

# Udržiavanie charakteristík v modeli používateľa kladením otázok využívajúcich koncepty doménovej ontológie

Tomáš Klempa, Anton Andrejko, and Mária Bieliková

Ústav informatiky a softvérového inžinierstva  
Fakulta informatiky a informačných technológií  
Slovenská technická univerzita v Bratislave  
Ilkovičova 3, 842 16 Bratislava, Slovensko  
tomas.k@zmail.sk, {andrejko,bielik}@fiit.stuba.sk

**Abstrakt** *V adaptívnych webových aplikáciách je proces personalizácie založený na modeli používateľa, ktorý je tvorený charakteristikami používateľa. Aby bolo prispôsobovanie efektívne, je potrebné, aby model používateľa vždy odrážal reálneho používateľa. Preto je potrebné model používateľa neustále aktualizovať. V príspevku opisujeme metódu explicitného získavania a udržiavania charakteristík v modeli používateľa kladením otázok v prirodzenom jazyku. Kontext otázky závisí na doménovej ontológii, ktorej koncepty sú viazané prostredníctvom internej reprezentácie. Celý proces získavania a udržiavania charakteristík je riadený používateľom definovanými pravidlami. Výsledkom navrhnutých metód je softvérový prototyp.*

## 1 Úvod

Web predstavuje rozsiahly zdroj informácií z mnohých oblastí, pričom množstvo informácií neustále rastie. Informačný obsah webu nie je vždy vhodne rozmiestnený, čo spôsobuje problémy s navigáciou, prípadne sa používateľ brodí obsahom, o ktorý nemá záujem a pod. Spôsob, ako zefektívniť prácu s informáciami ponúka personalizácia, kde sa vybrané črty používateľa reprezentujú v modeli používateľa a použijú sa na prispôbienie rôznych viditeľných aspektov systému.

Model používateľa uchováva abstrahované charakteristiky používateľa. Charakteristiky používateľa je možné získavať explicitným alebo implicitným spôsobom. Explicitný spôsob je založený na priamej odozve od používateľa a implicitný spôsob využíva sledovanie aktivity používateľa. Nevýhodou explicitného spôsobu je, že môže vyrušovať používateľa pri práci. Naopak získané informácie sú spravidla presnejšie ako pri implicitnom spôsobe, kedy je nevyhnuté sledované charakteristiky správne interpretovať, aby mohli byť uložené v modeli používateľa. Preto sa obidva prístupy často kombinujú.

Väčšina webových adaptívnych aplikácií kladie dôraz najmä na prispôbovanie viditeľných aspektov systému. Charakteristiky v modeli používateľa sa, podobne ako aj používateľ, v čase menia. Dôsledkom neudržiavania charakteristík je neefektívne prispôbo-

vanie, keďže model neodráža aktuálne charakteristiky reálneho používateľa.

Spôsob aktualizácie modelu používateľa založený na sledovaní aktivity používateľa je opísaný v [9]. Využíva informáciu o návšteve konceptu, ako aj kombináciu počtu návštev konceptu s časom stráveným v koncepte, čím eliminuje možnosť chyby vzniknutej na základe nesprávnych uzáverov ohľadom času zobrazenia príslušného konceptu. Získavanie charakteristík procesom učenia využíva systém UPRE [7].

Explicitnú spätnú väzbu využíva väčšina adaptívnych výučbových aplikácií na zistenie vedomostí používateľa o zobrazenom koncepte [2,5], čo sa prejaví aj v modeli používateľa, keďže vo výučbových systémoch model používateľa spravidla prekrýva model aplikačnej domény.

V príspevku opisujeme metódu inicializácie a udržiavania charakteristík v modeli používateľa, ktorá používa explicitný spôsob – odpovede na otázky. Motiváciou pre tento spôsob bol projekt M-PIRO, ktorý využíva sémantiku poskytovanú ontologickou reprezentáciou na generovanie textu v prirodzenom jazyku [1]. Nástroj vytvorený v rámci projektu generuje texty z ontológie, ktorá poskytuje informácie o entitách v doméne. V [8] je opísaný prístup plánovania výrokov v prirodzenom jazyku, ktoré môžu dynamicky využiť kontextové informácie o prostredí, v ktorom prebieha dialóg. Iným príkladom je vzdelávací systém OntoAIMS [6], kde používateľ komunikuje so systémom prostredníctvom grafického rozhrania, ktoré pozostáva z dialógu (s vopred pripravenými otázkami) a grafu, ktorým možno vytvárať a modifikovať vyhlásenia grafickým spôsobom.

