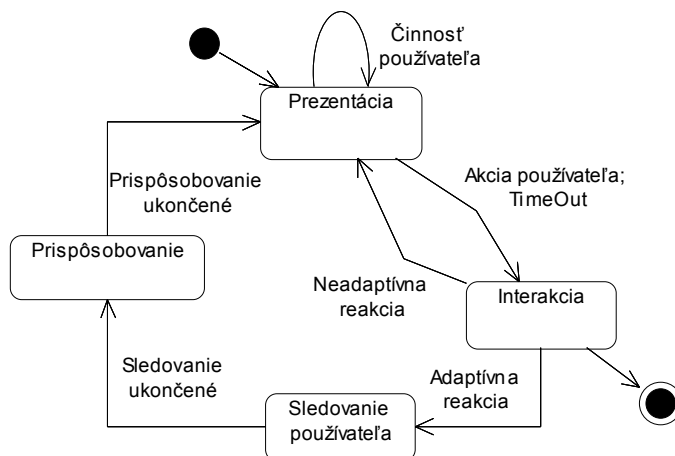
Vzhľadom na rozvoj mobilných technológií, zariadení s rozličnými možnosťami zobrazovania (PC, PDA, mobilný telefón) a tiež možností lokalizácie mobilných zariadení v blízkej budúcnosti sa zrejme podstatne viac rozšíria aplikácie AH systémov využívajúcich model prostredia pre adaptívnu prezentáciu informácií.

2.3 Model prispôsobovania

Prispôsobovanie predstavuje aktívny komponent AH systému. Riešime ním problémy načrtnuté v úvode. Napr. tým, že vytvoríme hierarchický obsah informačného priestoru a komentujeme, či inak označujeme v priebehu prezentácie jednotlivé položky znížime možnosť zablúdenia používateľa v informačnom priestore. Keď navyše používateľ bude vidieť, kde sa práve v informačnom priestore nachádza (prispôsobenie jeho aktuálnemu stavu), uvedená vlastnosť sa ešte vylepší.

Obr. 5 znázorňuje životný cyklus prispôsobovania. Stav *Prispôsobovanie* zahŕňa prispôsobovanie správania sa hypermediálneho systému a zmenu modelu kontextu (používateľa, prostredia). Zmena modelu kontextu sa môže vykonať pred samotným prispôsobením alebo po ňom.

Jednotlivé AH aplikácie sa odlišujú práve v spôsobe prispôsobovania. Model prispôsobovania reprezentuje znalosti a postupy ich spracovania (odvodzovania), ktoré určujú kvalitu adaptivity výslednej aplikácie. Z pohľadu znovupoužitelnosti a jednoduchosti údržby je v prípade tradičného prístupu „symbolickej“ umelej inteligencie výhodné oddeliť špecifikáciu prispôsobovania (znalosti o prispôsobovaní)



Obr. 5. Životný cyklus prispôsobovania.

a mechanismus prispôsobovania (interpretácia znalostí o prispôsobovaní). Oddelenie špecifikácie prispôsobovania umožní aj prispôsobovanie samotného systému definovaním meta- znalostí, ktoré budú modifikovať bázu znalostí o prispôsobovaní. Navyše sprístupnením znalostí tvorcovi obsahu adaptívneho systému možno znalosti o prispôsobovaní pre jednotlivé aplikácie jednoduchšie modifikovať (experta predstavuje tvorca obsahu).

Na reprezentáciu znalostí o prispôsobovaní sa zvyčajne používa pravidlový formalizmus (pozri napr. [3, 10, 11, 12, 22, 25, 26]). Pri známych cieľoch používateľa je vhodné spätné reťazenie. Častejšie však nájdeme AH systémy s priamym reťazením, kde výsledkom interpretácie pravidiel je zoznam akcií, ktoré sa vykonajú (napr. označenie vlastností fragmentov pre ich prezentáciu na stránke). Pravidlá definujú vzťah medzi modelom aplikačnej domény, modelom kontextu a špecifikáciou prezentácie.

