

Vysoká škola báňská – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

---

DISERTAČNÍ PRÁCE

ADAPTIVNÍ PERSONALIZOVANÁ  
NAVIGACE ŘÍZENÁ METADATY

Ing. Zdeněk Velart

Školitel: doc. RNDr. Petr Šaloun, Ph.D.

2011

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. 9. 2011

.....

Rád bych poděkoval Petru Šalounovi za veškerou pomoc, ochotu a trpělivost. Rád bych také poděkoval všem, kteří mi byli oporou a jakýmkoliv způsobem mi pomohli. Nebudu je zde vyjmenovávat, abych omylem někoho neopoměl. Všem patří můj dík.



# Abstrakt

S informacemi se setkáváme každý den a jejich množství neustále roste. S rozvojem nových technologií je množství informací, které se valí na uživatele větší, takže uživatel je doslova zavalen informacemi. Za účelem aby bylo možné množství informací omezit a vybírat pro uživatele takové informace, které pro něj jsou přínosné, vzniklo mnoho přístupů, které se problémem zabývají. Všechny tyto přístupy a technologie sdílejí jeden společný element a tím jsou metadata. Metadata představují data o informacích, jejich sémantiku takovým způsobem, že umožňují strojové zpracování. Sémantický web se zabývá metadaty a způsobem jak metadata definovat, přiřadit a jak s nimi pracovat. Adaptivní web využívá metadata pro vyhledávání, agregování, personalizování a navigování uživatele k požadovaným informacím. Dva hlavní přístupy k adaptaci jsou adaptivní prezentace, která představuje metody a způsob prezentování informací uživateli a adaptivní navigace, která definuje metody pro navigování uživatele informačním prostorem.

V práci prezentujeme způsob navigace uživatele založenou na metadatech. Používáme prostor konceptů, který popisuje problémovou doménu pomocí konceptů a vztahů mezi nimi. Algoritmus pro ohodnocení prostoru konceptů, který definujeme umožňuje ohodnotit každý koncept v závislosti na jeho poloze v prostoru konceptů a vazbách na další koncepty a jejich ohodnocení. Každý dokument, který je prezentován uživateli je navázán na prostor konceptů pomocí své množiny konceptů. Pro uživatele je uchovávaná množina dosažených znalostí, kde jsou zaznamenávány znalosti uživatele ve formě známých konceptů. Navigace je prováděna pro každého uživatele samostatně na základě jeho množiny dosažených znalostí a množin konceptů příslušných dokumentů. Pro vlastní výběr dokumentů a jejich seřazení do personalizovaného menu se využívají metriky, které implementují rozdílné přístupy pro seřazení dokumentů, například na základě mohutnosti množiny konceptů, na základě ohodnocení dokumentů podle množiny konceptů a další. Dokumenty, které pro uživatele vhodné jsou mu prezentovány ve formě menu, seřazeného podle vhodnosti. Uživatelovo chování a

jeho reakce jsou využívány k dalšímu formování navigace.

Řešení bylo realizováno jako prototyp adaptivního webového systému s jehož pomocí byly získány experimentální výsledky podporující správnost návrhu.

## **Klíčová slova**

metadata, sémantický web, adaptivní web, adaptace, navigace, personalizace, prostor konceptů

# Abstract

Every day we meet with new information, news, videos, photos and many others. The amount is so overwhelming, that we can talk about information flood. In order to minimize the amount and to pick up the information the user needs many approaches and principles were developed. All the principles and approaches share the same common element – metadata. Metadata are data about information and they describe the semantics of the information in a way, that computers can automatically evaluate them. The area which is concerning with the metadata, how the assign and work with them is called semantic web. Adaptive web uses the metadata to search, aggregate, personalize and navigate user to the desired information. Two main areas of adaptation are adaptive presentation which defines methods for presenting the information to the user and adaptive navigation which defines methods for navigating the user in the information space.

The work presents a scheme for navigating user based on metadata. We employ concept space that describes the problem domain by concepts and relations between them. Algorithm for concepts space evaluation was developed and is used to calculate an evaluation of a concept according its position and evaluation of its related concepts. Document which are presented to the users have defined a set of concepts which describes them. For user an achieved knowledge set is maintained where user's actual knowledge is stored in a form of known concepts. Navigation scheme takes the user's achieved knowledge set and concepts sets of documents and chooses such documents which will be best beneficial for users in meaning of gaining new knowledge. For choosing the documents we employ numerous metrics which implement different approaches in learning of new knowledge such as choosing documents where user knows the most of the concepts and others or choosing document according concepts similarity to presented document. The chosen documents are presented to the user in a form of menu where the best suitable document is on the top. The user's behavior and reactions to the presented documents help to shape the navigation prepared for them.

The navigation scheme was implemented in adaptivi web-based prototype, which was used for experimental evaluation. The obtained results support the accuracy of the design.

## **Keywords**

metadata, semantic web, adaptive web, adaptation, navigation, personalization, concept space



# Obsah

<b>Abstrakt</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Úvod</b>	<b>1</b>
<b>I Současný stav poznání</b>	<b>5</b>
<b>2 Adaptační, personalizace a základní pojmy</b>	<b>7</b>
2.1 Základní pojmy . . . . .	8
<b>3 Sémantický web</b>	<b>11</b>
3.1 Architektura sémantického webu . . . . .	13
3.2 Metadata a ontologie . . . . .	14
3.3 Porovnávání a ohodnocování ontologií . . . . .	17
3.4 Mapování a slučování ontologií . . . . .	18
3.5 Topic maps . . . . .	21
3.6 Sociální sémantický web, folksonomie . . . . .	22
3.7 Vytváření prostorů konceptů . . . . .	22
3.8 Shrnutí . . . . .	23
<b>4 Adaptivní web</b>	<b>25</b>
4.1 Adaptivní techniky . . . . .	26
4.2 Modely adaptivních hypermédií . . . . .	29
4.3 Model adaptace . . . . .	30
4.4 Model uživatele . . . . .	32

4.5	Reprezentace modelu uživatele . . . . .	35
4.6	Cold start problém . . . . .	36
4.7	Adaptivní systémy a metodologie . . . . .	36
4.8	Shrnutí a vlastní řešení – XAPOS . . . . .	40
<b>II</b>	<b>Vlastní výzkum</b>	<b>43</b>
<b>5</b>	<b>Ohodnocení konceptů</b>	<b>45</b>
5.1	PageRank . . . . .	47
5.2	Pojmy a značení . . . . .	47
5.3	Ohodnocení relací . . . . .	48
5.4	Ranking algoritmus . . . . .	49
5.5	Shrnutí . . . . .	53
<b>6</b>	<b>Navigace</b>	<b>55</b>
6.1	Propojení prostoru konceptů a výukových objektů . . . . .	56
6.2	Model uživatele . . . . .	57
6.3	Navigace uživatele . . . . .	57
6.4	Shrnutí . . . . .	63
<b>7</b>	<b>Ověření výsledků</b>	<b>65</b>
7.1	Statistiky . . . . .	66
7.2	Vyhodnocení stráveného času . . . . .	68
7.3	Vyhodnocení testů . . . . .	70
7.4	Vyhodnocení cest . . . . .	71
7.5	Vzory chování . . . . .	74
7.6	Shrnutí . . . . .	76
<b>8</b>	<b>Zhodnocení a závěr</b>	<b>79</b>
8.1	Budoucí práce . . . . .	81
<b>9</b>	<b>Conclusions and future work</b>	<b>83</b>
	<b>Literatura</b>	<b>85</b>

<b>III Přílohy</b>	<b>95</b>
<b>A Vlastní práce vztahující se k tématu disertace</b>	<b>97</b>
<b>B XAPOS</b>	<b>99</b>



# Seznam tabulek

1	Vypočtené ohodnocení relací v prostorech konceptů . . . . .	49
2	Ohodnocení jednotlivých konceptů pro různý počet iterací – C prostor konceptů	51
3	Ohodnocení jednotlivých konceptů pro různý počet iterací – C++ prostor konceptů . . . . .	51
4	Ohodnocení jednotlivých konceptů pro různý počet iterací – Lisp prostor konceptů . . . . .	52
5	Počty přístupů uživatele . . . . .	67
6	Počty navštívení dokumentu uživatelem . . . . .	68
7	Vzory chování uživatelů . . . . .	76



# Seznam obrázků

1	Propojení prostoru konceptů a obsahu . . . . .	10
2	Architektura sémantického webu [10] . . . . .	14
3	Přehled technik adaptace . . . . .	27
4	Prostor konceptů programovacího jazyka Lisp . . . . .	46
5	Součet ohodnocení konceptů v prostorech konceptů C a C++ . . . . .	52
6	Součet ohodnocení konceptů v prostoru konceptů Lisp . . . . .	53
7	Návrh uživatelského GUI . . . . .	57
8	Způsob navigace uživatele nad obsahem . . . . .	58
9	Aktualizace uživatelského modelu a logu . . . . .	59
10	Vyhodnocení testu a aktualizace uživatelského modelu . . . . .	60
11	Přizpůsobení menu za použití metrik . . . . .	61
12	Počty přístupů uživatelů k dokumentům v rámci experimentu . . . . .	67
13	Průměrné počty přístupů uživatelů k výukovým objektům . . . . .	68
14	Intervaly časů strávených uživateli na výukových objektech (v sekundách) . . . . .	69
15	Graf průchodů uživatelů CZ kurzem . . . . .	72
16	Graf průchodů uživatelů TR kurzem . . . . .	73
17	Vzory chování uživatelů . . . . .	75
18	Vícejazyčný obsah . . . . .	101
19	Rozvržení GUI systému . . . . .	103





# Kapitola 1

## Úvod

Informační technologie se staly každodenní součástí našeho života. Přístup k Internetu je dnes již v mnoha zemích považován za jedno ze základních lidských práv. K masovému rozšíření informačních technologií přispěla jednak jejich jednoduchost, a také jejich atraktivita. Čím více se usnadnil přístup lidem k Internetu, tím častěji se začalo narážet na problémy s neustále se zvyšujícím množstvím dostupných informací. A čím více informací se na Internetu nachází, tím více lidí se k němu připojuje. Ovšem ne každý představuje „informačního experta“, který je schopen vše a bez problémů nalézt. Z tohoto důvodu se již od začátku používání Internetu začaly objevovat snahy o zjednodušení získávání informací ve formě vyhledávačů či různých nápoředných systémů.

Právě při velkém objemu zpracovávaných informací a velkém množství uživatelů se začalo projevovat dědictví a problémy, které si Internet sebou nese od svého počátku. Reakcí na objevující se problémy a potřeby byl jednak vývoj existujících standardů jako je HTML, tak také vytváření nových standardů a přístupů k zobrazení a nabízení informací. Jedním z těchto rozšíření, které vzniklo jednak z potřeby lépe organizovat a zpracovávat dostupné informace, a také z důvodu jejich lepší prezentace, je sémantický web. Princip sémantického webu stojí na myšlence, že informacím se má přiřadit přesně definovaný význam, který umožní jejich automatizované zpracování [9].

Ačkoliv sémantický web je odpovědí na mnoho problémů, které se vyskytly, stále se i v dnešní době objevuje mnoho nových problémů. Objevené problémy a výzvy byly různými autory řešeny odlišnými principy a postupy. Vznikla řada jazyků pro sémantický web, které si navzájem konkurovaly a u kterých se až časem hledala shoda a interoperabilita. V ruku v ruce s tímto se objevují problémy s kooperací a interoperabilitou mezi různými přístupy

a nástroji, které byly vytvořeny. Mnoho práce je proto věnováno na nástroje a postupy pro mapování, spojování či integraci existujících technologií a přístupů.

Adaptivní web představuje směr výzkumu, který se zabývá způsoby a principem jak uživateli prezentovat relevantní informace nebo jej navigovat informačním prostorem tak, aby se dostal k požadovanému cíli. Nejdůležitějším aspektem celého snažení je uživatel, a proto je nutné o něm získávat a uchovávat množství informací a charakteristik. Charakteristiky, které o uživateli systém uchovávají, se dají rozdělit podle různých kritérií například na charakteristiky statické a dynamické. Charakteristiky je možné získávat jednak od samotných uživatelů tak, že se jich přímo zeptáme například formou dotazníku či sledováním jejich interakcí se systémem. Ne vždy se systémy zaměřují pouze na jednotlivého uživatele, ale zaměřují se na skupinu uživatelů s podobných chováním, ke které je jednotlivý uživatel následně přiřazen.

Navigace uživatele představuje důležitý proces, kdy již získané charakteristiky uživatele a informace o jeho chování prakticky využíváme pro přizpůsobení prezentovaných informací či pro jeho navigaci informačním prostorem. Ať již navigujeme jednotlivého uživatele či skupinu uživatelů, k přípravě a provedení navigace potřebujeme mít definován algoritmus, který ze získaných charakteristik a dostupných informací připraví a nabídne uživateli takové informace, které jej budou směřovat k jeho cíli. Principy sémantického webu přinesly do navigace uživatele další možnosti pro adaptivní navigaci, kdy je možné do čistě adaptivní navigace zahrnout i dostupná metadata o obsahu, který pro uživatele adaptujeme.

Naším cílem je zlepšit orientaci a navigaci uživatele nad prezentovanými obsahem za pomoci metadat daného obsahu. Naš algoritmus navigace je postaven na myšlence, že pro navigaci nad obsahem můžeme využít metadata – prostor konceptů či ontologii – protože vazby, které lze nalézt na úrovni metadat, lze také nalézt na úrovni samotného obsahu, které tato metadata popisují. K tomuto účelu jsme navrhli algoritmus pro ohodnocení prostoru konceptů, kdy konceptu je přiřazeno ohodnocení na základě jeho pozice v prostoru konceptů a jeho vazeb na ostatní koncepty.