V navrhnutých metóde využívame model aplikačnej domény reprezentovaný ontológiou a jeho koncepty na vygenerovanie otázok v prirodzenom (anglickom) jazyku s cieľom aktualizovať charakteristiky v modeli používateľa. Odpovede na tieto otázky sa priamo prejaví zmenou charakteristík v modeli. Proces kladenia otázok je riadený definovanými pravidlami. Doplnením priority k jednotlivým charakteristikám vieme vygenerovanie otázky úplne potlačiť.

## 2 Princíp generovania otázky

Pri explicitnom získavaní a udržiavaní charakteristík v modeli používateľa využívame priamu odozvu od používateľa, ktorá môže byť realizovaná napr. vyplnením anketu, formulára, odpoveďou na otázku a pod. Pri kladení otázok rozlišujeme (1) otázky, na ktoré možno odpovedať chýbajúcou časťou a (2) otázky s kladnou alebo zápornou odpoveďou. Pri návrhu metódy vychádzame z princípu, kde názov charakteristiky používateľa je priamo časťou otázky. Názov charakteristiky úzko súvisí s jej reprezentáciou v modeli používateľa prostredníctvom elementu vlastnosť (*property*).

Na obrázku 1 je uvedený príklad štyroch otázok a odpovedí, ktorý demonštruje opísaný spôsob generovania otázok. V otázke, kde očakávame nejaký údaj (napr. číselný, textový, ordinálny alebo booleovský) sa použije názov charakteristiky (zvýraznený podčiarknutým písmom). Odpoveď môže opäť obsahovať názov charakteristiky a jej príslušnú hodnotu, ktorú zadá používateľ (zvýraznené tučným šikmým písmom). Táto hodnota sa zapíše do modelu používateľa.

- Q: What is your name?
- A: My name is **John**.
- Q: Have you got any driving experiences?
- A: **Yes**, I have got.
- Q: Please enter your expected salary range.
- A: I expect salary from **25.000** to **40.000**.
- Q: How often are you willing to travel for a job?
- A: **Every day**.

Obrázok 1. Príklad otázok a odpovedí.

Pre takto navrhnutý spôsob generovania otázok, sme zvolili princíp šablón otázok. Šablóna otázky je vopred definovaná reprezentácia otázky, ktorá využíva slovnú zásobu. Postupnosť slov v šablóne otázky je vopred definovaná autorom šablóny. Slovo, ktoré má nahradiť názov charakteristiky, je označené špecifickou značkou. S kladením otázky súvisí nasledujúci kontext:

- *okolnosti* – čo musí nastať, aby bola pre používateľa vygenerovaná otázka,
- *predmet otázky* – aká otázka sa má vygenerovať, resp. na čo sa má systém “opýtať”.

Okolnosti, kedy bude používateľovi vygenerovaná otázka, sú kontrolované pomocou pravidiel a predmet otázky závisí od charakteristiky.

Na obrázku 2 je znázornený princíp generovania otázky. Obrázok slúži iba na základné vysvetlenie navrhnutej metódy, a preto neobsahuje úplný tok spracovania údajov.

Predpokladajme, že modelujeme znalosti používateľa v rámci aplikačnej domény pracovných ponúk. Na začiatku vytvoríme zoznam charakteristík pracovnej ponuky, na ktoré sa chceme vygenerovanými otázkami opýtať a takto inicializovať nové, prípadne aktualizovať existujúce charakteristiky v modeli používateľa. Postup získavania a následného vytvorenia novej (neexistujúcej) inštancie charakteristiky sa odlišuje od udržiavania (aktualizácie) existujúcej charakteristiky.