Pri explicitnej reprezentácii znalostí o prispôsobovaní sa rozlišujú znalosti všeobecné pre celú aplikačnú doménu a znalosti vzťahujúce sa k špecifickým konceptom alebo skupinám konceptov. Príkladom všeobecného pravidla môže byť takéto pravidlo v systéme GUIDE [10]: Ak používateľ už navštívil nejaké miesto, neukazuj mu podrobnosti o ňom alebo pravidlo, ktorým sa v systéme ALEA zabezpečí nastavenie farby odkazu pre prečítané, ale ešte nepochopené uzly (prefix *var*: identifikuje premennú):

```

Rule ColorLink (var:Link, var:ConceptInstance)
  If
      var:ConceptInstance.Read = True
      var:ConceptInstance.Knowledge < 2
  Then
      var:Link.SetColor1 (um:Preferences.Read1)
      var:Link.SetColor2 (um:Preferences.Read2)
End Rule

```

V systéme ALEA sú znalosti o prispôsobovaní reprezentované sadami pravidiel (zapísanými v XML). Sú organizované do troch vrstiev [27]:

- výber stratégie učenia,
- výber konceptu
- výber fragmentov.

Modul špecifikácie prispôsobovania v systéme ALEA definuje dve stratégie učenia (v súvislosti s aplikačnou doménou učenia programovania pomocou príkladov): „od všeobecného ku konkrétnemu“ a „od konkrétneho k všeobecnému“. V prvom prípade sa najskôr prezentujú schémy riešenia typických úloh a tieto sa potom precvičujú na konkrétnych príkladoch. V druhom prípade sa riešením príkladov určitého typu študent dostáva k zovšeobecneniu pomocou programovej schémy. V systéme sa nachádzajú programové schémy k spracovaniu zoznamov v programovacích jazykoch lisp a prolog [4]. Mechanizmus prispôsobovania v systéme ALEA využíva priame reťazenie v rámci jednotlivých sád pravidiel.

Mechanizmus prispôsobovania pri odvodzovaní vychádza z obsahu modelu aplikačnej domény (povaha jednotlivých konceptov, vzťahy medzi nimi) a modelu kontextu (stav prezentácie z pohľadu používateľa alebo prostredia). Čím bohatšia reprezentácia týchto modelov, tým lepšie sú možnosti špecifikácie znalostí

prispôsobovania a samotného prispôsobovania. Pri výučbových systémoch je napr. výhodné doplniť model aplikačnej domény o tzv. mapu obsahu domény [21], ktorá umožňuje definovať implicitné väzby medzi konceptami a tieto využiť pri prezentácii.

Mapa obsahu domény určuje pre každý elementárny koncept index tohto konceptu množinou znalostných položiek, ktoré opisujú obsah tohto elementárneho konceptu. Mapa obsahu je definovaná ako zobrazenie [21]

$$I: H \rightarrow P(S),$$

keď S je množina znalostných položiek a H je množina elementárnych konceptov.

Takýto prístup zjednodušuje pridávanie ďalšieho obsahu do hypermediálneho systému, čím sa z uzavretého systému môže stať systém otvorený. Informačné fragmenty rôznych typov (schéma, príklad, text), ktoré sa vzťahujú k rovnakým znalostným položkám možno automaticky prepojiť a napr. odporúčať spoločne. Možno zisťovať aj relevantné koncepty, napr. v určitom stave sa vyberie koncept P_2 ako relevantný pre P_1 , keď aspoň jedna znalostná položka $I(P_1)$ sa tiež nachádza v $I(P_2)$. Toto možno ďalej rozvíť napr. tak, že dva informačné fragmenty pre ktoré $I(P_1) = I(P_2)$ obsahujú informáciu o tej istej oblasti a teda predstavujú „alternatívne pohľady“. Ďalej ak $|I(P_1) \cap I(P_2)| < |I(P_1)|$, P_2 vysvetľuje iba niektoré aspekty obsahu P_1 . Ak navyše $|I(P_1) \cap I(P_2)| < |I(P_2)|$, P_2 obsahuje ďalšiu informáciu, ktorá sa priamo nevzťahuje k P_1 , v prípade $|I(P_1) \cap I(P_2)| = |I(P_2)|$, P_2 opisuje niektoré aspekty P_1 bez ďalších podrobností.