Takto zpracovaný a ohodnocený prostor konceptů využíváme pro navigaci uživatelů. Pro každého uživatele uchováváme množinu dosažených znalostí, která se naplňuje koncepty z prostoru konceptů. Tímto způsobem budujeme přehled o aktuálních znalostech uživatele a na jejich základě a na základě navázání obsahu na prostor konceptů připravujeme další navigaci pro uživatele. Pojem množina chápeme v infromatickém smyslu jako kolekci prvků, nikoliv striktně matematicky.

V rámci vytváření našeho přístupu jsme zpracovali a diskutovali dosažené výsledky v oblasti adaptivních hypermédií a v oblasti sémantického webu. Výsledky jsou prezentovány v dalším textu. Prakticky jsme vyzkoušeli adaptivní systém AHA! [24] a to včetně experimentu, který implementuje pravidlově orientovaný přístup, kdy pravidla musí zadat autor kurzu s ohledem na jeho obsah a zamýšlenou posloupnost jednotlivých stránek prezentovaných uživateli. Po vytvoření a otestování dvou kurzů v adaptivním systému AHA! jsme pravidlově orientovaný přístup opustili a zaměřili jsme se na princip navigace využívající metadata, která jsou definována v prostoru konceptů popisujícím danou problémovou doménu.

Naše výsledky v této oblasti byly prezentovány a diskutovány na recenzovaných lokálních i světových konferencích. Zpětná vazba týkající se použití prostoru konceptů respektive topic maps, nám pomohla ve směřování dalšího výzkumu. Princip využití prostoru konceptů tedy metadat nás v průběhu řešení přivedl na možnost navigace nad vícejazyčným obsahem. Koncepty v prostoru konceptů jsou uchovávány v anglickém jazyce. Navázaný obsah na prostor konceptů může být vícejazyčný, kdy „obálka“ dokumentu je navázána na koncepty a jeho obsah je uložen uvnitř dokumentu ve více jazycích. Tento princip jsme vyzkoušeli na společném experimentu s kolegy z univerzity v Ankaře, Turecko.

Navržený princip navigace společně s algoritmem pro ohodnocení konceptů jsme prakticky otestovali v rámci vytvořeného adaptivního webového systému XAPOS určeného pro doménu výuky programovacích jazyků, který jsme přihlásili jako autorizovaný SW v roce 2011 za VŠB v RIV [77].

## Průvodce textem

Kapitola 3 je věnována sémantickému webu a jeho architektuře a jazykům které se používají. Krátce se zde také zabýváme principy porovnávání, ohodnocování, mapování a slučování ontologií a vytváření prostoru konceptů.

V kapitole 4 diskutujeme adaptivní web s důrazem na adaptivní techniky – adaptivní prezentaci a navigaci a modely adaptivních hypermédií. Podrobněji se zabýváme nejpoužívanějším modelem AHAM [21] a jeho částmi modelem adaptace a modelem uživatele. Prezentujeme různé způsoby adaptace a možné způsoby pro vytváření a prezentaci modelu uživatele. Krátce prezentujeme vybrané přístupy a systémy, které vznikly v této oblasti.

V kapitole 5 přinášíme naše původní publikované výsledky – algoritmus pro ohodnocení prostoru konceptů, který vychází z PageRank algoritmu. Podrobně popisujeme způsob ohodnocení konceptu v závislosti na ohodnocení jeho sousedů a jeho umístění v prostoru konceptů.

V kapitole 6 představujeme náš původní princip navigace, který využívá ohodnocený prostor konceptů. Navigace nad obsahem probíhá na základě vztahů mezi metadaty obsahu a na základě uživatelské interakce.

V kapitole 7 prezentujeme výsledky námi navrženého experimentu, který jsme provedli pro ověření výsledků z předchozích kapitol. Experiment jsme prováděli společně s kolegy z univerzity v Ankaře, čímž jsme získali možnost testování nad vícejazyčnými výukovými materiály v reálném vícejazyčném prostředí.

Příloha A obsahuje přehled publikací, které se vztahují k tématu disertace.

V příloze B popisujeme systém XAPOS, který jsme vytvořili v rámci naší práce a kde jsme implementovali uvedené principy.

Část I

**Současný stav poznání**



## Kapitola 2

# Adaptace, personalizace a základní pojmy

Množství informací se kterým se v dnešní době setkáváme je ohromné. Zpravodajské weby, sociální sítě, vyhledávací služby, weby pro sdílení videa, . . . používáme ve svém životě každý den. Nové weby a informace na nich se objevují jako houby po dešti. V podstatě by se dalo říct, že jsme zahlceni informacemi. Vzniká potřeba informace nějakým způsobem zpracovat, případně omezit jejich množství, které se k nám ve výsledku dostane. Většina uživatelů si postupně vytvoří seznam „svých informačních“ webů, které navštěvuje v podstatě pravidelně a čerpá z nich informace. Toto obvykle platí pro různé zpravodajské weby, weby obsahující informace, které pokrývají zájem uživatele. Pokud má uživatel zájem o nezahrnutou oblast, případně potřebuje vyhledat nové informace obrací se obvykle na vyhledávač. Vyhledávače vracejí na vstupní frázi velké množství setříděných výsledků. Zde se poprvé setkáváme s personalizací.

Adaptace a personalizace představují metodu nebo postup, jak uživateli vyhledávače, nebo nějakého systému nabídnout, takové informace, které pro něj budou přínosné, a které odpovídají jeho požadovanému cíli, tomu co chce nalézt, případně se naučit, vyzkoušet a podobně. Personalizace informací se začala nejprve rozvíjet na akademické půdě, kde v rámci informačních systémů obvykle pro nabízení kurzů, případně různých expertních či nápoředních systémů byla postupně uváděna v život. V dnešní době se s větší či menší personalizací setkáváme skoro v každém systému ačkoliv to nemusí být vůbec zřejmé. Většina vyhledávačů nabízí personalizaci založenou například na předchozích vyhledáváních případně na tom, které z výsledků dřívějších vyhledávání stejného dotazu zvolili uživatelé před námi.

Adaptace a personalizace může být prováděna jednak pro konkrétního uživatele – obvykle je tento uživatel do systému přihlášen a může být tedy identifikován, nebo pro skupinu uživatelů – uživatel je ztotožněn se skupinou na základě své volby, nebo výsledkem vyhodnocení dotazníku. Každý z přístupů v sobě skrývá výhody i nevýhody. Výhodou adaptace pro konkrétního uživatele může být nabízení přesnějších výsledků či informací. Nevýhodou mohou být extrémně specifictí uživatelé, které systém nedokáže „analyzovat“ a pracovat s nimi, a také anonymní uživatelé, pokud vůbec s nimi systém pracuje. Výhodou druhého přístupu adaptace nad skupinu uživatelů je větší množství informací a dat, která se o uživateliích souhrnně získají a s nimi je pak možné lépe skupinově adaptovat – například novému uživateli mohou být po zařazení do skupiny okamžitě nabídnuty požadované informace. Nevýhodou je, že uživatel může být do skupiny zařazen špatně, nemusí se jeho preference shodovat s preferencemi celé skupiny a tím pádem adaptace není odpovídající.

Adaptace může být prováděna několika způsoby. Můžeme adaptovat jednak dostupné množství – množství dostupných odkazů nebo adaptovat prezentaci dostupných informací. Základy oboru adaptace informací a adaptivních hypermédií můžeme umístit do roku 1999, kdy byly v [15] a následně v [16] představeny základní architektura a techniky adaptaci informací.

Důležitým aspektem adaptace a personalizace je možnost její automatizace. Aby bylo možné informace automaticky zpracovávat je nutné informace popsat způsobem, umožňujícím automatické zpracování. Problémem se zabývá oblast nazvaná sémantický web, která definuje způsoby jak informacím dát přesně definovanou sémantiku, která by vyjadřovala charakteristiky a vlastnosti popisovaných informací.

## 2.1 Základní pojmy

Na úvod definujeme základní pojmy, tak jak je budeme používat v dalším textu.

**Metadata** představují doplňkové informace přidružené k informacím. Jejich cílem je zjednodušit a zpřesnit popis informací, které reprezentují, tak aby je bylo možné automatizovaně zpracovávat. Používání metadat je klíčovým aspektem sémantického webu, kde právě metadata dávají informacím sémantiku. Metadata můžeme rozdělit na obsahově nezávislá metadata, které uchovávají informace nezávislé na obsahu a smyslu dokumentu a obsahově závislá metadata, které se zaměřují na sémantiku a informační hodnotu dokumentu samotného [71].



**Koncept** představuje základní jednotku informace v problémové doméně. Koncepty jsou používány jednak pro popis problémové domény, tak také v navigaci uživatele, kde se pomocí nich uchovává informace o znalostech uživatele. Příkladem konceptu z oblasti programovacího jazyka Lisp je například *Recursion*, *CloseFunction* *ConditionalExpression*. Koncept bývá nazván výstižným výrazem určujícím jeho obsah, obvykle bývá popsán anotací a doplněn množinou klíčových slov.

**Prostor konceptů** (CS, concept space) nebo také ontologie, představuje množinu konceptů s jejich definovanými vazbami. Koncept může mít jednu či více vazeb různých typů na ostatní koncepty. Prostor konceptů si můžeme představit jako multidimenzionální graf s orientovanými hranami. Každý typ propojení mezi koncepty přidává do prostoru konceptů další rozměr, kde hrany odpovídající typu propojení jsou součástí tohoto rozměru. Příkladem vazby mezi koncepty může být například *subclassOf*, *relatedTo*, *definedBy* a jiné.

**Dokument** představuje základní datovou jednotku, která je popisována pomocí konceptů. Obvykle dokumentem chápeme datový soubor jako je například HTML či PDF soubor, ale dokumentem v pojetí sémantického webu může být i více souvisejících HTML stránek společně s grafikou, styly a dalšími multimediálními soubory či naopak jen odstavec.

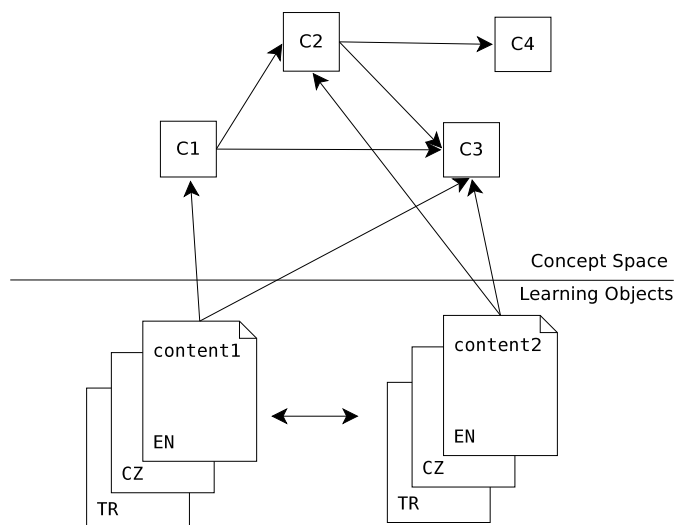
**Výukový objekt** (LO, learning object) představuje speciální případ dokumentu, který se používá v doméně výukových hypermédií. Výukový objekt se obvykle skládá z výukového materiálu společně s multimediálním obsahem případně příklady, které jsou prezentovány uživateli. Příkladem výukového objektu může být kapitola z kurzu Lisp *Základní typy* nebo *Vícenásobná rekurze*. Výukový objekt může být popsán pomocí existujících standardů jako je LOM<sup>1</sup> (datový model pro popis výukových objektů) či SCORM<sup>2</sup> (Sharable Content Object Reference Model – kolekce standardů pro webový elearning). Jejich popis ve standardizovaném formátu umožňuje sdílení výukových objektů mezi více systémy.

**Prostor dokumentů** je množina všech dokumentů, se kterými systém či nástroj pracuje. Dokumenty v prostoru dokumentů mohou mít definovány vazby na jiné dokumenty z prostoru.

---

<sup>1</sup><http://ltsc.ieee.org/wg12/>

<sup>2</sup><http://www.adlnet.gov/capabilities/scorm/scorm-2004-4th>



Obrázek 1: Propojení prostoru konceptů a obsahu

**Propojení prostoru konceptů a prostoru dokumentů** představuje definování vazeb mezi prostorem konceptů a prostorem dokumentů (viz obrázek 1). Každý dokument může mít přiřazenou množinu konceptů, které jej definují a popisují. V některých systémech či metodách tato množina bývá jen jedna v jiných bývá rozdělena na množinu prerekvizit (toho co potřebujeme znát předtím) a množinu výstupů (toho co se v dokumentu dozvíme).

## Kapitola 3

# Sémantický web

Myšlenka sémantického webu je označována jako rozšíření webu, v němž informace mají přidělen dobře definovaný význam lépe umožňující počítačům a lidem spolupracovat. Tim Berners-Lee představil ideu sémantického webu v roce 2001 v [9]. Obvyklým způsobem, jak rozšířit stávající data o sémantické informace, je doplnění stávajících dat o metadata, která popisují charakteristiky a vlastnosti daných dat. Přidání metadat usnadňuje automatické strojové zpracování dat.

Přidání sémantiky do dokumentů přináší nejen možnost automatizovaného zpracování informací, ale také možnosti vytváření nových propojení mezi dostupnými dokumenty, automatizované vytváření slovníků popisující problémové domény nebo odvozování nad metadaty. V neposlední řadě se metadata používají v sémantických vyhledávacích, které se zaměřují na zlepšení přesnosti hledání pochopením záměru uživatele s přihlédnutím k významu zadaného vyhledávacího dotazu v problémové doméně vyhledávání.