V prípade vytvorenia novej inštancie charakteristiky sú z *Úložiska viazania* vybrané charakteristiky, ktoré nemajú doposiaľ vytvorenú inštanciu charakteristiky v modeli používateľa. Pri udržiavaní sú zvolené charakteristiky na základe vyhodnotenia pravidiel. V oboch prípadoch je však vytvorený zoznam jednoznačných pomenovaní charakteristík, usporiadaných podľa priority. Priorita rieši aj prípady, kedy je vhodné vygenerovanie otázky potlačiť (napr. pohlavie používateľa je charakteristika, ktorú stačí zistiť raz).

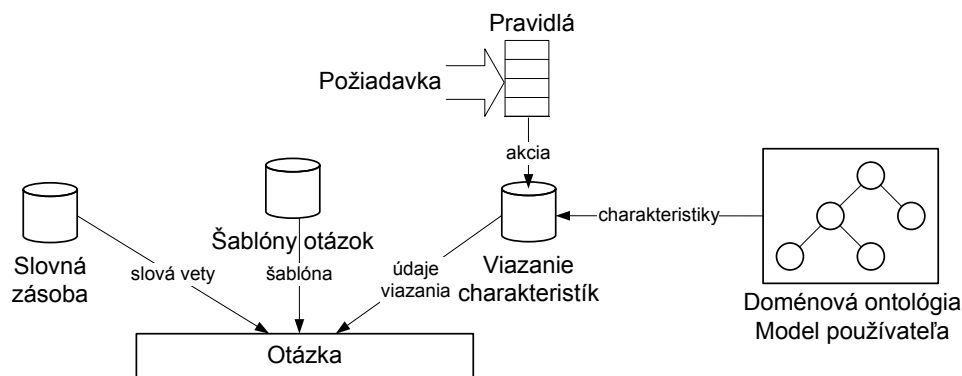
Pomenovanie a priorita charakteristiky, spolu s jej štruktúrou, sú uložené v *Úložisku viazania*. K charakteristike je tiež naviazaná šablóna otázky a podstatné meno, ktoré je vo vygenerovanej otázke použité ako názov charakteristiky. Podľa jedinečného identifikátora charakteristiky sa vyhledá v *Úložisku šablón* otázok príslušná šablóna. Jednotlivé slová sú uložené v *Slovnéj zásobe*, odkiaľ sa pre príslušnú šablónu načítajú. Týmto spôsobom vznikne otázka, ktorá je položená používateľovi. Používateľ taktiež získa časť vopred pripravenej odpovede v prípade ak je odpoveďou hodnota z množiny typov ordinálna hodnota alebo voľba.

### 2.1 Slovná zásoba

Pre slovnú zásobu sme pôvodne navrhli a vytvorili ontológiu *Part of Speech*, ktorá uchováva slová vo forme inštancií, pričom každé slovo má špecifikovaný slovný druh a gramatickú kategóriu. Od tohto prístupu sme ustúpili z dôvodu časovej náročnosti potrebnej na naplnenie ontológie a nízkej efektívnosti softvérových prostriedkov využívajúcich ontológiu, čo by sa prejavilo časovo kritickým generovaním otázok.

Používame slovnú zásobu, ktorá predstavuje použitie výsledkov existujúceho projektu – tréningové a testovacie údaje využité pre určenie slovných druhov v texte (Text Chunking<sup>1</sup>). Údaje sú reprezentované vo forme *identifikátor;slovo;značka slovného druhu*, napr. *25;ability;NN (POS tag)* v anglickom jazyku, kde prvý stĺpec reprezentuje identifikátor slova, na ktorý

<sup>1</sup> <http://www.cnts.ua.ac.be/conll2000/chunking/>



Obrázok 2. Doménovo závislá časť modelu používateľa.

sa odkazuje v šablóne otázok, nasleduje samotné slovo a značka slovného druhu. Značka slovného druhu v niektorých prípadoch určuje nielen slovný druh, ale aj gramatickú kategóriu slova (*NV* – podstatné meno v jednotnom čísle). Po vyčistení slovné zásoby (odstránenie čísel, prebytočných členov a pod.) obsahuje približne 17.000 slov. Značky slovných druhov špecifikujú nasledovné gramatické kategórie:

1. *číslo* – použitie pri podstatných menách, zámenách; môže byť jednotné a množné,
2. *čas* – pri slovesách; minulý, prítomný a budúci,
3. *osoba* – pri podstatných menách, zámenách a slovesách; prvá, druhá a tretia osoba.