Napriek tomu, že na špecifikáciu prispôsobovania väčšina existujúcich prístupov využíva pravidlový formalizmus a dokonca v známych referenčných modeloch hypermediálnych systémov zohľadňujúcich prispôsobovanie ([12, 25]) sa explicitne hovorí o pravidlách, na špecifikáciu prispôsobovania a následné prispôsobovanie možno využiť aj iné formalizmy pre reprezentáciu znalostí známe z umelej inteligencie ako aj prostriedky subsymbolickej umelej inteligencie [17].

3 Techniky prispôsobovania

Mechanizmus prispôsobovania interpretuje znalosti o prispôsobovaní tak, že sa použije niektorá z techník prispôsobovania. AH systémy rozlišujú viacero druhov prispôsobovania. Najviditeľnejšie je prispôsobovanie obsahu a jeho prezentácie. Prezentácia toho istého obsahu sa prispôsobuje vzhľadom na používateľove záujmy, znalosti alebo preferencie alebo charakteristiky prostredia, v ktorom používateľ študuje obsah. Druhou možnosťou je prispôsobenie obsahu samotného. Prispôsobovanie prezentácie je veľmi úzko zviazané s prispôsobovaním navigácie v obsahu. Pri adaptívnej navigácii AH systém môže odporúčať rôzne postupnosti pohybu v hyperpriestore.

3.1 Prispôsobovanie obsahu

Pri prispôsobovaní obsahu sa využíva metóda modifikácie obsahu, ktorej podstatu tvorí zobrazenie iba relevantného obsahu alebo metóda variantov obsahu. Na prispôsobenie textového obsahu sa používajú najmä tieto techniky:

- *vkładanie/odstraňovanie fragmentov*: informačné fragmenty, ktoré nie sú „vhodné“ sa nezobrazujú. Napr. pri použití stratégie „od konkrétnemu ku všeobecnému“ sa na začiatku nezobrazujú programové schémy;

- *rozíhovací text (strečtext)*: zobrazovanie textu v skrátenej a aj v plnej (rozťahnutej) podobe, pričom formu zobrazenia jednotlivých fragmentov určuje AH systém.

Táto a aj vyššie uvedená technika sa realizuje najčastejšie pomocou podmieneného textu;

- *alternatívy fragmentov*: existujú viaceré varianty textu fragmentov alebo varianty prezentácie fragmentov, prezentuje sa „vhodný“ variant. Táto technika je vhodná napr. pri prezentácii textu v učebnici v rôznych úrovniach obtiažnosti pre študentov s rôznou úrovňou vedomostí;
- *usporadúvanie fragmentov*: prezentované fragmenty sa usporiadajú podľa „vhodnosti“. Napr. pri prezentácii príkladu v systéme ALEA sú fragmenty zobrazené v poradí príklad, pomôcka a riešenie.

3.2 Prispôsobovanie prezentácie

Prispôsobovanie prezentácie spočíva v zmene prezentácie obsahu použitím rôznych prezentačných techník. Tieto zmeny sa často uskutočňujú spolu so zmenami obsahu. Zaradujeme sem aj viacjazyčnú prezentáciu určitého obsahu (hoci jazykové varianty by sme mohli považovať aj za varianty obsahu, najmä vtedy, keď ich obsah nie je ekvivalentný). Používajú sa tie isté techniky ako pri prispôsobovaní obsahu okrem techniky strečtext. Navyše sa používajú tieto techniky:

- *alternatívy štýlov*: existujú viaceré varianty štýlov zobrazovania obsahu (napr. farba pozadia, typ a veľkosť písma a pod.);
- *anotácia fragmentov*: AH systém označuje „vhodné“ fragmenty na stránke (napr. farebným odlišením písma alebo pozadia). Táto technika sa dá využiť aj na *zahmlievanie fragmentov*, t.j. fragmenty sa objavia v prezentácii, ale sú označené tak, že nie sú „vhodné“ na štúdium a teda ich prezentácia je nevýrazná. Táto technika je vhodná najmä v prípade, keď sa prispôsobovanie deje aj vzhľadom na čas [22].