Přidání metadat v prostředí Internetu, kde se jako primární jazyk pro prezentaci informací používá jazyk HTML se začaly využívat tagy meta. Protože toto řešení nebylo úplně dostačující, začaly se objevovat další možnosti a specifikace jako je například „Semantic HTML”, kde se správným použitím tagů a stylů klade důraz na sémantiku dokumentu. Pro vlastní prezentaci dokumentu uživateli se pak doporučuje používat kaskádové styly (CSS). Typickým příkladem, který je doporučován také ve specifikaci HTML je použití tagu *em* (zvýrazněný text) místo tagu *i* (text kurzívou), všude tam kde má být text zvýrazněn a formu zvýraznění definovat v CSS. Doporučením je také definovat jména tříd v HTML podle jejich významu.

Dalším příkladem rozšířením doplňujícím význam obsahu webu, které se ujalo a je používáno, jsou mikroformáty<sup>1</sup>. Mikroformáty staví na využití existujících HTML tagů s využitím názvů tříd (atribut class). Mikroformáty vznikly z potřeby automatizovaně zpracovávat krátké datové bloky jako jsou například události, kontaktní informace, geografické informace a jiné. Cílem mikroformátů je datové bloky automatizovaně zpracovávat a přitom zachovat čitelnost pro člověka. V současné době existuje řada fakticky definovaných a funkčních mikroformátů pro definování událostí, kontaktních informací, audio a video obsahu, curriculum vitae, zpravodajských informací až po recepty. Každý z mikroformátů má definované názvy tříd, které se musí v HTML použít a jejich hierarchii. Příklad převodu HTML do mikroformátu můžeme vidět v následujícím příkladu.

Máme definováno:

```
<div>
  <div>Zdenek Velart</div>
  <div>Společnost</div>
  <div>123-456-789</div>
  <a href="http://example.com/">http://example.com/</a>
</div>
```

Po převodu do mikroformátu hcard, který slouží k definici vizitek vznikne:

```
<head profile="http://www.w3.org/2006/03/hcard">
  ...
</head>
...
<div class="vcard">
  <div class="fn">Zdenek Velart</div>
  <div class="org">Společnost</div>
  <div class="tel">123-456-789</div>
  <a class="url"
    href="http://example.com/">http://example.com/</a>
</div>
```

Mnoho z těchto formálních doporučení, které byly uvedeny dříve, se dostalo do specifikace HTML5<sup>2</sup>, která představuje další krok ve vývoji HTML jazyka. Tato specifikace navíc obsahuje i nové tagy jako například article, footer, section a další které pomáhají s definicí

<sup>1</sup><http://microformats.org/>

<sup>2</sup><http://www.w3.org/TR/html5/>

struktury dokumentu. Společně se specifikací HTML5 se vytváří specifikace Microdata<sup>3</sup>, která definuje obdobnou funkčnost jako mikroformáty.

Kromě doplnění sémantických informací do existujících HTML dokumentů na webu se pro zápis a sdílení sémantických informací využívají ontologie nebo topic maps zapsány pomocí jazyků RDF, OWL a XTM. Jejich cílem je definovat společný standardizovaný formát pro zápis a výměnu sémantických informací mezi systémy a nástroji, které se sémantickými informacemi pracují.

Ačkoliv si toho vůbec nemusíme být vědomi, tak se sami také můžeme podílet na automatizovaném vytváření sémantiky dostupných informací. Může se například jednat o jednoduché ohodnocení výsledku vyhledávání. Buďto explicitně určíme, že výsledky, které jsou nám prezentovány odpovídají námi položenému dotazu, či třeba jen zvolením výsledku vyhledávání, označíme tento jako odpovídající, což vyhledávač může vyhodnotit. Velmi oblíbenou součástí života se staly různé sociální sítě. Jejich prostřednictvím s ostatními „přáteli“ sdílíme například své fotografie či videa, kde určujeme osoby zde přítomné, místa kde byly fotografie či videa pořízena. Tímto velmi přesně definujeme sémantiku těchto informací. Tato oblast je označována jako sociální sémantický web či také jako folksonomie.

Pokud zůstaneme u příkladu fotografií, tak mnoho moderních fotoaparátů dnes obsahuje GPS chip, který zaznamenává do fotografie přímo polohu, kde byla vytvořena. Nástroje pro zpracování a prezentování fotografií umí s tímto údajem pracovat a mohou tedy automatizovaně nabídnout doplňující informace o daném místě případně nabídnout fotografie vytvořené jinými autory pro porovnání.

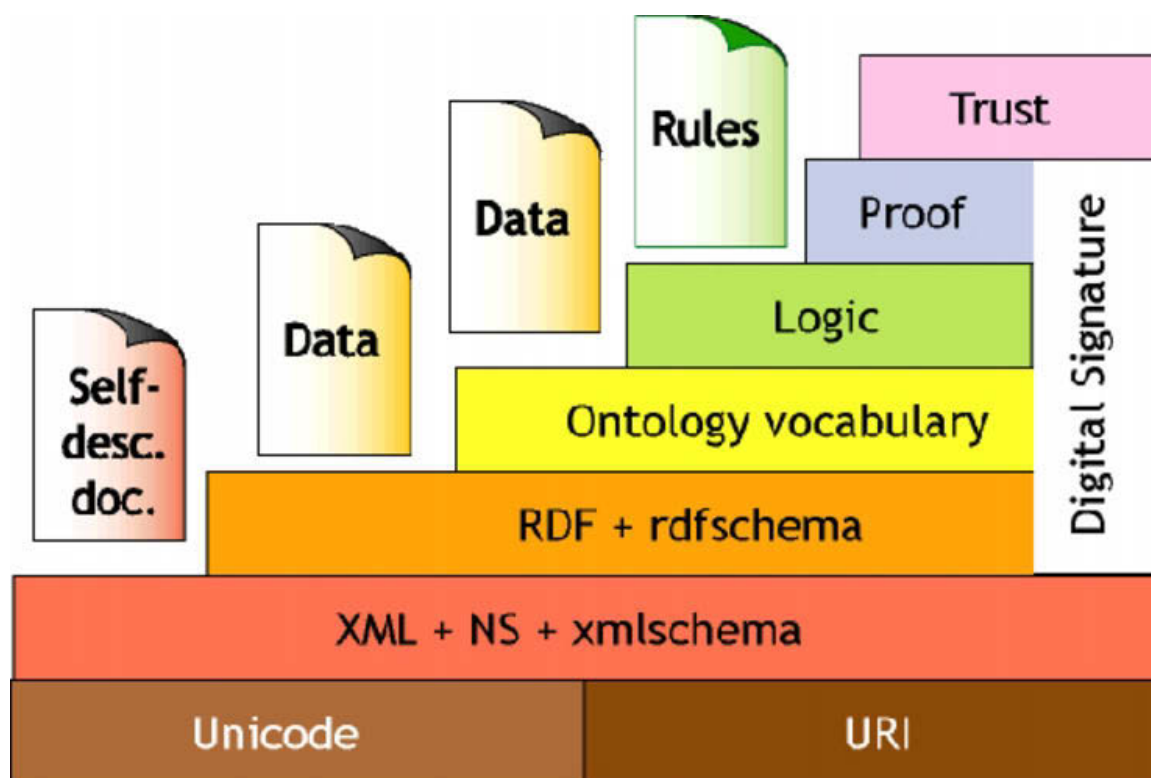
### 3.1 Architektura sémantického webu

V [10] byla představena formální architektura sémantického webu (viz obrázek 2), která slouží k transformaci stávajících zdrojů na zdroje sémantického webu a k zápisu nových zdrojů. Jednotlivé vrstvy mají předem určenou roli. Unicode vrstva představuje definici a způsob zápisu mezinárodních znakových sad. URI vrstva slouží k jednoznačné identifikaci a adresaci zdroje v prostředí webu. XML + NS + XML Schema vrstva slouží jako prostředek pro fyzický zápis metadat ve standardizovaném jazyku a umožňuje snadnější kooperaci mezi jednotlivými standardy postavenými na XML. RDF + RDFSchemata vrstva slouží k zápisu

---

<sup>3</sup><http://www.w3.org/TR/microdata/>

metadat ve formě výrazů a vytváření hierarchií z nich. Vrstva ontologického slovníku představuje vyšší možnost expresivního vyjádření a umožňuje vytvářet komplexnější a složitější vztahy mezi koncepty.



Obrázek 2: Architektura sémantického webu [10]

### 3.2 Metadata a ontologie

Pro zápis metadat v prostředí sémantického webu se používají ontologie a jazyky pro jejich popis. Pojem ontologie pochází z filozofie. Asi nejznámější a nejpoužívanější definice ontologie z hlediska počítačového vnímání a sémantického webu je obsažena v [69], která navazuje na [30].

*Ontologie je formální a explicitní specifikace sdílené konceptualizace.*

Pokud rozložíme definici na jednotlivé části, tak definice říká že ontologie představuje formální – s přesně definovanou strukturou, kterou lze zpracovávat strojově – explicitní

popis použitých konceptů a jejich vztahů, který je sdílen a představuje zjednodušený popis reálného světa, který reprezentuje.

Ontologie používají pro popis problémové domény a konceptů následující terminologii:

- *Třídy (classes)* – třídy objektů problémové domény, množiny, kolekce, koncepty.
- *Individua (individuals)* – instance objektů a objekty.
- *Atributy (attributes)* – vlastnost, charakteristiky které objekty mohou mít. Instance obsahují konkrétní hodnoty atributů.
- *Relace (relations)* – definují relace mezi třídami a instancemi.
- *Funkce (functions)* – komplexní struktury vytvořené z relací. Mohou být použity ve výrazech místo individuí.
- *Restrikce (restrictions)* – omezení, která musí být splněna, aby mohlo být tvrzení přijato jako vstup.
- *Pravidla (rules)* – výroky ve formě if-then vět, které popisují logické závěry, které mohou být odvozeny z tvrzení.
- *Axiomy (axioms)* – tvrzení (včetně pravidel) v logické formě, které tvoří popisovanou problémovou doménu.
- *Události (events)* – události, kdy jsou změněny hodnoty atributů nebo relací.

*Doménové ontologie* specifikují model konkrétní problémové domény, která představuje kousek reálného světa. Význam jednotlivých konceptů je tedy úzce svázán s konkrétní problémovou doménou. To znamená, že koncepty definované v jedné doménové ontologii mohou mít v jiné doménové ontologii odlišný význam ačkoliv mají stejný název.

Naproti tomu *referenční ontologie* (upper ontology, top-level ontology, foundation ontology) představuje model obecných objektů, které mají stejný význam ve větším množství doménových ontologií. Představují v podstatě obecný slovník definic konceptů a jejich popisu. Mezi nejznámější referenční ontologie můžeme zařadit například Dublin Core<sup>4</sup>, General Formal Ontology (GFO)<sup>5</sup> nebo Suggested Upper Merged Ontology (SUMO)<sup>6</sup>.

<sup>4</sup>[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=52142](http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=52142)

<sup>5</sup><http://www.onto-med.de/ontologies/gfo/>

<sup>6</sup><http://www.ontologyportal.org/>

## Jazyky

K tomu, aby bylo možné ontologii sdílet a zpracovávat, je nutné ji popsat pomocí obecně známých a definovaných pravidel. Pro tento účel vzniklo několik jazyků pro popis ontologie jako jsou DOGMA, KIF, Ontolingua, DAML, DAML+OIL, RDFS, OWL. V následujícím textu popíšeme některé z nich.

- RDF<sup>7</sup> představuje datový model pro popis objektů a relaci mezi nimi. RDFS představuje slovník po popis vlastností a tříd RDF zdrojů. Formální model používá RDF trojice jako abstraktní syntaxi pro RDF zdroje. RDF trojice se skládá ze subjektu, predikátu a objektu, kde subjektem může být URI reference nebo prázdný uzel, predikátem je URI reference na konstantu reprezentující binární predikát a objektem je opět URI reference, prázdný uzel nebo datová hodnota.
- DAML+OIL<sup>8</sup> – jazyk vznikl kombinací jazyků DAML (Darpa Markup Language) společně s jazykem OIL (Ontology Inference Layer nebo Ontology Interchange Language). Jazyk OIL je tvořen pomocí vrstev, kde každá vrstva přidává funkčnost a komplexnost k té předchozí. Jednotlivé vrstvy jsou tyto:
  - Core OIL – shoduje se s RDFS, s výjimkou některých částí.
  - Standard OIL – představuje jazyk určený k zachycení všech potřebných základních prvků, které mají dostatečnou vyjadřovací schopnost a jsou velmi dobře pochopitelné a umožňují přesné vyjádření sémantiky a dostatečné odvozovací schopnosti.
  - Instance OIL – představuje jednotlivé integrace. Instance OIL má stejné schéma jako Standard OIL. Instance jsou pak přímo popisovány pomocí RDF.
- OWL<sup>9</sup> – Web Ontology Language je značkovací jazyk vytvořeny konsorciem W3C k publikování a sdílení ontologií. OWL představuje slovníkové rozšíření jazyka RDF a je odvozen od DAML+OIL.

Jazyk OWL je z hlediska implementace tvořen třemi podjazyky:

---

<sup>7</sup><http://www.w3.org/RDF/>

<sup>8</sup><http://www.daml.org/language/>

<sup>9</sup><http://www.w3.org/TR/owl-features/>



- OWL Lite – umožňuje definování základních hierarchií s jednoduchými omezeními. Například OWL Lite podporuje omezení kardinality, ale pouze s hodnotami 0 a 1.
- OWL DL – je určena pro ty uživatele, kteří chtějí maximální vyjadřovací schopnosti při zachování výpočetní úplnosti (je zaručeno, že všechny závěry budou vypočteny) a rozhodnosti (všechny výpočty skončí v konečném čase). OWL DL obsahuje všechny konstrukty jazyka OWL, které ale mohou být použity pouze za určitých podmínek.
- OWL Full – je určen těm, kteří chtějí maximální vyjadřovací schopnosti a syntaktickou volnost RDF bez zaručení vypočtení výsledků.