Hlavnou výhodou tohto spôsobu reprezentácie je skutočnosť, že poznáme sémantiku jednotlivých slov, z ktorých sa skladajú otázky a vyššia efektivita spracovania oproti využitiu ontológie *Part Of Speech*. Slovná zásoba navyše obsahuje dostatočný počet slov, takže nepredpokladáme potrebu jej udržiavania. Údržba je vzhľadom na použitý formát slovné zásoby jednoduchá.

## 2.2 Šablóny

Z dôvodu znovupoužitia šablón, ako aj možnosti kedykoľvek vymeniť slovnú zásobu, sme sa rozhodli oddeliť slovnú zásobu od šablón otázok. Šablóna otázky pozostáva zo slov, ktoré sú reprezentované v XML súbore prostredníctvom identifikátora, tak ako sú uložené v slovné zásobe.

Na nasledujúcom výpise je znázornený príklad šablóny otázky *ValueTemplate01*:

```
<question name="ValueTemplate01">
  <pos order="1" instanceName="15432"/>
  <pos order="2" instanceName="7398"/>
  <pos order="3" instanceName="12456"/>
  <pos order="4" instanceName="n01"/>
</question>
```

Element *question* obsahuje elementy *pos*, ktoré špecifikujú identifikátory slov v slovné zásobe. Identifikátory 15432, 7398, 12456 budú neskôr nahradené slovami *what*, *is*, *your*. Špecifikované je aj poradie slov v otázke prostredníctvom atribútu *order*.

Vytvorili sme niekoľko šablón otázok. So zvyšujúcim sa počtom šablón otázok, ktoré budú k dispozícii, bude konverzácia s používateľom prirodzenejšia. Avšak veľký počet šablón, t.j. stav, kedy je pre každú vygenerovanú otázku vytvorená samostatná šablóna, nie je veľmi vhodný z hľadiska efektívnosti ich vytvárania. Z tohto dôvodu zohľadňujeme pri vytváraní šablón otázok a ich viazaní na konkrétnu charakteristiku v modeli používateľa univerzálnosť šablóny.

## 3 Viazanie charakteristík

Pre viazanie sme sa rozhodli z praktického dôvodu, keďže priamo do ontológie nemusíme pridávať koncepty, ktoré by špecifikovali údaje potrebné pre generovanie otázky, ako napríklad priority charakteristík. Do ontológie pristupujeme iba vtedy, ak potrebujeme načítať konkrétne hodnoty charakteristiky alebo sa vytvorila, resp. aktualizujú príslušné inštanície charakteristiky.

Viazaním obohacujeme charakteristiky modelu používateľa o špecifické údaje potrebné pre navrhnutú metódu. Špecifickými údajmi sú:

- jedinečný identifikátor charakteristiky,
- priorita charakteristiky,
- šablóna otázky a
- podstatné meno použité vo vete, ktoré predstavuje názov charakteristiky (nemusí byť jedinečné).

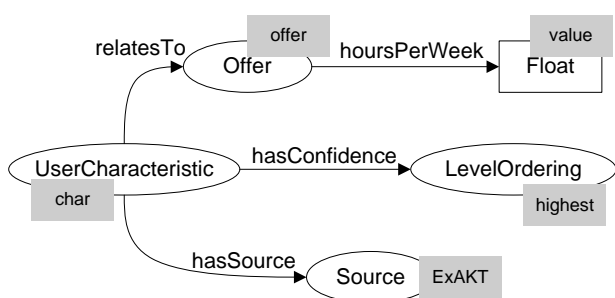
Charakteristika je v ontológii zložená z rôznych konceptov. Pri viazaní štruktúry charakteristiky vižeme nasledujúce koncepty:

- priestory mien (*namespaces*),

- triedy (*classes*),
- vlastnosti (*properties*) dátové alebo objektové a
- inštancie.