Alternatívy štýlov umožňujú prispôsobiť celú prezentáciu, ale aj jednotlivé časti. Na prispôsobenie prezentácie častí informačných fragmentov (anotácia fragmentov) sa najčastejšie využíva farebné odlišenie textu alebo pozadia. Možno však využiť aj zmenu veľkosti písma (zoom) a tak, zdôrazniť niektoré časti a stále zachovať prehľad o rozsahu informačného obsahu [29].

3.3 Prispôsobovanie navigácie

Prispôsobenie navigácie je z pohľadu podstaty hypermediálneho systému veľmi dôležité. Prispôsobovanie spočíva v ovplyvňovaní cesty používateľa v informačnom priestore. Toto ovplyvňovanie môže byť direktívne v tom zmysle, že systém znemožní cesty, ktoré nie sú pre používateľa v danom kontexte „vhodné“, kedy rôznymi prostriedkami používateľského rozhrania systém prezentuje používateľovi odporúčané (alebo neodporúčané) cesty v informačnom priestore. Cieľom prispôsobovania navigácie je zabezpečenie, aby nasledujúca informácia bola pre používateľa zaujímavá, či užitočná. Využívajú sa pri tom rôzne metódy, napr. globálne/lokálne navádzanie používateľa (vzhľadom na určený cieľ), globálne/lokálne orientovanie používateľa alebo personalizované pohľady na hyperpriestor.

Na realizáciu uvedených metód navigácie v informačnom obsahu sa používajú najmä tieto techniky:

- *priame vedenie*: AH systém vedie používateľa v informačnom priestore, t.j. vyberá najvhodnejšie koncepty a fragmenty im priradené. Realizuje sa pomocou tlačidla „Ďalej“. V systéme ALEA tlačítko „»“ (pozri obr. 6, vpravo hore);
- *usporadúvanie odkazov*: odkazy na ďalšie koncepty sa usporiadajú podľa vhodnosti. V systéme ALEA sa zoznamy zoskupujú (pozri obr. 6, tri skupiny odkazov v ľavej časti);
- *anotácia odkazov*: AH systém označuje „vhodné“ odkazy. Napr. systém ALEA farebne rozlišuje navštívené, nenavštívené a pochopené koncepty (pozri obr. 6);
- *skrývanie odkazov*: odkazy, ktoré vedú k neodporúčaným informáciám sa skrývajú. Skrývanie možno realizovať niekoľkými formami: odkaz sa nezobrazí (zobrazí sa iba text odkazu), odkaz sa blokuje (spôsob prezentácie závisí od kombinácie nezobrazenia odkazu a anotácie odkazu) alebo odkaz sa zruší z prezentácie;
- *pasívna navigácia*: implicitné odkazy, ktoré využíva systém po rozpoznaní určitého vzoru správania sa používateľa; používateľ je neaktívny a navigáciu zabezpečuje systém (možno využiť aj pre tlačítka „späť“ a „ďalej“);
- *generovanie odkazov*: AH systém dynamicky generuje nové odkazy (napr. objavuje súvislosti medzi jednotlivými konceptami);
- *adaptácia máp*: AH systém na základe modelu používateľa a/alebo modelu prostredia dynamicky vytvára mapu domény (grafická prezentácia navigácie).

The screenshot shows the ALEA system interface for editing a logic schema. The top navigation bar includes 'Odhlásenie', 'Obsah', 'História', 'Nastavenia', 'Zmena hesla', and 'Záložky'. The main content area is titled 'Schéma Zobraz' and contains a logic schema definition:

```

Schéma:
<zobraz>([ ], [ ]).
<zobraz>([H1 | T1], [H2 | T2]) :-
  <transformuj>(H1, H2),
  <zobraz>(T1, T2).

```

Below the schema, there is a 'Popis' section with the text: 'Schéma Zobraz je podrobne vysvetlená v učebnici [Bieliková Mária, Návrat Pavol: Funkcionálne a logické programovanie].'

The interface also features a 'Komentáre' section with a table of comments:

Meno	Dátum :..	Text
Rado Kostelník	11.11.2002 22:01:52	Tato schema je jednoduchá...