Jazyky DAML+OIL a OWL jsou velmi příbuzné jazykům deskripční logiky s odpovídající terminologií. Jsou postaveny na individuích, které mají členství ve třídách a mají vztah k jiným individuím nebo datovým hodnotám skrze vlastnosti.

Přístup pro reprezentaci znalostí ve DAML+OIL a OWL nelze přímo reprezentovat v RDFS, pro jejich zápis je nutné v RDFS používat přímo RDFS konstrukce, kde požadovaná funkčnost existuje (například `rdfs:subClassOf` k vyjádření vztahů mezi třídami) a specifické konstrukce tříd a vlastností které rozšiřují funkcionalitu RDFS.

### 3.3 Porovnávání a ohodnocování ontologií

Ontologií může být mnoho, dokonce i takových, které popisují stejnou problémovou doménu. V ideálním případě by pro každou problémovou doménu existovala jedna všeobjímající ontologie. Protože tomu však není je nutné jednotlivé ontologie popisující stejnou či podobnou doménu porovnávat. Pro porovnání ontologií je nutné, aby bylo možné je srovnávat z hlediska stejných a rozdílných aspektů.

V [45] autoři představují množinu pravidel, pro porovnávání ontologií z hlediska lexikálního a konceptuálního. Výsledkem aplikování definovaných pravidel je zjištění, jakou měrou specifikace jedné ontologie odpovídá specifikaci druhé ontologii a opačně.

Autoři v [36] definují algoritmus SimRank, který porovnává podobnost objektů s ohledem na strukturální kontext ve kterém se objekt nachází a relací s ostatními objekty. Ideou algoritmu je, že „objekty jsou si podobné, když je na ně odkazováno podobnými objekty”.

Autoři v [3, 4] porovnávají ontologie na základě jejich atributů a objektů nacházejících se v instancích ontologie.

Ohodnocování ontologií slouží k snadnějšímu porovnání ontologií například z hlediska relevantnosti. Autoři v [1] ohodnocují ontologie ve smyslu párování tříd, hustoty, podobnosti nebo také srovnávání sémantické podobnosti a ohodnocení ontologie jako celku. Prezentovaná ohodnocovací metoda je využívána pro setřídění ontologií na základě jejich relevance k položenému dotazu.

### 3.4 Mapování a slučování ontologií

Mapování ontologií je morfismus mezi dvěma ontologiemi [39]. Při mapování a slučování ontologií můžeme narazit na problémy a rozpory mezi ontologiemi, které by mohly zhatit celý proces. Mezi obvyklé problémy na které lze narazit jsou problémy s nejednoznačností ontologií zanesené autory, kdy v ontologii se odráží znalosti a chápání problémové domény autorem. Dalším významným problémem se může ukázat použití odlišných konvencí při psaní ontologií, odlišných ontologických jazyků a jejich sémantiky či jiný význam sémantiky v ontologii. Problémem také může být použití výrazových prvků ontologie jako jsou negace, výrazy, spojení, průniky a další či použití stejných názvů označující odlišné koncepty či naopak různé názvy pro stejný koncept.

Principy, které se využívají při mapování a slučování ontologií můžeme rozdělit:

- Společná referenční ontologie a externí zdroje – staví na principu použití referenční ontologie, která definuje společný slovník nebo společný jazyk nebo na použití existujících databází terminologií a jejich definic. Příkladem je Suggested Upper Merged Ontology<sup>10</sup>(SUMO) [50], Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)<sup>11</sup>, S-Match [29]<sup>12</sup> nebo Wordnet<sup>13</sup>, který bývá označován některými jako ontologie jinými jako pouze lexikální databáze.
- Lexikální informace – ontologie se upravují z lexikálního hlediska tak že se provede normalizace řetězců (malá/velká písmena, úprava prázdných znaků, diakritiky, odstranění stop slov), porovnání řetězců (Hammingova vzdálenost, editační vzdálenost) či

---

<sup>10</sup><http://www.ontologyportal.org/>

<sup>11</sup><http://www.loa-cnr.it/DOLCE.html>

<sup>12</sup><http://semanticmatching.org/s-match.html>

<sup>13</sup><http://wordnet.princeton.edu/>

použití thesauru. Různí autoři následně provádějí slučování ontologií analýzou názvů konceptů a jejich definic za pomoci metod analýzy přirozeného jazyka [57].

- Struktura ontologie – použití struktury ontologií, toku informací, metrik pro porovnání OWL konceptů a podobnosti ontologií. Mezi příklady můžeme zařadit Information-Flow-based method for ontology mapping (IF-MAP) [38], Quick Ontology Mapping (QOM) [27], Chimaera [46], Prompt [54] a další.
- Uživatelský vstup – uživatel provádí základní mapování ontologií, poskytuje zpětnou vazbu navrhovaných projení, volí které akce se budou pro mapování a slučování používat a v jakém pořadí.

Vlastní metody, které provádějí mapování a slučování ontologií je možné rozdělit do následujících kategorií:

- Heuristické metody a metody založené na pravidlech – obvykle se jedná o metody pro analýzu struktury a lexikální analýzu.

V Prompt [54] autoři používají lexikální analýzu na identifikaci konceptů s podobnými názvy – provádí se například normalizace stringů, vyhledávání synonym, stejných částí řetězců apod. Takto nalezené koncepty představují první kandidáty na spojení. Pro koncepty, které jsou prohlášeny za stejné se následně provádí vyhledávání dalších vhodných kandidátů v jejich okolí.

Chimaera [46] identifikuje možné kandidáty na sloučení na základě lexikální podobnosti názvů, definic, akronymů nebo na základě stejných vlastností. Vyhledávání vhodných kandidátů je ovlivněno uživatelem zvolenou mírou, která určuje, do jaké hloubky a které z uvedených přístupů se použije například pro expanzi akronymů.

V metodě QOM [27] autoři možné kandidáty na spojení nacházejí pomocí výběru pevného počtu kandidátů; volbou kandidátů, kteří spolu sousedí v seřazeném seznamu jmen; oblastí, které sousedí s již nalezenými kandidáty; propagace spojení, kdy v dalším kroku se volí jen takoví kandidáti jejichž sousedi byli v předchozí iteraci spojení; použití hierarchie, kdy se postupuje od kořenů a porovnávají se koncepty v jednotlivých vrstvách. U nalezených kandidátů na spojení se provedou výpočty podobnosti jako například shodnost objektů, shodnost řetězců, podobnost řetězců, shodnost vlastností. Všechny vypočtené podobnosti jsou vloženy agregované podobnosti, na základě které se rozhodne o spojení konceptů.

Autoři v [29] definují metodu na propojení dvou ontologií vytvořením mapování mezi sémantický stejnými koncepty – autoři vytváří propojení výpočtem sémantických vazeb, které se zjišťují analýzou významu konceptu dle struktury a elementů ontologie.

- Analýza grafů – tyto metody berou ontologie jako grafy. Porovnáním stejných podgrafů nebo cest grafem se hledají stejné části, které lze spojit. Mezi příklady můžeme zařadit metodu porovnávání grafů Similarity Flooding [47] nebo Anchor-Prompt [55], která staví na myšlence, že pokud se najdou dva páry stejných konceptů a mezi příslušnými koncepty v ontologii existuje cesta, tak koncepty na této cestě jsou obvykle také podobné.
- Metody strojového učení – použití metod pro strojové učení, využití statistických informací z ontologií a jejich instancí.

V metodě GLUE [26] autoři využívají strojové učení pro vytváření mapování mezi ontologiemi na základě jejich instancí, kdy podobnost dvou konceptů A a B je založena na množinách jejich instancí a použití klasifikátoru s množinou A jak trénovací množiny pro určení, zda instance množiny B jsou také instancemi množiny A. Na základě získaných výsledků se vypočte společné rozdělení pravděpodobnosti pro koncepty které určuje zda koncepty budou či nebudou označeny jako stejné.

- Pravděpodobnostní a logické přístupy – využití výsledků heuristických a statistických metod a využití logiky, odvozování a dokazování pro hledání mapování ekvivalence, generalizace a specializace.

OMEN (Ontology Mapping ENhancer) [48] představuje přístup založený na Bayesovských sítích. Princip je založen na vytvoření Bayesovské sítě, kde uzel představuje mapování mezi koncepty nebo vlastnostmi vstupních ontologií a hrany představují vlivy mezi uzly. Vytvořením pravděpodobnostní tabulek danou síť založených na mapovaných konceptech a jejich susedech za použití definovaných meta-pravidel se připraví základ pro odvození pravděpodobnosti pro jednotlivé uzly. Uzly jejichž vypočtená pravděpodobnost je větší než daný práh jsou vybrány pro spojení.

## 3.5 Topic maps

Velmi blízkým příbuzným standardem k sémantickému webu je standard Topics maps<sup>14</sup> [53], který byl standardizován ISO<sup>15</sup>. Topic map reprezentuje informace za pomoci:

- topic – reprezentuje znalost vyjádřenou ve formě konceptu,
- asociace (associations) – reprezentuje propojení mezi jednotlivými topic,
- výskyty (occurrences) – reprezentují informační zdroje relevantní k danému topicu.

V rámci vývoje topic maps jako ISO standardu bylo vytvořeno několik datových formátů pro zápis. Mezi tyto formáty patří hojně používaný formát XTM (Topic Maps – XML Syntax), CXTM (Canonical XML Topic Maps format), CTM (Compact Topic Maps Notation) a GTM (Graphical Topic Maps Notation). V rámci ISO standardů byly vytvořeny také příbuzné standardy s Topic maps a to TMAPI (Common Topic Maps Application Programming Integrace) jako obecné API pro práci, TMQL (Topic Maps Query Language) pro vytváření dotazů a TMCL (Topic Maps Constraint Language) pro definování nebo testování sémantické validity mapy pro příslušnou doménu.

Topic Maps a RDF jako základní jazyk sémantického web sdílejí mezi sebou mnoho stejných vlastností a v mnoha ohledech se liší. Mezi nejdůležitější společné vlastnosti můžeme zařadit použití XML jazyka pro zápis sémantiky (i když existují i jiné možnosti zápisu, XML je preferovanou možností); možnost odvozování; pro obě oblasti existují jazyky pro dotazování a definování omezení.

Mezi hlavní rozdíly mezi ontologiemi a topic maps patří rozdílná úroveň sémantiky, která je zaznamenána. Každý ze standardů definuje svůj vlastní a navzájem odlišný model a v neposlední řadě i základní určení. Primárním určením RDF (a OWL) je uchování a zápis velkých datových celků a jejich automatizované zpracování pomocí umělé inteligence či agentů, kdežto TM je primárně zacíleno na možnost nalezení informací a znalostí člověkem.

---

<sup>14</sup><http://www.topicmaps.org/>

<sup>15</sup>[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=38068](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38068)

### 3.6 Sociální sémantický web, folksonomie

Jedním z rozšíření a nadstaveb sémantického webu je sociální sémantický web (social-semantic web, s2w) [13, 49], který využívá sociální interakce mezi uživateli webu pro vytvoření sémanticky bohatých znalostí. Principem je kolektivní znalost, která je založena na příspěvcích uživatelů jejichž kvalita se odvíjí od počtu uživatelů – čím větší počet uživatelů spolupracuje, tím kvalitnější znalost.

V dnešní době se s tímto pojetím sémantického webu setkáváme stále častěji v různých sociálních službách jako jsou Facebook, Google+, MySpace, LinkedIn, Flickr a jiné. V oblasti sociálního sémantického webu se uplatňují ontologie (ve smyslu semi-formální ontologie), taxonomie a folksonomie. Hlavní důraz klade sociální sémantický web na uživatelem vytvořenou sémantiku.

Folksonomie jsou jednou z možností, jak získávat základ pro sémantiku pro sociální sémantický web. Folksonomie představuje sociální tagování (social tagging) a podle autora v [81] představuje osobní neomezené tagování stránek a objektů pro vlastní vyhledávání informací, které splňuje následující charakteristiky – probíhá v sociálním prostředí; je vykonáváno člověkem; přidanou hodnotou je, že tagování přidává kontext a perspektivu a jsou zdrojem pro chybějící metadata; při tagování využívají lidé vlastní slovník.

### 3.7 Vytváření prostorů konceptů

Vytvořit prostor konceptů je možné několika možnými způsoby. Můžeme si pozvat experta na problémovou doménu, který bude schopen prostor konceptů vytvořit ručně, což je většinou časově (i finančně) náročné a v prostoru konceptů se odrážejí zkušenosti a preference experta. Případně můžeme zvolit některý z (polo)automatizovaných postupů. Druhý z přístupů sice úplně nevyklučuje přítomnost experta v procesu vytváření prostoru konceptů, minimalizuje se však jeho preference při tvorbě prostoru konceptů a potřebný čas. Expert zde spíše působí jako „korektor“ vytvořeného prostoru konceptů.

Autoři v [25] představují dva přístupy na vytváření prostoru konceptů. Prvním přístup je založen na automatickém vytvoření draftu prostoru konceptů, se kterou je následně možné pracovat. Tento přístup je založen použití heuristického přístupu pro získávání sémantických informací z HTML dokumentů se znalostí jejich struktury. Druhý přístup se zaměřuje na pomoc autorům při procesu vytváření prostoru konceptů dolováním dostupných znalostí z Wikipedie – autorům jsou navrhovány názvy pro nové koncepty společně s množinou

souvisejících konceptů z dané oblasti.

Autoři v [28] představují techniku poloautomatizovaného vytváření prostoru konceptů za pomoci technik pro vytěžování dat z dokumentů. V prezentovaném systému autoři používají techniky Latent Semantic Indexing pro získávání znalostí z textových dokumentů a K-mean shlukování pro rozdělení dokumentů na shluky, které obsahují pouze podobné koncepty dle použité metriky. Vlastní prostor konceptů následně vytváří autor, kde tyto techniky jsou využívány jako nápověda pro nové koncepty a pro automatické přiřazení dokumentů ke konceptům.