Jednoduché charakteristiky majú tvar RDF trojice, t.j. subjekt, predikát a objekt, kde subjekt odpovedá triede charakteristiky, predikát názvu a objekt hodnote charakteristiky. Takáto štruktúra je jednoducho viazateľná. Pri experimentoch s ontológiou pracovných ponúk sme zistili, že uvedený spôsob nepostačuje pre reprezentáciu komplexnejších charakteristík, ktoré môžu spolu súvisieť, môžu vzájomne vytvárať taxonómiu alebo ak je charakteristika vyjadrená ordinálnou hodnotou (jednotlivé ordinálne hodnoty môžu byť reprezentované triedou alebo inštanciou).

Pri takýchto charakteristikách je zložitejší aj spôsob vytvorenia, resp. aktualizácie charakteristiky. Obrázok 3 znázorňuje štruktúru časti modelu používateľa pre charakteristiku vyjadrujúcu používateľom preferovaný počet pracovných hodín v týždni, pričom pri pridaní charakteristiky sa určí aj jej ďalší atribút – dôveryhodnosť.



**Obrázok 3.** Príklad štruktúry konceptov pre reprezentáciu počtu pracovných hodín do týždňa.

Pre ľahšie pochopenie štruktúry sme názvy tried a atribútov nahradili jednoduchými symbolickými menami. Trieda *UserCharacteristic* predstavuje samotnú charakteristiku, má reláciu s pracovnou ponukou (*Offer*), ktorá má dátovú vlastnosť *hoursPerWeek* typu reálne číslo (*Float*). Charakteristika má reláciu vyjadrujúcu jej dôveryhodnosť, ktorá je ordinálnou hodnotou reprezentovanou inštanciami triedy *LevelOrdering* a reláciu *hasSource* s triedou vyjadrujúcou zdroj, ktorý charakteristiku do modelu používateľa pridal. V sivých obdĺžnikoch sú uvedené názvy inšancií, ktoré majú byť vytvorené po zadaní používateľovej odpovede.

Podobne sme navrhli spôsob pomenovania aj pre inštancie, aby bola zachovaná štruktúra konceptov pri vytvorení inštanície. Napríklad pre triedu *UserCharacteristic* a jej inštanciu *char* sme určili, že bude obsahovať dodatočný sufix: časovú pečiatku. Tento spôsob

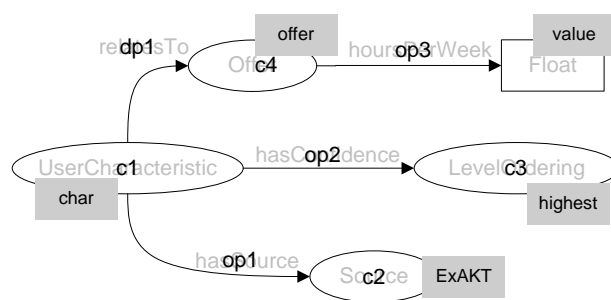
vytvárania inštancie sme pomenovali vzor (*pattern*). Rovnaký princíp uplatníme aj pre inštanciu *offer*. Inštancia *highest* (v rámci ontológie už existuje) vyjadrujúca úroveň dôveryhodnosti bude prepojená s inštanciou *char*. Inštancia triedy *Source* predstavuje názov nástroja, ktorý urobil zmenu charakteristiky ako posledný, bude mať názov *ExAKT*.

Pre zhrnutie, definovali sme tri spôsoby pomenovania inštancie:

- *pomenovanie podľa vzoru* – vytvorí sa inštancia podľa stanoveného názvu so sufixom časovej pečiatky, pričom jej formát je konfigurovateľný,
- *pomenovanie podľa názvu* – vytvorí sa inštancia podľa definovaného názvu,
- *nastavenie želanej hodnoty* (ordinálna, ale aj textová, numerická) – k relácii bude priradená želaná hodnota.

Pre reprezentáciu viazaných konceptov sme zvolili spôsob pomenovania podľa jednoznačných identifikátorov – tzv. kľúčov, ktoré sú uložené v *Úložisku viazania*. Názov kľúča pozostáva z písmena (*c* – trieda, *op* – objektová vlastnosť, *dt* – dátová vlastnosť, *i* – inštancia) a *postfixu* – číslo poradia v rámci *Úložiska viazania*.