At the bottom, there is a 'Zobraz všetky' button and a 'Pridaj' button.

Obr. 6. Prispôsobovanie odkazov v systéme ALEA.

4 Návrh adaptívnych hypermédií

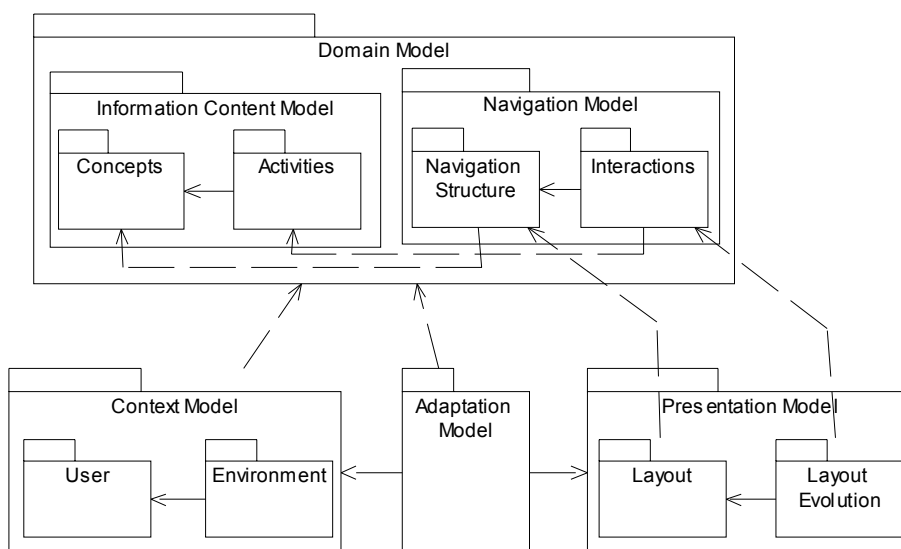
V porovnaní so štandardnými softvérovými aplikáciami sa návrh adaptívnych hypermediálnych systémov odlišuje najmä v tom, že sa sústreďuje na modely vlastné hypermediálnym systémom [20, 15] a ďalšie modely, ktoré sú špecifické pre adaptívne hypermediálne systémy [14]. Ide o tieto modely (obr. 7):

- model informačného obsahu
- model navigácie
- model prezentácie
- model kontextu (používateľ a/alebo prostredie)
- model prispôsobovania.

Modely vyjadrujú rôzne pohľady na AH systém. Na ich opis sa využívajú techniky a prostriedky modelovania softvérových systémov s prípadným prispôbením (najmä špecializácia) pre hypermediálne, resp. AH systémy (napr. v [13] sa definujú rozšírenia UML potrebné pre modelovania navigácie).

Pri modelovaní treba brať do úvahy statické aj dynamické aspekty. Väčšina existujúcich prístupov sa zameriava iba na modelovanie statických aspektov (využitím diagramov tried) a dynamickým aspektom (modelovanie správania) sa vôbec nevenuje. Pričom dynamické aspekty sú dôležité najmä pri návrhu navigácie v AH systémoch. Na modelovanie dynamických aspektov jednotlivých modelov (činnosti a interakcie v modeli aplikačnej domény a vývoj štruktúry prezentácie v modeli prezentácie – pozri obr. 7) sa používajú najmä stavové diagramy a diagramy činnosti.

Prehľad existujúcich prístupov k modelovaniu hypermediálnych systémov možno nájsť v [15]. Zo súčasného stavu vidieť prechod od rôznych techník modelovania (entitno-relačné diagramy, Petriho siete, CRC Cards a ďalšie špeciálne typy semiformálnych a formálnych opisov) k UML.



Obr. 7. Závislosti medzi modelmi AH systému.

K vyššie uvedeným modelom pri vývoji AH systému pribúda model prípadov použitia, ktorý definuje základné požiadavky vo forme prípadov použitia na hypermediálny systém. Prípady použitia zahŕňajú a požiadavky na prispôsobovanie. Treba poznamenať, že oblasť metodiky vývoja softvérových systémov na webe je v začiatkoch. Sú známe viaceré metódy vývoja takýchto systémov (napr. [20]), nemožno však ešte hovoriť o akceptovaní týchto metód.