V [65, 66] autoři představují postup pro automatické získávání metadat pro vytváření konceptů a relací mezi nimi. Metoda je založena na zpracování textových dokumentů, vytvoření jejich vektorové reprezentace, očištění od stop slov a úpravy vah. Z připravených vektorů autoři získávají kandidáty na koncepty – pseudokoncepty, které jsou následně zdrojem pro vytvoření ontologie a vztahů mezi jednotlivými koncepty za použití metod pro zjištění podobnosti konceptů.

Autoři v [31] se zaměřují na tvorbu prostoru konceptů s využitím existujících HTML dokumentů. Pro zpracování dokumentů používají dva přístupy. První přístup provádí lexikální zpracování obsahu nadpisů (headings) HTML dokumentů a druhý přístup využívá strukturu HTML nadpisů pro identifikaci nových konceptů a jejich vztahů.

### 3.8 Shrnutí

V našem přístupu jsme jako hlavní cíl sledovali adaptivní personalizovanou navigaci postavenou nad prostorem konceptů. Pro realizaci experimentu, kterým jsme ověřovali vlastnosti našeho navigačního principu jsme použili prostor konceptů, který byl ovšem ručně vytvořen expertem. Prostor konceptů byl vytvořen pro sadu textových dokumentů z odpovídající aplikační domény, konkrétně šlo o dokumenty z domény programovacího jazyka C++ a dokumenty z domény programovacího jazyku Lisp [61, 62]. Souběžně jsme pracovali na poloautomatizovaném vytváření prostoru konceptů z textových dat [64], tyto naše první publikované výsledky v oblasti inženýrského vytěžování dat a poloautomatizované tvorby prostoru konceptů jsou velmi příslibné.





## Kapitola 4

# Adaptivní web

Adaptivní hypermédia a web mají velmi blízko k sémantickému webu. Adaptivní web se zaměřuje na navigování a personalizování nabízených informací. Aby to bylo možné provádět automatizovaně, je nutné aby systémy, které budou s informacemi pracovat, je uměly také vyhodnotit – znaly jejich sémantiku.

Adaptivní hypermédia se zrodila jako oblast v 90. letech minulého století, kdy byly poprvé představeny základní myšlenky, techniky a první modely. V práci [15] a [16] se můžeme poprvé setkat a seznámit se základními přístupy a technikami pro adaptaci obsahu a adaptaci navigace.

V průběhu následujících let vzniklo mnoho přístupů, formálních metodologií a modelů definující základní součásti, které by adaptivní systém měl obsahovat. Známý a často citovaný model AHAM [21] definuje tři základní součásti, které by měl každý systém obsahovat – model domény, model uživatele a model adaptace. Bylo navrženo a implementováno mnoho systému experimentálních či reálně nasazených do používání. Nejširší využití si adaptivní hypermédia našla v oblasti výuky, kde se adaptivní systémy využívají pro výuku programování, SQL, pro přístup k výukovým příkladům, nabízení výukových kurzů, doporučování literatury a v různých dalších směrech.

Při vytváření a implementování adaptivních systémů se největší míra věnuje modelu uživatele, který jednak uchovává informace o uživateli a jednak představuje – společně s modelem domény – základ pro adaptaci. Protože model uživatele je velmi komplexní vzniklo mnoho způsobů jak jej definovat – překryvný model, stereotypový model či sdílený model – a jak jej reprezentovat.

Důležitým krokem celého procesu adaptace, a jak se také ukázalo krokem v podstatě

nejtěžším, je první krok procesu v situaci kdy se uživatel poprvé přihlašuje (pokud není anonymní) do systému a chce s ním začít pracovat nebo se spouští nový systém. V této chvíli nastává takzvaný cold start problém, který se řeší například formou testovacího provozu či inicializace základních charakteristik uživatele pomocí defaultních hodnot nebo pomocí hodnot získaných od podobných uživatelů.

V závěru této kapitoly přinášíme přehled některých systémů a metodologií z oblastí adaptivních hypermédií a adaptace. Zmiňujeme se také o našich zkušenost s některými adaptivními systémy, jež jsme měli možnost otestovat a popisujeme adaptivní systém XA-POS, který jsme vyvinuli pro naše účely testování.

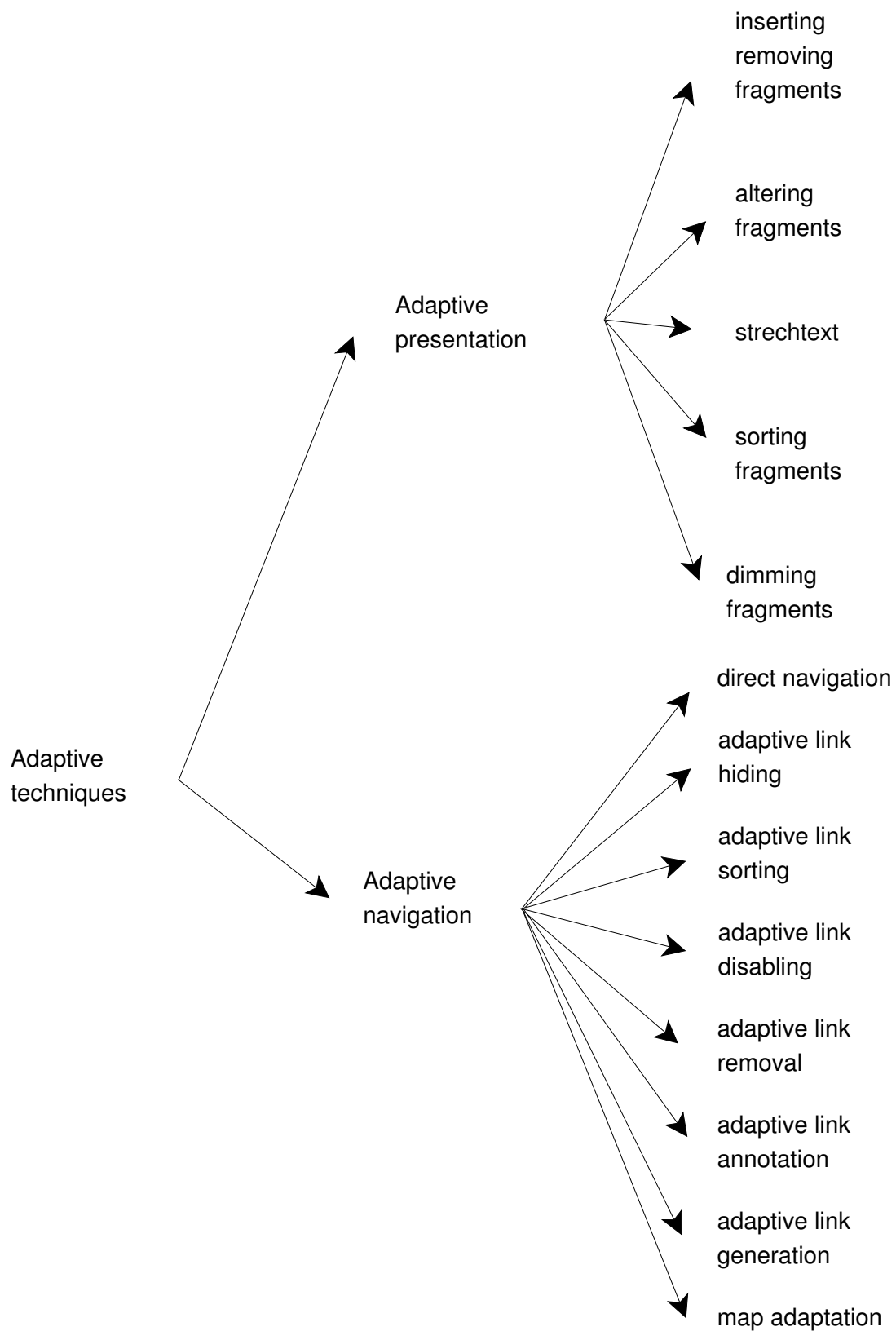
## 4.1 Adaptivní techniky

Adaptivní techniky slouží k adaptaci prezentovaného obsahu nebo jeho formy uživateli [15]. Adaptivní techniky rozdělujeme podle způsobu určení na *adaptivní navigaci* (adaptive navigation) a *adaptivní prezentaci* (adaptive presentation). Jejich rozdělení můžeme vidět na obrázku 3.

### Adaptivní prezentace

Tyto techniky pracují s informacemi ve formě dokumentu, případně ve formě stránky nebo její části, jako je například odstavec. Zaměřují se na přímou úpravu zobrazovaných informací uživateli a to jejich modifikací či určením, zda se budou zobrazovat a v jakém rozsahu. V následujících technikách budeme mluvit o fragmentu (místo dokumentu) jako jednotce, se kterou techniky pracují z důvodů zachování ustálených názvů technik:

- **Vkládání a odebrání fragmentů (inserting and removing fragments)** – jednotlivé fragmenty mohou být do zobrazovaného výsledku přidány na předem zvolené místo, případně mohou být odebrány pokud nesplňují podmínky pro zobrazení.
- **Záměna fragmentů (altering fragments)** – fragment obsahující podrobněji popis, definici, vysvětlení může být uživateli prezentován místo fragmentu, který obsahuje jen stručný popis nebo opačně. Obvykle se této techniky využívá v případech, kdy uživatel se s nějakou definicí či problémem setkává poprvé a je mu tedy prezentován rozšířený či obohacený fragment a následně již jen jeho kratší varianta.



Obrázek 3: Přehled technik adaptace

- **Roztahovací text (stretchtext)** – pouze krátká část (ukázka) nebo zástupce fragmentu je zobrazena uživateli. Uživatel má možnost obvykle pomocí kliknutí myši nebo najetím myši nad daný fragment zobrazit a prohlédnout si celou informaci daného fragmentu nebo naopak fragment schovat a zobrazovat jen jeho zástupce.
- **Třídění fragmentů (sorting fragments)** – fragmenty s informacemi jsou pro uživatele seříděny podle relevantnosti obsažených informací.
- **Zašedování fragmentů (dimming fragments)** – informace, které jsou pro uživatele nedůležité mohou být vizuálně odlišeny, například formou zašedění textu.

### Adaptivní navigace

Tyto techniky pracují s odkazy nebo se skupinou odkazů, kdy odkazy prezentované uživateli mohou být modifikovány či podmíněně zobrazovány. Odkazem v těchto technikách rozumíme nejen odkaz ve formě hyperlinku ale také způsob jakým může být uživatel mezi fragmenty navigován:

- **Přímé navigování (direct guidance)** – uživatel je přímo veden k požadovanému cíli. Obvykle jsou pro navigaci uživateli nabídnuta pouze tlačítka další a zpět.
- **Třídění odkazů (adaptive link sorting)** – prezentované odkazy jsou seříděny tak, aby se odkazy na relevantní informace vyskytovaly co nejvýše v seznamu. S touto technikou se můžeme běžně setkat ve vyhledávačích.
- **Schovávání odkazů (adaptive link hiding)** – odkazy v tomto způsobu prezentace jsou funkční, zobrazeny barvou okolního textu, tak aby na první pohled nebylo zřejmé, že se jedná o odkazy.
- **Znefunkčnění odkazů (adaptive link disabling)** – odkazy jsou do výsledné prezentace zahrnuty pouze jako text.
- **Odstranění odkazů (adaptive link removal)** – odkazy jsou zcela odstraněny z textu.
- **Anotace odkazů (adaptive link annotation)** – odkazy jsou vizuálně nebo textově odlišně zobrazovány podle jejich relevance, například oblíbeným způsobem je zobrazovat zajímavé odkazy zelenou barvou a irelevantní červenou. Poznamenejme,

že způsob anotace odkazů pouze barvou není slučitelný s WAI<sup>1</sup> (Web Accessibility Initiative). Proto je vhodné podle doporučení používat kombinaci textové a barvené anotace odkazů.

- **Vytváření odkazů (adaptive link generation)** – v rámci své práce může systém vyhodnotit zajímavé propojení mezi existujícími fragmenty a následně tuto informaci zahrnout jako odkaz do prezentace uživateli.
- **Adaptace mapy odkazů (map adaptation)** – grafická reprezentace problémové domény je prezentovaná uživateli. Uživatel se může navigovat přímo ke zvolenému cíli kliknutím na reprezentaci fragmentu v mapě. Uživateli může být zobrazována rozdílná mapa na základě jeho aktuálních znalostí.

## 4.2 Modely adaptivních hypermédií

Navigace uživatele k požadované informaci je nejdůležitějším úkolem každého systému postaveného na principu adaptivních hypermédií. K tomuto potřebují systémy sbírat a uchovávat informace o svých uživateli. Čím více a přesnější informace bude systém schopen o uživateli získat, tím přesnější navigaci bude schopen nabízet. V oblasti výukových systémů, sbírají systémy informace o uživateli ve formě naučených znalostí, učebního stylu, navštívených výukových materiálů nebo jejich kombinaci.

Většina systémů sdílí stejné či podobné charakteristiky a vlastnosti. Jejich porovnáním a zkombinováním byl navržen model, který definuje obecný model a jeho části, které by měl systém implementovat. Vnitřní uspořádání systémů může být postaveno na principu konceptu jako základní jednotky informace, které systémy využívají pro sledování charakteristik uživatele, pro popis doménového prostoru a pro vlastní navigaci. Koncept v mnoha systémech hraje dvojí roli v navigaci – výstup a prerekvizita. Koncepty, které jsou poprvé představeny ve výukovém materiálu jsou označovány jako výstupy a koncepty používané ve výukovém materiálu ale představeny již dříve jsou označovány jako prerekvizity.