Každý kľúč má priradený názov, podľa ktorého sa odvolávame na daný koncept, k nemu príslušný názov konceptu, priestor mien konceptu (v prefixovom tvare) a názov a typ inštancie, ktorá sa má vytvoriť. Využitie viazania prostredníctvom kľúčov zvyšuje flexibilitu viazania štruktúry v prípade, ak dôjde k zmenám v doménovej ontológii, resp. v modeli používateľa. Úložisko viazania zároveň obsahuje aj zoznam priestorov mien viazanej ontológie (pre viazanie prefixového tvaru na kompletný priestor mien vo forme URI). Obrázok 4 ilustruje štruktúru ontológie s uvedením jednotlivých kľúčov priradených ku konceptom.



**Obrázok 4.** Kľúče konceptov.

Ak by charakteristika obsahovala kompletnú štruktúru konceptov, t.j. relácie medzi triedami a inštanciami, bolo by viazanie neefektívne z hľadiska veľkej

závislosti na štruktúre charakteristiky v modeli používateľa. Namiesto tohto sa načítajú iba objektové alebo dátové vlastnosti. Ich doménové triedy a triedy rozsahu sa načítajú prostredníctvom špecifických dopytov.

Môže však nastať prípad, kedy jedna viazaná vlastnosť (objektová alebo dátová) obsahuje viacero doménových tried alebo tried rozsahu. Vtedy je potrebné pre konkrétnu vlastnosť definovať, ktorá jej doménová trieda alebo trieda rozsahu bude použitá pre vytvorenie inštancie. Keďže táto definícia je uvedená iba v konkrétnej charakteristike, neovplyvní to ostatné viazané charakteristiky, ktoré obsahujú rovnakú vlastnosť.

Počas experimentov s viazaním štruktúry charakteristiky sme zistili, že v niektorých prípadoch je nevyhnutné špecifikovať konkrétnu doménovú triedu alebo triedu rozsahu (napr. ak je vlastnosť špecifikovaná pre konkrétnu triedu, ale trieda má niekoľko potomkov, pričom chceme špecifikovať vlastnosť pre konkrétneho potomka).

Na nasledujúcom výpise je znázornená časť charakteristiky *hoursPerWeek*:

```
<properties characteristic="hoursPerWeek"
  priority="2">
  <property name="op3" noun="none"
    sentenceTemplate="none" domainConcept="c1"
    rangeConcept="none" increment="false"/>
  <property name="op8" noun="none"
    sentenceTemplate="none"
    domainConcept="c2" rangeConcept="c100"
    increment="false">
    <setOptionValue name="c100" value="dp4"/>
  </property>
  <property name="op9" noun="hours per week"
    sentenceTemplate="QuantityTemplate"
    domainConcept="c2" rangeConcept="dp5"/>
</properties>
```

Špecifikácia štruktúry charakteristiky je ohraničená elementom *properties*. Atribút *characteristic* definuje názov charakteristiky. Vlastnosti (dátové a objektové) sú reprezentované elementom *property*. Atribút *name* je identifikátor konceptu, *noun* špecifikuje podstatné meno, ktoré sa použije v otázke, *sentenceTemplate* špecifikuje názov šablóny otázky, *domainConcept* a *rangeConcept* špecifikujú triedy domény a rozsahu – implicitne sú nastavené na *none*, v prípade ak sa požaduje explicitne špecifikovať triedy, uvedie sa názov konceptu. Atribút *increment* je aplikovateľný iba v prípade číselných literálov, napr. počet aktualizácií charakteristiky. Ak atribút *increment* obsahuje hodnotu *true*, zodpovedajúci literál bude inkrementovaný pri každej aktualizácii charakteristiky. Element *setOptionValue* nastavuje hodnotu typu voľba ako triedu rozsahu v rámci elementu *property*.

