

Analýza obsahu kurzu, návrh navigace vytvořené z textu při použití pojmů ontologie

Zdeněk Velart¹ a Petr Šaloun²

¹Katedra informatiky, FEI, VŠB-TU Ostrava
zdenek.velart.fei@vsb.cz

²Katedra informatiky a počítačů, PřF Ostravská univerzita v Ostravě
petr.saloun@osu.cz

Abstrakt Kurz členěný na koncepty a jejich další části postupně vytváří pojmový aparát znalostí odpovídajících obsahu. Analýza kurzu založená na klíčových slovech domény použitých v příkladech a na jejich spojení s ontologií aplikační domény umožňuje navrhnout více vstupních bodů či více doporučených průchodů kurzem. Znalosti jsou buď vytvářeny nebo používány. Nabídku navigace lze adaptovat podle vstupních znalostí uživatele nebo podle požadovaného cíle. V příspěvku popisujeme náš probíhající výzkum v této oblasti.

Klíčová slova: personalizace, webové adaptivní systémy, ontologie, navigace

1 Úvod

Ve většině současných kurzů je obsah představován posloupností konceptů (tedy kapitol, podkapitol případně jejich částí), tak jak je autor vytvořil a případně navíc propojil odkazy. Takový sekvenční průchod nemusí vyhovovat všem.

Naším cílem je tvorba navigace nad obsahem kurzu, konkrétně *Programování v C++*, založená na znalostech obsažených v elementech ontologie a jejich vzájemných vazbách. K tomu využíváme analýzu ukázkových zdrojových textů programů a četnosti výskytu klíčových slov jazyka C++ ve zdrojových textech a v doprovodném textu v jednotlivých konceptech. Znalosti dělíme na množinu nových znalostí a množinu znalostí vyžadovaných v obsahu konceptu.

Navigace uživateli nabídne pojmy ontologie aplikační domény, které jsou spojeny s klíčovými slovy jazyka C++. Námí vytvořená ontologie C++ má přibližně 300 tříd. Odpovídající třídy jsou rozšířeny o klíčová slova jazyka, což umožňuje propojení pojmu ontologie s nalezeným klíčovým slovem. Tuto činnost podporuje programový nástroj vytvořený na základě námi navrženého algoritmu.

Popsaný způsob adaptace a personalizace jsme diskutovali v [6]. Potřebné propojení pojmů ontologie a obsahu nepodporuje ani AHA!, viz [3], se kterým jsme dosud pracovali [2,5], ani žádný z dalších dostupných adaptivních webových systémů, jakými jsou například ALEA [1], NavEx [4], QuizGuide a SQLTutor.

2 Struktura kurzu založená na ontologii

Kurz je zpravidla rozdělen do kapitol. Kapitola je rozdělena do konceptů, koncept představuje jednu stránku, nebo také fragment stránky, případně skupinu více stránek. Každý koncept obvykle obsahuje odkazy na jeden či více ukázkových příkladů. V okolí začlenění se diskutují zejména nové skutečnosti demonstrované v ukázce.

Navrhovaná navigace nad koncepty kurzu je založena na použitých a definovaných množinách pojmů a znalostí spíše, než na propojení jednotlivých konceptů či stránek pomocí odkazů. Každý koncept popisujeme dvěma nezávislými množinami, a to *množinou prerekvizit* a *množinou naučených informací*. Tyto množiny se skládají z elementů ontologie, která popisuje aplikační doménu.

Navigace v kurzu je následující: studentovi je umožněn přístup jen na koncepty, jejichž prerekvizity jsou obsaženy ve studentově znalostech. Tedy je student již nabyt v minulosti, například studem příbuzného programovacího jazyka, nebo prošel koncepty, které je obsahovaly v množinách naučených informací.

Za vstupní body kurzu je možno prohlásit takové koncepty, které mají prázdnou množinu prerekvizit případně jejich prerekvizity odpovídají již dosaženým znalostem. Není-li podmínka splněna, určuje autor kurzu některý z konceptů jako vstupní. Například v kurzu C++ je možné jako vstupní bod klasicky použít koncept se základními stavebními kameny jazyka, nebo alternativně začít zpracováním řetězců. Možnost volby je pro studenta významná.