5 Záver

V súčasnosti existuje viacero systémov, ktoré podporujú prispôsobovanie používateľovi a/alebo prostrediu, vytvorených v prostredí webu. Z pohľadu použitých technológií pri návrhu a realizácii AH systémov na webe môžeme sa tieto aplikácie vyvíjajú v súlade s trendmi pre distribuované prostredie Internetu všeobecne. Na reprezentáciu dátových štruktúr sa využívajú XML štandardy v kombinácii s relačnými databázami. Prvé AH systémy na webe realizovali špecifikáciu prispôsobovania priamo v informačnom obsahu ako komentáre v HTML dokumente [11].

Vývoj AH systémov na webe umožňuje prechod od uzavretých systémov, ktoré na prezentáciu informácií využívajú prostredie Internetu pre vzdialený prístup k AH systému (tenký klient v podobe prehliadača), k systémom, ktoré využívajú informačné zdroje okolia a stávajú sa súčasťou distribuovaného softvérového systému (realizované napr. ako webová služba). Otvorené systémy umožnia zdieľanie meta-údajov pre rôzne adaptívne hypermédiá v rovnakých oblastiach [16]. Môže to mať významný vplyv napr. na oblasť vzdelávania, kde sa rola tvorcu AH učebnice posunie z tvorby samotného informačného obsahu k návrhu vhodných postupností prezentácie, úrovne podrobností a pod.

Prispôsobovanie sa deje na základe sémantiky, ktorú do hypermediálneho systému vkladá jeho autor. Súčasný AH systémy sú väčšinou uzavreté (hoci realizované v otvorenom priestore webu) práve preto, lebo meta-údaje využívané pri prispôsobovaní sú definované a spracúvané v každom prípade inak. Výskum a vývoj v oblasti pridávania sémantiky do webu (iniciatívy označované *Semantic Web*) sú výzvou aj pre oblasť otvorených AH systémov.

Prispôsobovanie nemusí mať len pozitívny vplyv na prezentáciu rôznych materiálov. Intenzívne sa diskutuje napr. vnímanie zmien prezentácie a kontextových odkazov používateľmi. Niektorí používatelia odmietajú priame vedenie AH systémom a vyžadujú prostriedky na podporu orientácie v informačnom priestore (ideálne nemenné zobrazenie štruktúry alebo mapy informačného priestoru). Treba si však uvedomiť, že zväčšovaním informačného priestoru a jeho dynamickými zmenami je pochopenie celej štruktúry informačného priestoru aj tak nemožné.

Riziká AH systémov spočívajú najmä v skutočnosti, že časť riadenia získavania informácií sa presúva z človeka na systém. Hypermediálny systém na webe štandardne umožňuje voľnú navigáciu v informačnom priestore. AH systém ju spravidla určitým spôsobom ohraničuje, čím ju prispôbuje konkrétnemu používateľovi, skupine používateľov, či prostrediu. Preto môže takéto rozhranie človek-počítač pôsobiť na človeka deprimujúco, najmä z hľadiska obmedzení, ktoré zavádza a nepredvídateľnosti jeho správania. Ako príklad možno uviesť problém

s tlačítkom „spät“. Ak sa používateľ chce vrátiť späť k nejakej informácii a podľa zvyku využije tlačítko „spät“, často sa stane, že prezentácia, ktorá sa zobrazí je iná ako tá, ktorá sa zobrazila predtým (keďže sa stav modelu používateľa zmenil). Preto je dôležité, aby používateľ poznal takéto črty AH systému, resp. aby mal príležitosť overiť si spôsob práce systému a tak si vybudovať dôveru k nemu.