Nejznámějším modelem popisujícím obecnou architekturu adaptivního systému je model AHAM (Adaptive Hypermedia Abstract Model) [21]. Model definuje tři základní části:

- **model domény** – reprezentuje problémovou doménu nebo oblast ve které systém pracuje. Problémovou doménu tvoří její popis ve formě metadat, prostoru konceptů

---

<sup>1</sup><http://www.w3.org/WAI/>

nebo ontologie [11], případně také samotný obsah (dokumenty), který bude prezentován uživatelům. Mnoho systémů využívá doménovou ontologii pro popis domény a ontologii struktury (concept structure ontology) pro popis relací mezi dokumenty.

- **model uživatele** – uchovává informace o uživateli systému a jejich chování v rámci systému. Tyto informace se následně využívají k adaptaci. Obvykle je uživatel do systému přihlášen, takže jeho identifikace je jednoznačná a systém pracuje přímo s jeho profilem. Některé systémy ovšem také umožňují práci anonymních uživatelů.
- **model adaptace** – specifikuje způsob jak vlastní adaptace probíhá. Pracuje současně s modelem uživatele, ze kterého získává data uživatele pro adaptaci a také s modelem domény, ze kterého se vybírají dokumenty, které budou uživateli zobrazeny. Zobrazení dokumentu uživateli a jeho reakce následně zpětně ovlivňují model uživatele.

### 4.3 Model adaptace

Model adaptace určuje, jakým způsobem bude adaptace probíhat. Vlastní adaptace uživatele může probíhat jedním či více možnými způsoby nebo jejich kombinací, které představujeme v následujícím textu.

#### Pravidlový přístup

Adaptace je založena na pravidlech která jejichž obvyklý formát je

```
if <condition> then <action>
```

kde podmínka *condition* může představovat externí akci, která musí být vykonána (jako například přístup na stránku) nebo otestování hodnoty atributu z doménového modelu či z modelu uživatele. Pokud je podmínka splněna, provede se akce *action*. Akce může představovat vytvoření události pro uživatele, na kterou bude moci reagovat či úpravu atributů v modelu uživatele či v modelu domény.

Pravidla mohou být definována v modelu domény jako součást dokumentů nebo samostatně. Po splnění podmínek je dokument zobrazen, přičemž jeden dokument může mít definováno více pravidel. Druhou možností je mít pravidla definována jako součást adaptačního modelu, kdy při interakci s uživatelem se vyhledávají pravidla, jejichž podmínka v aktuální chvíli může být vyhodnocena jako true a následně je vyvolána příslušná akce.

### Prerekvizity

Každý dokument v modelu domény má definovanou množinu prerekvizit. Obvykle se jako prerekvizity používají koncepty z prostoru konceptů v modelu domény. Informace o tom, co již uživatel zná se uchovává pro každého uživatele zvlášť v jeho instanci modelu uživatele, například jako množina dosažených znalostí.

V rámci jednoho kroku adaptace se provede porovnání toho, co uživatel má ve své množině dosažených znalostí, a co tedy zná s množinami prerekvizit dokumentů. Takové dokumenty u nichž průnik množiny prerekvizit a množiny dosažených znalostí uživatele odpovídá nastavené adaptaci jsou zobrazeny uživateli. Adaptace může být nastavena tak, aby všechny koncepty z množiny prerekvizit byly obsaženy v množině dosažených znalostí nebo aby jejich procentuální poměr odpovídal nastavenému prahu. Nastavení plně závisí na autorech adaptace a může být i víceúrovňové, kdy například při plném pokrytí je dokument označen jako doporučený, při 75 % jako vhodný a při nižším procentu jako nevyhovující.

### Filtrování na základě obsahu

Filtrování na základě obsahu (content-based filtering) který je předkládán uživateli je jednou z možností, jak redukovat množství informací, které uživatel dostává. Autoři v [72] vyhodnocují obsah dokumentu, který je pro uživatele zajímavý (ať rozhodnutím systému či explicitním určením uživatele). Dokumenty, které jsou vyhodnoceny na základě korelace k danému dokumentu jsou následně pro uživatele vybrány jako další vhodné. Tento přístup závisí na třech faktorech – obsah dokumentu, ohodnocení dokumentu získané od uživatele a filtrovacím algoritmu.

### Kolaborativní filtrování

Kolaborativní přístup (collaborative filtering) staví na tom, že uživatelé s podobnými zájmy či zaměřením budou preferovat obdobné informace [34, 70]. Uživatelé jsou rozdělení do skupin podle svých zájmů. Adaptace se provádí pro konkrétního uživatele ale s přihlédnutím na získané ohodnocení od všech uživatelů stejné skupiny. Pro správné fungování tohoto přístupu je nutné získávat zpětnou vazbu od uživatelů. Nevýhodou tohoto přístupu je, že uživatel může být zařazen do nesprávné skupiny na základě chybných počátečních údajů nebo uživatel ačkoli má stejné zájmy, nemusí mu nabízené informace vyhovovat.

## 4.4 Model uživatele

Uživatel je v modelu uživatele reprezentován různými charakteristikami a vlastnostmi. Tyto charakteristiky jsou jednak sbírány systémem na základě interakcí uživatelů se systémem anebo také pomocí dotazníků, které mají za cíl zjištění informací o uživateli. Každý systém může model uživatele implementovat odlišně a také schraňovat o uživateli jiné charakteristiky (podle použití terminologie charakteristiky, atributy nebo vlastnosti). První pokusy o sjednocení a standardizaci terminologie nacházíme v práci popisující meta-ontologii uživatelského modelu (User Model Meta-Ontology) [79].

Charakteristiky uživatelů je možné dělit z několika různých hledisek. V [51] je autoři dělí podle jejich změny vzhledem k systému:

- permanentní charakteristiky – nejsou závislé na systému a na uživatelově interakci se systémem. Obvykle se jedná o charakteristiky jako jsou osobní informace o uživateli (pohlaví, jméno, . . . ), zájmy, schopnosti a preference, které se dynamicky nemění.
- variabilní charakteristiky – dělí se na charakteristiky, které nezávisí na používání a interakci se systémem a mohou být změněny nezávisle na systému, jako je například věk a charakteristiky, které se mění v závislosti na používaném systému.

Další dělení charakteristik obdobné předchozímu vychází z předpokladu, jak často jsou charakteristiky uživatele měněny [40]:

- statické charakteristiky – charakteristiky uživatele, které se nemění nebo mění jen zřídka jako jsou jméno, datum narození, pohlaví a jiné. Tyto informace o uživateli jsou obvykle získány formou nějakého dotazníku či přenesením informací z jiných systémů.
- dynamické charakteristiky – obvykle tyto charakteristiky jsou svázány s uživatelovými znalostmi dané problémové domény, jeho cíli nebo k vyjádření charakteristik, které se mění častěji než statické charakteristiky. Tyto charakteristiky jsou obvykle získávány samotným systémem sledováním uživatelova chování a používání systému.

Charakteristiky uživatele mohou být dále děleny na základě příslušnosti k problémové doméně, na doménově nezávislé charakteristiky představující informace, které nejsou svázány s danou problémovou doménou a mohou být sdíleny mezi více systémy a doménově závislé charakteristiky, které jsou závislé na dané problémové doméně.



Osamostatněním doménově nezávislých charakteristik lze definovat a vytvořit obecný model uživatele, který bude obsahovat charakteristiky, označené v dřívějším textu jako doménově nezávislé, statické či permanentní. Jejich sdílením je možné usnadnit uživatelům nutnost vyplňovat základní informace v každém novém systému se kterým budou pracovat.

Pro vytvoření modelu uživatele může být použito jednoho či kombinace několika následujících přístupů.

### Překryvný model

Překryvný model (overlay model) je založen na principu že znalosti uživatele představují podmnožinu modelu domény, a tedy model uživatele lze reprezentovat jako vrstvu nad modelem domény [15, 42, 40]. Pro každého uživatele je vytvořena kopie prostoru konceptů z modelu domény, kde se zaznamenávají znalosti uživatele. Podle autorů v [15] se informace o znalosti konceptu může zaznamenávat ve formě binární, kvalitativní nebo kvantitativní.

Vytvořením kopie prostoru konceptů z modelu domény pro každého uživatele může představovat enormní paměťové a výpočetní nároky, proto autoři v [42] přišli s definicí *rozdílového modelu uživatele* (differential user model). Model uživatele neobsahuje v tomto případě kopii celého prostoru konceptů ale jen podmnožinu konceptů, které představují to, co uživatele zajímá případně co již zná.

Nevýhodou přístupu překryvného modelu je nutnost počáteční inicializace. Podle způsobu zaznamenávání informací o znalostech se jednotlivým konceptům musí přiřadit například 0 jako počáteční hodnota binární znalosti, nebo nějaká střední hodnota. Jedním z řešení tohoto problému je kombinace překryvného modelu s dalším přístupem.

### Stereotypový model

Je jedním z modelů, které se používají pro vytváření modelu uživatele (stereotype model). Stereotyp představuje kolekci charakteristik uživatele [58]. Uživatel je asociován s jedním či více stereotypy, které reflektují jeho znalosti, obecné charakteristiky, zkušenosti a další vlastnosti [40]. Stereotypy mohou být setříděny hierarchicky [51] a charakteristiky definované v rámci stereotypu se dědí. Když uživateli přiřadíme některý ze stereotypů automaticky získává všechny asociované charakteristiky a také zděděné charakteristiky.

Při vytváření nového systému při identifikování stereotypů autoři v [44] identifikovali tři důležité kroky, které je nutné vykonat, aby stereotypy byly správně identifikovány –

identifikace podmnožin uživatelů na základě podobných charakteristik, které se vztahují k navrhovanému systému; identifikování množiny klíčových charakteristik, které umožní asociování uživatele se skupinou; formalizace nalezených charakteristik skupin uživatelů dle reprezentaci v systému.

Podle způsobu, jakým designéři systémů zahrnují charakteristiky do stereotypu, označují autoři v [42] jako ruční přístup (handcrafted) nebo empirický přístup (empirically-based approach). Ruční přístup staví na znalostech a schopnostech designéra a na jeho pozorování skupiny uživatelů. Empirický přístup sbírá data o uživatelích a jejich akcích v systému a na tomto základě vytváří stereotypy.

Nevýhodou tohoto přístupu je, že uživatel může získat v rámci zděděných stereotypů charakteristiky, které jsou pro něj nepotřebné nebo dokonce nevhodné. Je to způsobeno tím, že model adaptace se zaměřuje a provádí adaptaci pro skupinu uživatelů ne pro uživatele jako jednotku. Definice uvedená v [41] říká, že stereotyp je staticky založené odvozování o skupině lidí. Na druhé straně výhodou tohoto modelu je, že asociováním uživatele s nějakými stereotypy usnadňuje počáteční inicializaci a následnou interakci uživatele se systémem.

## Sdílený model

Společně s předchozími modely se obvykle uvádí ještě sdílený model (shared model), který sám o sobě není modelem, ale představuje způsob jak sdílet charakteristiky uživatele mezi více systémy [7]. Cílem sdíleného modelu je propojit předchozí uvedené modely mezi sebou. Data, která se získají o uživateli v jednom systému, mohou být dostupná ostatním systémům.

Velkou výhodou uvedeného přístupu je, že uživatel nemusí v každém novém systému vyplňovat své základní charakteristiky, které se nemění nebo nemají přímou souvislost s problémovou doménou a službami nabízenými systémem, jako je například jméno, pohlaví a další. V rámci zjednodušení uživatelé mohou mít i stejné uživatelské jméno a heslo do těchto systémů. Tyto informace o uživateli jsou logicky uloženy na jednom místě a systémy k nim mají přístup.

Další výhodou je, že kromě základních charakteristik si systémy mohou vyměňovat informace o uživateli z hlediska jeho znalostí a preferencí. Toto však předpokládá, že si systémy budou navzájem rozumět. Různé systémy mohou používat různou terminologii a reprezentaci charakteristik. Pokud systémy používají stejnou terminologii, přesto může vzniknout problém, pokud systémy budou stejné charakteristiky chápat různě, různě vyhodnocovat

a různě s nimi pracovat. Systém tedy může mylně interpretovat nebo zapsat uživatelské znalosti a preference.

Další velkou otázkou, která se v dnešním světě čím dál tím více objevuje je problém bezpečnosti sdílení mezi systémy. Ne každému systému chce uživatel poskytnout své charakteristiky. Může například jednomu systému umožnit své charakteristiky pouze číst, jinému číst a zapisovat.

## 4.5 Re prezentace modelu uživatele

Model uživatele můžeme implementovat jako nativní formát definovaný autory systému nebo jako ontologii.

Nativním formátem se obvykle rozumí uložení dat modelu uživatele do relační databáze. V tomto případě je model uživatele reprezentován množinou tabulek. Charakteristiky uživatele mohou být v těchto tabulkách uloženy jako pojmenované sloupce, kde sloupec představuje jednu charakteristiku nebo jako uložené hodnoty například ve formě klíč – hodnota, kde klíč představuje charakteristiku. Použití relační databáze je pro implementaci odzkoušeným a úspěšným řešením, zaručující bezpečnost (dle použité technologie), stabilitu, výkon a zálohování.

Další možností nativního formátu představuje použití vlastního XML formátu pro uložení dat. XML je následně uloženo buďto na souborovém systému jako soubor, případně se používá XML databáze. V dnešní době nelze skoro zaznamenat rozdíly mezi relační a XML databází co se týče rychlosti či nabízené funkčnosti.