3 Algoritmus propojení kurzu a ontologie

Náš přístup je založen na propojení klíčových slov domény s elementy ontologie a jejich použití ve výukovém materiálu. Algoritmus jsme implementovali do funkčního prototypu programu.

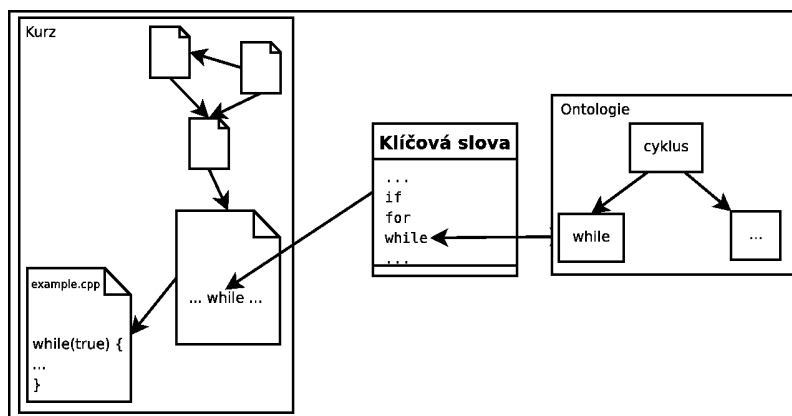
Mějme obsah kurzu rozdělený do konceptů v jednotlivých souborech označovaných XHTML nebo DocBook. Dále mějme ontologii aplikační domény a případně seznam klíčových slov, pokud již nejsou součástí ontologie. Každou třídu ontologie tedy máme spojenou s klíčovým slovem nebo slovem, které tuto třídu charakterizuje v textu kurzu. Nalezneme všechny výskyty klíčových slov v každém konceptu a uspořádáme je podle jejich četnosti. Pokud koncept obsahuje začleněné příklady kódu v programovacím jazyce, provedeme pro ně totéž. Triviální korelaci mezi četností výskytu klíčových slov ve zdrojových textech a četností výskytu slov vztahujících se ke třídě ontologie v okolním textu konceptu rozdělíme nalezená klíčová slova do tří nezávislých množin, ma *množinu vstupních slov*, *množinu výstupních slov* a *množina nepoužitých slov*. Tyto množiny slouží ke stanovení množin prerekvizit a nových znalostí, které jsou východiskem pro navigaci nad obsahem. Naše dosavadní metrika vyžaduje zásah uživatele do navrženého rozdělení. Na zkvalitnění algoritmu a naznačené metriky intenzivně pracujeme.

Nalezené množiny *vstupních a naučených znalostí* ukládáme ve formě metatagů přímo do konceptů kurzu, ať již se jedná o XHTML nebo DocBook

značkování. Alternativou je ukládání nalezených množin do samostatného XML souboru pro pozdější použití v systému nabízejícím tento kurz.

1. Pro kurz definuj množinu konceptů $K = \{k|k \text{ je koncept}\}$.
2. Načti a/nebo definuj množinu $KS = \{ks|ks \text{ je klíčové slovo}\}$ klíčových slov.
3. Načti existující ontologii $O = \{o|o \text{ je element}\}$.
4. $\forall o \in O$ urči KSO , kde $KSO \subset KS$.
5. $\forall k \in K$ najdi výskyty klíčových slov a definuj množinu dvojic $KSK = \{\{ks, pocet\}|ks \in KS \text{ a } pocet \text{ je počet výskytů } ks \text{ v } k\}$. Seřď KSK podle $pocet$ počtů výskytů.
6. Rozděl množinu KSK na množinu vstupních slov VsS , množinu výstupních slov VS a množinu nepoužitých NS , kde $VsS \cup VS \cup NS = KSK$, kdy platí $VsS \cap VS = \emptyset \wedge VsS \cap NS = \emptyset \wedge VS \cap NS = \emptyset$.
7. $\forall k \in K$ definuj množinu prerekvizit P a množinu naučené informace N , kde $P = \{o|o \in O \wedge (\forall ks \in VsS \wedge ks \in KSO)\} \wedge N = \{o|o \in O \wedge (\forall ks \in VS \wedge ks \in KSO)\}$ a platí $P \cap N = \emptyset$.
8. Ulož nalezené informace.