## 4 Pravidlá

Pre efektívne udržiavanie charakteristík v modeli používateľa sme zvolili využitie pravidiel. Výhodou tohto prístupu je jednoduchosť vytvárania a údržby. Využívame klasický koncept pravidiel: *IF* [*podmienka*] *THEN* [*akcia*]. Súčasťou podmienky je vlastnosť z modelu používateľa a k nej príslušný operand a parameter podmienky. Akcia je štandardne určená ako vygenerovanie otázky. Pre definovanie podmienky sme navrhli využitie údajových typov – literálov a objektov, ktoré môžu reprezentovať hodnotu charakteristiky. Literály sú reprezentované údajovými typmi jazyka XML. Objektové hodnoty sú nasledujúce:

- *ordinálna hodnota* – vyjadruje mieru charakteristiky, ktorá nie je reprezentovaná číslom, ale jej hodnoty možno zoradiť a vzájomne porovnávať a
- *voľba (option)* – vyjadruje nemerateľnú charakteristiku (napr. znalosti z programovacieho jazyka Java).

V podmienke môžu byť použité známe operandy (menší, väčší, rovný) a špeciálny operand inštancia (*instanceOf*), ktorý sa využije iba pri hodnote typu voľba a môže vyjadrovať podmienku relácie medzi inštanciou a triedou. V prípade literálov musí byť parametrom podmienky zodpovedajúci typ, t.j. pre číselný literál (patrí sem aj dátum) sa použije číselná hodnota parametra a pre reťazec hodnota typu reťazec. V prípade literálu typu dátum je nevyhnutné uviesť časovú jednotku, ktorá môže byť v rozsahu minúty až roky. V prípade objektových hodnôt sú to existujúce názvy inštancií alebo tried.

Vhodne navrhnutým pravidlom môžeme vyvolať vygenerovanie otázky, napr. ak špecifická charakteristika je staršia ako definované časové obdobie. Ďalším príkladom môže byť vygenerovanie otázky pre charakteristiku v závislosti od toho, ktorý nástroj zabezpečil jej vytvorenie, resp. aktualizáciu.

Nasledujúci výpis predstavuje vybrané pravidlá:

```
<rule type="dateTime">
  <condition parameter="gu:hasTimeStamp"
    operator="less" value="1" unit="minute"/>
  <action type="ask"/>
</rule>
<rule type="ordinalValue">
  <condition parameter="c:hasConfidence"
    operator="less" value="c:_loAverage"/>
  <action type="ask"/>
</rule>
```

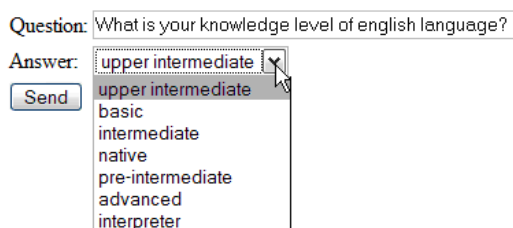
## 5 Zhodnotenie

Väčšina adaptívnych webových systémov sa zameriava na prispôsobovanie viditeľných aspektov (obsahu, na-

vigácie, prezentácie) na základe modelu používateľa. Prínosom tohto príspevku je metóda na prispôbovanie samotného modelu používateľa, ktorého aktuálnosť je nevyhnutná pre efektívne prispôbovanie.

Metóda explicitného získavania a udržiavania charakteristík v modeli používateľa je založená na kladení otázok v prirodzenom jazyku. Na vygenerovanie otázok využívame koncepty modelu aplikačnej domény reprezentovaného ontológiou.

Navrhnutá metóda bola overená softvérovým nástrojom ExAKT, ktorý je implementovaný v jazyku Java. Na obrázku 5 je zobrazený výstup z nástroja – vygenerovaná otázka s možnosťou voľby odpovede. Alternatívy odpovede sú získané z doménového modelu. Nástroj pri práci s ontologickými modelmi využíva rámec Sesame (<http://www.openrdf.org>). Modely sú prezentované v jazyku OWL DL.



**Obrázok 5.** Používateľské rozhranie pre odpoveď s viacerými hodnotami.

Overenie prebehlo na doménovej ontológii pracovných ponúk zúženej na oblasť informatiky (vytvorená v rámci projektu NÁZOU [4]), ontológii publikácií (projekt MAPEKUS [3]) a odpovedajúcich modeloch používateľa. V oboch prípadoch sme naviazali vybrané charakteristiky, vytvorili pre ne šablóny otázok a definovali pravidlá pre udržiavanie. Overili sme vytvorenie nových charakteristík (vhodné na inicializáciu modelu používateľa pri prvom prihlásení používateľa do systému) a aktualizáciu existujúcich charakteristík v modeli používateľa.