Na záver spomenieme etický rozmer AH systémov. Automatické prispôsobovanie je možné iba vtedy, ak má systém dostatok informácií o používateli a jeho prostredí (s predpokladom, že znalosti o prispôsobovaní a spôsob usudzovania v AH systéme spĺňajú kritériá kladené na AH systém). Ťažko môžeme odporučiť niekomu dobrú reštauráciu, ak nevieme, kde sa nachádza. Ak má AH systém odporučiť študentovi vhodné príklady na precvičenie v rámci obmedzeného času, ktorý má študent k dispozícii a chce ho využiť čo najlepšie, je nevyhnutné poznať ako sa mu predtým darilo, koľko času strávil riešením predchádzajúcich úloh a pod. Poskytnutie týchto informácií – nevyhnutných pre prispôsobovanie – však môže zasiahnuť do nášho súkromia. Budeme chcieť byť takto sledovaní? Čas ukáže, či výhody technického a technologického pokroku prevýšia prípadnú stratu súkromia.

Táto práca bola čiastočne podporená grantom Slovenskej grantovej agentúry č. VG1/0162/03 „Kolaboratívne sprístupňovanie, analýza a prezentácia dokumentov v internetovom prostredí pomocou moderných softvérových nástrojov“.

Literatúra

1. Bieliková, M.: Adaptive Presentation of Evolving Information Using XML. In: *Proc. of IEEE Int. Conf. on Advanced Learning Technologies*, T. Okamoto, R. Hartley, Kinshuk, J.P. Klus (Eds.), IEEE Press, Madison, Wisconsin (2001) 193-196.
2. Ágh, P.: *Zvyšovanie efektívnosti procesu učenia pomocou adaptívnych hypermédií*. Diplomová práca, M. Bieliková (supervisor). STU Bratislava (2001).
3. Bieliková, M., Pilka, M.: User Interface Adaptation through Data Mining. In: *Proc. of DATAKON 2001*, M. Bieliková (Ed.), Brno (2001), 310-318.
4. Bieliková, M., Návrat, P.: Use of Program Schemata in Lisp Programming: An Evaluation of its Impact on Learning. *Informatika* 9:1 (1998) 5-20.
5. Billsus, D., Brunk, C.A., Evans, C., Gladish, B., Pazzani, M.: Adaptive Interfaces for Ubiquitous Web Access. *Communications of the ACM* 45:5 (2002) 34-38.
6. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, Kluwer academic publishers 6:2-3 (2001) 87-129.
7. Brusilovsky, P.: Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, Kluwer academic publishers 11:1-2 (2001) 87-110.
8. Campbell B., Goodman J.: HAM: A General Purpose Hypertext Abstract Machine. *Communications of the ACM*, 31:7 (1988) 856-861.
9. Cannataro, M., Pugliese, A.: XAHM: an XML-based Adaptive Hypermedia Model and its Implementation. In: *Proc. of 3rd Workshop on Adaptive Hypertext and Hypermedia*, Denmark (2001). Available at <http://www.wis.win.tue.nl/ah2001>