Oba tyto přístupy – relační databáze a XML formát – jsou v podstatě rovnocenné. Oba popisují uživatelské charakteristiky z hlediska prostého úložiště dat. Lepším přístupem, který umožňuje nejen ukládat charakteristiky, ale také s nimi pracovat na vyšší úrovni je použití ontologie. Tento přístup používá dříve popsané ontologické jazyky RDF a OWL pro popis charakteristik uživatele. Využitím ontologie pro popis charakteristik se zjednodušuje znovupoužití a sdílení charakteristik a harmonizace různých terminologií [43]. Přestože ontologie představuje dobrý základ pro sdílení charakteristik uživatele, strukturální a syntaktické rozdíly mohou být překážkou pro mapování mezi ontologiemi různých systémů. Autoři v [32] navrhuji obecnou ontologii uživatelského modelu – GUMO. Ačkoliv se jeví definování obecné ontologie pro model uživatele jako dobrý nápad, v prostředí tak různorodém jako je Internet, je téměř nemožné toto prosadit.

## 4.6 Cold start problém

Pokud se uživatel poprvé setká s novým systémem, neexistuje o něm v systému dostatek informací a charakteristik, aby systém mohl provádět personalizaci odpovídající profilu uživatele. Jednou z možností, jak bylo uvedeno dříve je získání charakteristik sdílením s existujícími systémy a jejich přebrání. Pokud však tato možnost neexistuje, vzniká cold start problém v těchto dvou případech:

- cold start problém nového systému – pokud se spouští nový systém, nejsou v něm žádné informace o žádném uživateli. Řešením problému může být například provoz systému v testovacím módu, ze kterého lze následně odvodit základní hodnoty, které budou uživateli nastaveny. Druhou možností je začít „od nuly“, kdy ovšem hrozí, že personalizace v systému nebude na počátku plnit svůj účel.
- cold start problém nového uživatele – pokud se uživatel setkává s novým systémem, který již nějakou dobu běží. Jednou z možností, jak již bylo uvedeno je použití stereotypů. Identifikováním uživatele se stereotypy, které jsou v systému uživatel, získá přidružené charakteristiky a jejich základní hodnoty. U tohoto řešení je nutné identifikovat stereotypy uživatele například pomocí dotazníku nebo testu případně jeho zaražením do skupiny uživatelů.

V systémech, které používají překryvný model nebo rozdílový model uživatele je řešením tohoto problému nastavení základních hodnot jednotlivým konceptům – u přístupů s binárním zaznamenáváním znalostí například nulu, u přístupů s kvantitavním zaznamenáváním znalostí například medián či průměrnou hodnotu.

## 4.7 Adaptivní systémy a metodologie

Adaptivní systémy a metodologie vznikaly a vznikají za účelem testování nových přístupů. Mnoho vlastností z testovacích systémů bylo následně uvedeno do praxe. Velké množství systému implementuje již dříve uvedený AHAM model, případně jej dále rozvíjí.

### **PersonalReader Framework**

Představuje prostředí pro implementaci a přístup k personalizovaným službám [33]. Pro středí je postaveno na třech základních typech služeb. Connector service pro posílání požadavků uživatele a odpovědí mezi uživatelským rozhraní a personalization service. Personalization service pro adaptaci, jako je například nabízení dokumentů, odkazování se na detailní informace, testy, dotazníky a jiné. Visualization service, která interpretuje výsledky personalization service a vytváří aktuální uživatelské rozhraní.

### **TANGRAM**

Je webový výukový systém [37] využívající ontologie pro reprezentování struktury výukových objektů, typů výukových objektů a výukové ontologie pro definování pedagogických relací mezi doménovými koncepty. Systém skládá personalizovaný výukový obsah pro každého uživatele na základě jeho znalostí, preferencí a výukového stylu. Výukový obsah se skládá z existujících výukových jednotek. Po uživateli je požadováno vyplnění dotazníku a absolvování testu za účelem stanovení výukového stylu a úrovně znalostí uživatele. Na základě zjištěných výsledků a za použití ontologií systém vytváří vizuální reprezentaci adaptovaného obsahu pro daného uživatele ve formě stromu odkazů na který jsou použity adaptační techniky anotace a schovávání odkazů.

### **AHA!**

Je webový adaptivní systém [20, 23, 22] který má pro každý fragment (dokument nebo část) autorem definovaná pravidla. Fragment může mít definovaných více pravidel. Na základě splnění podmínky pravidla se spustí akce asociovaná s pravidlem. Pravidla mohou rozhodovat o tom, zda daný fragment bude či nebude zobrazen nebo také o tom, zda daný fragment bude označen jako vhodný pro uživatele v daném čase.

Systém definuje a ukládá charakteristiky uživatelů ve formě atribut – hodnota. V modelu uživatele jsou pro každého uživatele uchovávány hodnoty definovaných atributů. Tyto atributy jsou pro uživatele vytvořeny při založení profilu s předem definovanými defaultními hodnotami. Pravidla u fragmentů se odkazují na tyto atributy, a to buď v podmínkové části, kdy jejich hodnoty musí odpovídat, aby akce svázaná s pravidlem byla spuštěna, či jako součást akcí, kdy jsou tyto hodnoty pro uživatele nastavovány. Typickým příkladem

atributu uživatele je čítač návštěv fragmentu, kdy může být definováno jednoduché pravidlo, jehož podmínka je nastavena jako true a v rámci akce se provede přičtení jedničky do atributu čítač návštěv u uživatele.

Pravidla v kurzu pro fragmenty definuje autor kurzu, který tak přenáší do kurzu svůj náhled na to jak by měl být kurz procházen a jakým způsobem by měla navigace fungovat.

## NavEx

Webový systém NavEx [17, 78, 80] je určen primárně pro nabízení výukových příkladů zdrojových kódů programovacího jazyka. Pro každý výukový příklad jsou definovány dvě množiny konceptů – prerekvizity a výstupy. Pro každého uživatele je v jeho profilu udržována množina naučených informací. Výukový příklad je pro uživatele označen jako vhodný v případě, že splní všechny jeho prerekvizity, tzn. že ve své množině naučených informací má všechny koncepty z množiny prerekvizit. Zobrazení informace o vhodnosti či nevhodnosti výukového příkladu NavEx provádí adaptací GUI a to změnou barvy a symbolu u příslušného výukového příkladu.

## ALEF (Adaptive LEarning Framework)

Je framework [67] pro tvorbu a nabízení kurzů. Systém se zaměřuje na jednoduchou tvorbu modelu domény, rozšířitelnou personalizaci a adaptaci kurzu a aktivní účast studenta ve výukovém procesu. V rámci frameworku ALEF bylo definováno schéma Adaptive Web-based Learning 2.0, které využívá všechny možnosti Webu 2.0

Framework definuje několik modulů, které představují základní bloky pro tvorbu adaptivní výuky, a to *content recommender*, *navigation tracer*, *annotator* a *collaborative question creator*. Content recommender ohodnocuje a nabízí studentům výukové objekty na základě vhodnosti asociovaných konceptů, obtížnosti výukového objektu a času od poslední návštěvy výukového objektu. Navigation tracer se zaměřuje na získávání informací o uživatelské pohybu v prezentovaném výukovém objektu, na základě získaných dat se uživatelům může upravovat grafická prezentace. Annotator umožňuje uživatelům vkládat do výukového objektu poznámky, případně odpovídat již na existující – poznámky uživatelů představují jednak důležitou zpětnou vazbu a jednak i zdroj doplňujících informací pro další uživatele. Collaborative question creator využívá princip annotatoru k vytváření speciálních anotací s otázkami. Uživatelé mohou vytvářet testovací otázky k danému výukovému materiálu a tak se podílet na tvorbě obsahu.

## LAOS

Je model [6, 19] pro tvorbu webových systémů za účelem jednoduššího vytváření kurzů z pohledu autora. LAOS staví na AHAM a rozšiřuje jej o další dvě vrstvy – model cílů a model omezení. Model cílů umožňuje užěji specifikovat co se má prezentovat a model omezení slouží k omezení prostoru, který se prohledává při adaptaci obsahu uživateli. Autoři navrhuji pro jednotlivé modely algebraické operátory a operace, které jsou určeny k automatizované tvorbě a kontrole vytvářeného obsahu. Příkladem systému, který implementuje tento model je MOT (My Online Teacher) [18], který využívá konceptové mapy pro návrh kurzu.

## OntoAWare

Je systém pro tvorbu, správu a nabízení výukového obsahu s využitím technologií sémantického webu [35]. Systém umožňuje poloautomatickou tvorbu výukových objektů z doménových ontologií (uloženy jako OWL model) a personalizovanou navigaci uživatele založenou na ontologii. Každý uživatel má možnost si zvolit v rámci systému možnost navigace. Systém umožňuje řízenou navigaci podle výukového plánu sestaveného učitelem, volného pohybu uživatele bez omezení a adaptivní řízení navigace, kde systém sleduje které koncepty již uživatel navštívil a předpokládá, že je již uživatel zná a znovu je nenabízí. Pro lepší adaptaci systém může pro uživatele připravit testy spouštěné před a po návštěvě výukového objektu.

## TM4L

Je prostředí pro tvorbu, udržování a používání výukových repozitářů postavených na ontologii [25]. TM4L staví na myšlence, že přístup k výukovému materiálu za pomoci konceptů implementovaných jako Topic map pomáhá k překonání propasti mezi studentem a požadovanou znalostí. Cílem práce autorů bylo umožnit větší znovupoužití dostupných výukových zdrojů za použití vytvořené předmětové ontologie v různých kurzech s různou strukturou ale stejným zaměřením. Výukový repozitář je definován pomocí tří vrstev, kde každá popisuje jiné vlastnosti – sémantická vrstva popisuje problémovou doménu z hlediska konceptů a jejich relací; vrstva zdrojů obsahuje odkazy na výukový materiál a další zdroje z dané problémové domény; kontextová vrstva obsahuje definice různých pohledů (kontextů) na daný repozitář například z hlediska požadovaného cíle nebo uživatele s využitím předchozích dvou vrstev.

### **AIMS (Agent-based Information Management System)**

Umožňuje kooperativní tvorbu [5, 6] výukového obsahu více autorům (plánování, vytváření, sdílení ideí a dat, kontrolu, řízení a dialog mezi autory). Systém se zaměřuje nejen na oblast tvorby výukového obsahu ve formě doménové ontologie, ale také umožňuje vizualizaci, grafickou navigaci a vyhledávání nad problémovou doménou s využitím definované doménové ontologie a také navigaci uživatele s využitím konceptů z doménové ontologie.

### **QuizGuide**

Adaptivní systém [68] určený k napomáhání uživatelům s výběrem nejvhodnějšího testu. QuizGuide používá adaptivní anotaci k zobrazení, které témata jsou pro uživatele aktuálně důležitá a která ještě vyžadují další znalosti. Vedle každého odkazu vedoucího na test QuizGuide zobrazuje adaptivní obrázek ve formě terče se šípem, který vyjadřuje míru znalosti uživatele a jeho výukový cíl. Počet šípů v terči označuje míru znalosti tématu uživatelem a intenzita barvy určuje míru relevance daného tématu k aktuálnímu výukovému cíli. Témata, která nejsou vhodná ke studiu jsou označena přeškrtnutým terčem. Systém rozlišuje čtyři úrovně znalostí tématu uživatelem – nepřipraven, důležité, méně důležité, nedůležité.

## **4.8 Shrnutí a vlastní řešení – XAPOS**

V rámci našeho výzkumu jsme otestovali adaptivní systém AHA!, který jsme po úpravách využili pro výuku našich studentů. Adaptace v adaptivním systému AHA! je postavena na pravidlech, která se definují pro jednotlivé fragmenty, jak jsou v AHA! nazývány stránky nebo jejich části. V modelu uživatele se uchovávají získané informace ve formě definovaných atributů, kde jsou pro každého uživatele ukládány hodnoty pro jednotlivé atributy.

Při praktickém testování [12, 73, 74] jsme provedli řadu úprav a optimalizací, které zvýšily výkon a odezvu adaptivního systému AHA!, které jsme i zpětně nabídli pro začlenění do hlavního kódu autorům systému. Výsledkem našeho testování a návrhu navigačního mechanismu bylo rozhodnutí navrhnout a vytvořit systém vlastní raději než upravovat existující.

Při tvorbě adaptivního systému XAPOS jsme využili zkušenosti z předchozího experimentu. Model uživatele je v systému XAPOS navržen jako překryvný model uživatele, kdy pro každého uživatele v systému je uchovávána kopie prostoru konceptů z modelu domény, kde se zaznamenávají znalosti a postup uživatele. Protože udržovat kopii pro každého uživatele se jevílo jako neefektivní zvolili jsme variantu rozdílového modelu uživatele. Pro



každého uživatele je uchovávána jen ta část prostoru konceptů, kterou již uživatel skutečně zná. Informace o znalosti konceptu uchováváme v binární formě, tedy pouze ve smyslu zná/nezná daný koncept.

Model domény je definován pomocí prostoru konceptů a pomocí dokumentů. Prostor konceptů představuje popis problémové domény z hlediska metadat. Dokumenty (výukové objekty) pro nás představují jednak výukový materiál, který je uživateli předkládán nebo také výukové příklady. V systému se s nimi pracuje stejně.

System XAPOS popisujeme podrobněji v příloze B.