Pri testovaní metódy na doménovej ontológii cestovného ruchu sme narazili na ohraňovanie metódy, ktoré spočíva vo viazaní konceptov. Napr. chceli by sme otázkou zistiť ohodnotenie ubytovanie (trieda *Accommodation*) v rámci destinácie (trieda *Destination*) a zistiť kontaktné údaje. Vytvorené by však boli dve nezávislé inštancie. To odporuje metóde, ktorá očakáva v otázke názov jednej charakteristiky.

Pri vyhodnotení univerzálnosti šablón sme sledovali počet otázok, ktoré možno vygenerovať z jednej šablóny. Pre doménu pracovných ponúk sme vytvorili 7 šablón, pomocou ktorých sme vygenerovali otázky pre 22 charakteristík. Pre doménu publikácií sme vytvorili 3 šablóny a vygenerovali 5 otázok. V prípade

podobnosti štruktúry viacerých charakteristík sa miera univerzálnosti využitia šablón zvyšuje.

Aktualizácia jednotlivých charakteristík používateľa priamo ovplyvnila 10 – 16 výrokov v modeli používateľa. Preto sme sa zamerali aj na vyhodnotenie času potrebného na aktualizáciu charakteristiky. Ten sa pohyboval rádovo v jednotkách sekúnd v závislosti od komplexnosti charakteristiky. Pre 10 výrokov sme dosiahli priemerný čas 4,7 sekundy a 6,4 sekundy pre 16 výrokov.

Pri overení metódy sme sa zamerali na udržiavanie charakteristík jedného používateľa. V ďalšej práci plánujeme rozšírenie pre viacerých používateľov. Zároveň pracujeme na implementácii nástroja pre poloautomatizované viazanie charakteristík modelu používateľa.

*Tento príspevok vznikol za podpory Štátneho programu výskumu a vývoja “Budovanie informačnej spoločnosti” č. úlohy 1025/04 a Vedeckej grantovej agentúry VEGA v rámci grantovej úlohy č. VG1/3102/06.*

## Referencie

1. Androutsopoulos, I., Kallonis, S., Karkaletsis, V.: Exploiting OWL Ontologies in Multilingual Generation of Object Descriptions. In Proceedings of the 10th European Workshop on Natural Language Generation, Aberdeen Scotland, August (2005) 150–155
2. Bieliková, M.: An adaptive web-based system for learning programming. Int. J. Continuing Engineering Education and Life-Long Learning, Inderscience, **16** (2006) 122–136
3. Bieliková, M., Návrat, P.: Modelovanie a získavanie, spracovanie a využívanie znalostí o konaní používateľa v hyperpriestore Internetu. In Proc. of Znalosti 2007, Ostrava, Česká republika, Február (2007) 368–371
4. Bieliková, M., Návrat, P., Vojtáš, P., Hluchý, L., Bartoš, P.: Softvérové nástroje pre získavanie, organizovanie a prezentáciu pracovných ponúk na webe. In Proc. of Datakon 2006, Brno, Česká republika (2006) 1–20
5. De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, S., Stash, N.: AHA! The adaptive hypermedia architecture, In Proceedings of the ACM Conf. on Hypertext and Hypermedia, Nottingham, UK (2003) 81–84
6. Denaux, R., Aroyo, L., Dimitrova, V.: An Approach for Ontology based Elicitation of User Models to Enable Personalization on the Semantic Web. In Proceedings of WWW 2005 conference, ACM Press (2005) 1170–1171
7. Gurský, P., Horváth, T., Novotný, R., Vaneková, V., Vojtáš, P.: UPRE: User preference based search system. In Int. Conf. on Web Intelligence. (2006) 841–844
8. Kruijff, G. M.: Context sensitive utterance planning for CCG. In Proc. of the 10th European Workshop on Natural Language Generation, Scotland (2005) 83–90
9. Šimún, M., Andrejko, A., Bieliková, M.: Inicializácia a aktualizácia modelu používateľa pri hľadaní pracovných ponúk na webe. In Proc. of Znalosti 2007. Ostrava, Česká republika, Február (2007) 109–120