Pro lepší celkovou představu o popisovaných vztazích a činnosti algoritmu viz obr. 1. V současnosti intenzivně testujeme implementaci algoritmu, zejména hledáme lepší metriky pro činnost podle bodů 5 – 7 algoritmu. K dispozici máme kurz C++ ve formátu XHTML, vlastní ontologii C++ jsme vytvořili v jazyce OWL. Elementy ontologie jsou stromově uspořádány podle hierarchie vazeb předek–potomek. Vztahy mezi elementy ontologie jsme definovali vycházejíce z normy jazyka C++.



Obrázek 1. Vztah mezi kurzem, klíčovým slovem a elementem ontologie

4 Závěr a budoucí práce

V příspěvku popisujeme probíhající výzkum nalezených možností navigace nad obsahem kurzu vycházející z jeho obsahu, z ontologie aplikační domény, z klíčových slov jazyka a z aktuálních znalostí uživatele. Předložili jsme algoritmus, který v jeho implementaci testujeme a ladíme s reálnými daty kurzu programování v C++. Navigace nabízí více možných průchodů obsahem kurzu.

V budoucnu připravíme množiny znalostí uživatelů znalých programování v příbuzných programovacích jazycích tak, abychom jim nabízeli pro ně nové znalosti. S tím související mapování pojmů ontologií příbuzných objektových či procedurálních jazyků předpokládáme jako blízký směr našeho zájmu. Budeme muset realizovat i interpretaci navigace nad obsahem kurzu, neboť dostupné systémy, včetně AHA!, nemají podobnou funkcionalitu implementovanou.

Reference

1. Bieliková M., Moravčík M. Modeling the Content of Adaptive Web-Based System Using an Ontology. *In proceedings of 1st International Workshop on Semantic Media Adaptation and Personalization SMAP06*. IEEE Computer Society. ISBN: 0-7695-2692-6
2. Bober, M., Šaloun, P., Velart, Z. Knowledge mining from adaptive course. In IC-ETA 2007. Vysoké Tatry, Stará Lesná. Košice, Slovensko : elfa, 2007. s. 1-6. ISBN 978-80-8086-061.
3. De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Santic T., Smits D., Stash N. AHA! The Adaptive Hypermedia Architecture. *In Proceedings of the ACM Hypertext Conference*, Nottingham, UK, August 2003, pp. 81-84
4. Yudelso M., Brusilovsky P., Sosnovsky S. Accessing Interactive Example with Adaptive Navigation Support. *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)* ISBN 0-7695-2181-9/04
5. Velart Z., Šaloun P. Výuka programování C++ v adaptivním systému AHA!. *In Technologie pro e-vzdělávání 2006*, ČVUT FEL, 2006, ISBN 80-01-03512-3
6. Velart Z. Šaloun P., Ontology Based Course Navigation. *In HT'07 (Hypertext 2007) Proceedings*. Manchester, United Kingdom, 2007, p. 151-152 ISBN 978-1-59593-820-6

Annotation:

Analysis of the course content and creation of the navigation based on interpretation and use of the elements of the ontology.

Courses which are presented to the users must not suit to different groups of users, despite the intentions and efforts of the author of the course. Course analysis, based on the keywords of the domain, their interconnection with the elements of ontology and their usage in particular concepts allows to propose such a course structure with more entry points and more recommended paths through the course adapted according the user knowledge or goals. The basic idea of use of the keywords is based on the number of occurrences in the learning material and associated examples. Proposed algorithm for processing the course and creating such a course structure is presented in the article.