10. Cheverst, K., Davies, N., Mitchel, K., Smith, P.: Providing Tailored (Context-Aware) Information to City Visitors. In: *Proc. of Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*. P. Brusilovsky, O. Stock, C. Strapparava (Eds.). Springer Verlag, LNCS 1892, Trento, Italy (2000) 73-85.
11. De Bra, P., Calvi, L.: AHA: a Generic Adaptive Hypermedia System. In: *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia*. Pittsburg, USA (1998) 5-12. Available at <http://www.wis.win.tue.nl/ah98>
12. De Bra, P., Houben, G.J., Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In: *Proc. of the ACM Conf. on Hypertext and Hypermedia*, Darmstadt, Germany (1999) 147-156.
13. Dolog, P., Bieliková, M.: Modelling Browsing Semantics in Hypertexts using UML. In: *Proc. of ISM 2001 - Information Systems Modelling*. J. Zendulka (Ed.). MARQ Ostrava (2001) 181-188.
14. Dolog, P., Bieliková, M.: An Approach to Designing Adaptive Hypermedia. In: *Proc. of Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*. P. De Bra, P. Brusilovsky, R. Conejo (Eds.). Springer Verlag, LNCS 2347, Malaga Spain (2002) 586-591.
15. Dolog, P., Bieliková, M.: Hypermedia Systems Modelling Framework. *Computers and Informatics* 21:3 (2002) 221-239.
16. Dolog, P., Gavrioloaie, R., Nejd, W., Brase, J.: Integrating Adaptive Hypermedia Techniques and Open RDF-Based Environments. In: *Proc. of WWW2003*, Budapest, Hungary, ACM Press (2003).
17. Encarnação, L. M., Stoev, S. L.: An Application-Independent Intelligent User Support System Exploiting Action-Sequence based User Modeling. In: *Proc. of 7th Int. Conf. on User Modeling*. Wien, Springer (1999) 245-254.
18. Halasz, F., Schwartz, M.: The Dexter Hypertext Reference Model. In: *Proc. of NIST Hypertext Standardization Workshop* (1990) 95-133.
19. Halasz, F., Schwartz, M.: The Dexter Hypertext Reference Model. *Communications of the ACM* 37:2 (1994) 30-39.
20. Hennicker, R., Koch, N.: Systematic Design of Web Applications with UML. In: *Unified Modeling Language: Systems Analysis, Design and Development Issues*. K. Siau, T. Halpin (Eds.). Idea Group Publishing (2001) 1-20.
21. Henze, N., Nejd, W.: Extensible Adaptive Hypermedia Courseware: Integrating Different Courses and Web Material. In: *Proc. of Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*. P. Brusilovsky, O. Stock, C. Strapparava (Eds.). Springer Verlag, LNCS 1892, Trento, Italy (2000) 109-120.
22. Habala, R.: *Časovo ohraničená prezentácia informácií*. Diplomová práca, M. Bieliková (supervisor). STU Bratislava (2003).
23. Kobsa, A.: User Modeling: Recent Work, Prospects and Hazards. In: *Adaptive User Interfaces: Principles and Practice*, M. Schneider-Hufschmidt, T. Kühme, U. Malinowski (Eds.) Elsevier (1993).
24. Koch N.: *UML+OCL Specification of the Dexter Hypertext Reference Model*. Ludwig-Maximilians-Univ. Munich, Inst. of Computer Sci., Tech. Rep. (2000).
25. Koch, N., Wirsing, M.: The Munich Reference Model for Adaptive Hypermedia Applications. In: *Proc. of Int. Conf. on Adaptive Hypermedia and Adaptive Web-*

- Based Systems*. P. De Bra, P. Brusilovsky, R. Conejo (Eds.). Springer Verlag, LNCS 2347, Malaga Spain (2002) 213-222.
26. Kostelník, R.: *Podpora výučby programovania pomocou príkladov v pavučine*. Diplomová práca, M. Bieliková (supervisor). STU Bratislava (2002).
 27. Kostelník, R., Bieliková, M.: Web-Based Environment using Adapted Sequences of Programming Exercises. In: *Proc. of Information Systems Implementation and Modelling*. M. Beneš (Ed.). MARQ Ostrava, Brno, Czech Republic (2003) 33-40.
 28. Mladenic, D.: *Personal WebWatcher – Implementation and Design*. Tech. Report IJS-DP-7472, Dept. of Computer Science, J. Stefan Inst. (1996). Available at <http://www-ai.ijs.si/DunjaMladenic/pww.html>
 29. Tsandilas, T., schraefel, m.c.: Adaptive Presentation Supporting Focus and Context. In: *Proc. of Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems* (2003). Available at <http://www.wis.win.tue.nl/ah2003>
 30. Vassileva, J.: A Three-dimensional Perspective on the Current Trends in Student Modeling. In: *Proc. of East-west Conf. on Emerging Technologies in Education* (1992) 315-320.

Annotation:*Presentation of Adaptive Hypermedia on the Web*

Adaptive hypermedia and especially adaptive presentation of information represented using hypermedia are becoming very popular mainly due to the rapid expanse of available information on the web. We focus on several aspects of adaptive hypermedia related to information presentation on the web environment. We analyse methods, techniques and tools of adaptive content, adaptive presentation and adaptive navigation support. Adaptation decisions are based on taking into account not only various user characteristics represented by a user model (e.g., user preferences, knowledge level, goals, user background) but also environment characteristics (user locality, equipment used, language, time). An application in the domain of educational adaptive systems is presented. We briefly discuss also modelling issues of adaptive web systems expressed in the UML. Specific problems and tasks narrowed to the hypermedia as such are not discussed, the paper is oriented mainly on the text presentation.