## Část II

# Vlastní výzkum



## Kapitola 5

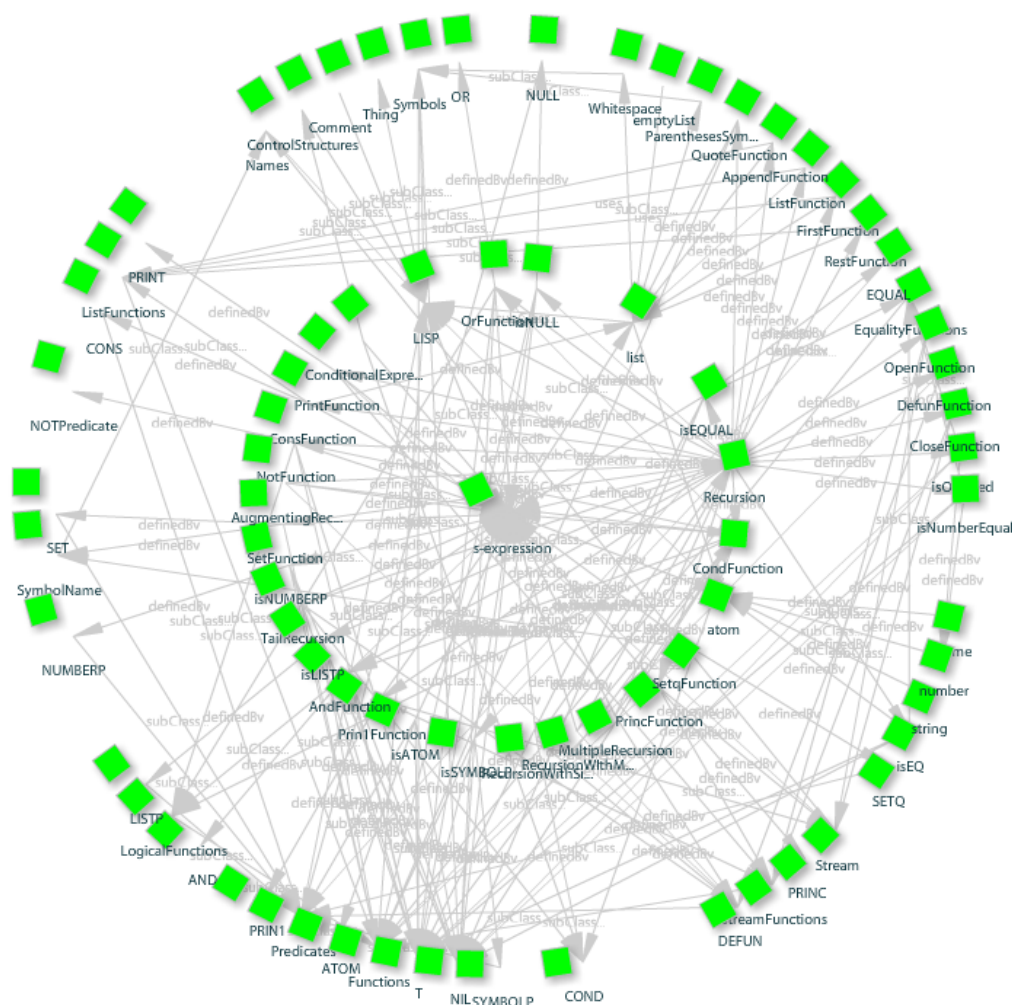
# Ohodnocení konceptů

Cílem kapitoly je zhodnotit a navrhnout způsob porovnání konceptů z hlediska jejich složitosti. Při stanovení ohodnocení konkrétního konceptu pracujeme s myšlenkou, které koncepty musím znát předtím, abych mohl pochopit tento koncept. Ohodnocení konceptu stanovíme na základě pozice konceptu v prostoru konceptů a na základě vazeb na další koncepty. Cílem tohoto ohodnocení je číselně vyjádřit náročnost konceptu.

Algoritmus pro ohodnocení konceptů [60, 61] vychází z PageRank algoritmu, který rozšiřuje a využívá pro výpočet ohodnocení konceptů z prostoru konceptů z hlediska jejich složitosti v závislosti na relacích s ostatními koncepty. Důležitým faktorem, který ovlivňuje náš algoritmus ohodnocení konceptů je stanovení množin relací mezi koncepty, které rozlišujeme. V rámci prostoru konceptů rozlišujeme dvě skupiny relací – strukturální a závislostní, které lze charakterizovat:

- strukturální relace – vyjadřují strukturu prostoru konceptů ve smyslu předeek – potomek, předchůdce – následník. Příkladem takového typu relací jsou relace `subClassOf`, `isA`, `partOf` a podobně.
- závislostní relace – definují závislosti mezi koncepty, které nejsou v přímé strukturální relaci. Obvyklými představiteli těchto relací jsou `definedBy`, `uses` a další.

Prostor konceptů je představován množinou konceptů a jejich vztahů. Podle typů relací tvoří vztahy  $n$ -rozměrný graf, kde  $n$  kde je počet typů relací v prostoru konceptů. V extrémním případě má každý koncept relaci všech typů se všemi ostatními koncepty. Na obrázku 4 je ukázka prostoru konceptů Lisp, který byl vytvořen pro účely testování.



Obrázek 4: Prostor konceptů programovacího jazyka Lisp

Prvním krokem ohodnocení konceptů je určení typů relací, které je následováno ohodnocením jednotlivých typů relací v celém prostoru konceptů. Dalším krokem pak je výpočet ohodnocení konceptů za použití vypočtených ohodnocení typů relací s přihlédnutím k sousedům konceptů a typům relací kterými je s nimi propojen.

Vypočtené ohodnocení konceptů je následně používáno v procesu navigace uživatele nad dokumenty. Toto ohodnocení se provádí pouze v procesu vytváření kurzu nebo v situacích, kdy dojde ke změně v prostoru konceptů. Protože toto se neděje často a pro každého uživatele je ohodnocení stejné je výhodné ohodnocení prostoru konceptů předpočítat.

## 5.1 PageRank

PageRank je algoritmus pro analyzování linků mezi stránkami. Algoritmus představený v [56] ohodnocuje každý prvek z množiny propojených dokumentů numerickou hodnotou představující jeho relativní důležitost v rámci množiny. Hodnota PageRank algoritmu je pro konkrétní stránku vypočítávána rekurzivně a závisí na počtu a hodnotách stránek, které na danou stránku odkazují. Stránka, která je odkazována stránkami s vysokou PageRank hodnotou sama získá vysokou hodnotu.

PageRank představuje rozdělení pravděpodobnosti, že osoba náhodně klikající na odkazy se dostane k jakékoliv stránce. Na začátku výpočtu je rozložení pravděpodobnosti mezi dokumenty rovnoměrné. Pro výpočet ohodnocení je nutné provést několik iterací nad množinou dokumentů. Pravděpodobnost je vyjádřena numerickou hodnotou mezi 0 a 1. Hodnota pravděpodobnosti 0.5 v PageRank algoritmu vyjadřuje 50% šanci, že osoba klikající na náhodný odkaz bude přesměrována na dokument s PageRank hodnotou 0.5.

Rovnice pro výpočet PageRank hodnoty je definována:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \quad (5.1)$$

kde  $p_i$  představuje jednotlivé stránky,  $d$  je damping faktor,  $M(p_i)$  je množina stránek, které se odkazují na  $p_i$ .  $L(p_j)$  je počet odkazů na stránce  $p_j$  a  $N$  je počet všech stránek. Damping faktor představuje pravděpodobnost, že osoba, která náhodně kliká na odkazy přestane. Obvyklou hodnotou, která se určuje jako damping faktor je okolo 0.85 [14].

## 5.2 Pojmy a značení

Základní pojmy a jejich případné defaultní hodnoty, které budeme používat v následujícím textu jsme definovali následovně:

$c_i$  – konkrétní koncept  $c_i$  patřící do prostoru konceptů – ( $c_i \in setC$ );

$r_i$  – konkrétní relace v prostoru konceptů, může se například jednat o relaci subClassOf, definedBy, uses;

$setC$  – označuje prostor konceptů, se kterým pracujeme;

$setR$  – představuje množinu relací, které se nacházejí v prostoru konceptů ( $r_i \in setR$ );

$N$  – je počet konceptů v prostoru konceptů  $setC$ ;

$d$  – damping faktor, tak jak jej definuje PageRank algoritmus. Zvolili jsme hodnotu 0.85 [14];

$c_i.rating$  – uchovává vypočtenou hodnotu ohodnocení konkrétního konceptu;

$c_i.related$  – definuje množinu konceptů, na které má koncept  $c_i$  vazby definované jako strukturální typ relace;

$c_i.countr$  – je počet odkazujících se strukturálních relací na daný koncept;

$c_i.depend$  – představuje množinu konceptů na které má koncept  $c_i$  relaci definovanou závislostním typem relace;

$r_i.rating$  – představuje vypočtené ohodnocení relace v prostoru konceptů.

### 5.3 Ohodnocení relací

Prvním krokem našeho algoritmu před výpočtem ohodnocení konceptů je výpočet ohodnocení relací. Ohodnocení relace je nutné z toho důvodu, že zavádíme do ohodnocení konceptu závislost konceptu na počtu a typech relací k ostatním konceptům.

Rozdělení relací do typů je prováděno autorem prostoru konceptů nebo tvůrcem kurzu, ve kterém prostor konceptů bude použit.

Ohodnocení relace určujeme jako její mohutnost v prostoru konceptů a vypočítává se následovně

$$r_i.rating = \frac{count(r_i)}{N} \quad (5.2)$$

pro každou relaci jako podíl výskytů dané relace ( $count(r_i)$ ) k počtu všech konceptů ( $N$ ) v množině konceptů.

Pro účely testování jsme použili prostory konceptů programovacích jazyků C, C++ a Lisp. Rozdělení relací na strukturální a závislostní typy bylo provedeno námi jako autory prostoru konceptů. V prostoru konceptů C jsou dvě relace a to strukturální relace *subClassOf* a závislostní relace *definedBy*. V prostoru konceptů C++ jsou tři relace a to strukturální relace *subClassOf* a závislostní relace *definedBy* a relace *uses*, které jsou také v prostoru konceptů Lisp. Vypočtené ohodnocení relací nalezneme v tabulce 1.



relace $r_i$	C	C++	Lisp
subClassOf	0.99727	0.97959	0.99065
definedBy	0.41363	0.46647	1.46728
uses	ndef.	0.32215	0.02803

Tabulka 1: Vypočtené ohodnocení relací v prostorech konceptů

### Zápis algoritmu

Zápis algoritmu ohodnocení relací vypadá následovně:

```

FOR each  $r_i$  IN  $setR$ 
   $r_i.rating = \frac{count(r_i)}{N}$ 
END FOR

```

## 5.4 Ranking algoritmus

Vytvořený Ranking algoritmus vychází z definice PageRank algoritmu. Definici algoritmu jsme rozšířili, tak aby vyhovovala potřebám ohodnocení prostoru konceptů, za situace kdy rozlišujeme dva typy relací v prostoru konceptů.

Ranking algoritmus vypočítává ohodnocení konceptu ve smyslu složitosti na základě jeho pozice v prostoru konceptů, relací s ostatními koncepty a jejich ohodnocení. Výsledné ohodnocení konceptů se provádí v několika iteracích, obdobně jako u PageRank algoritmu.

Výpočet ohodnocení konceptů se provádí pomocí následujícího vzorce pro každý koncept  $c_i$

$$\begin{aligned}
 c_i.rating = & \frac{1-d}{N} + d \times \left( \sum_{c_j \in c_i.related} \frac{c_j.rating}{c_j.countr} \right. \\
 & \left. + \sum_{c_k \in c_i.depend} c_k.rating \times r_k.rating \right)
 \end{aligned} \tag{5.3}$$

Rovnici lze rozdělit na několik samostatných částí. Část

$$\sum_{c_j \in c_i.related} \frac{c_j.rating}{c_j.countr} \quad (5.4)$$

určuje částečné ohodnocení konceptu založené na strukturálních relacích, kterými se koncept odkazuje. Každý koncept  $c_j$ , na který se odkazuje koncept  $c_i$  pomocí strukturální vazby přispěje k ohodnocení konceptu  $c_i$  poměrnou částí svého vlastního ohodnocení.

Druhá část rovnice

$$\sum_{c_k \in c_i.depend} c_k.rating \times r_k.rating \quad (5.5)$$

určuje ohodnocení konceptů z konceptů na kterých koncept  $c_i$  závisí. V tomto případě se sumuje ohodnocení konceptu  $c_k$  vynásobené ohodnocením relace  $r_k$ , kterou jsou tyto dva koncepty spojeny.

Finální ohodnocení konceptu se vypočte jako suma předchozích dvou částí s použitím damping faktoru a celkového počtu konceptů v prostoru konceptů.

### Zápis algoritmu

Celý algoritmus výpočtu ohodnocení konceptů vypadá následovně.

```

WHILE numIterations > 0 DO
  FOR each  $c_i$  IN setC
    rating = 0
    FOR each  $c_j$  IN  $c_i.related$ 
      rating +=  $\frac{c_j.rating}{c_j.countr}$ 
    END FOR
    FOR each  $c_k$  IN  $c_i.depend$ 
      rating += ( $r_k.rating \times c_k.rating$ )
    END FOR
     $c_i.rating = \frac{1-d}{N} + d \times rating$ 
  END FOR
  numIterations --
END WHILE

```

Pro správné fungování algoritmu ohodnocení je nutné určit základní ohodnocení konceptů, které je pro všechny koncepty v prostoru konceptů stejné a to  $\frac{1}{N}$ .

### Počet iterací

Pro dosažení správného rozložení ohodnocení mezi koncepty je nutné algoritmus provádět vícekrát. Počet iterací jsme stanovili pro prvotní pokusy jako  $\frac{N}{10}$ . Provedením experimentu jsme ověřili že pro ustálení ohodnocení v prostoru konceptů plně postačuje provést 10 iterací pro rozumný počet prvků v prostoru konceptů. Pro více iterací se ohodnocení konceptů mění minimálně. Experimentální data zobrazující koncepty a jejich ohodnocení, získaná pro prostory konceptů C, C++ a Lisp pro 1, 5, 10, 20, 100 a  $\frac{N}{10}$  iterací.

Ohodnocení vybraných konceptů pro prostor konceptů C se nachází v tabulce 2. Prostor konceptů C obsahuje 1100 konceptů a tedy  $\frac{N}{10} = 110$  iterací.

počet iterací	ExpressionStatement	UnaryExpression	CaseStatement
1	0.000861	0.002422	0.001672
2	0.000509	0.002025	0.000948
5	0.000376	0.001306	0.000648
10	0.000369	0.001269	0.000637
20	0.000369	0.001269	0.000637
100	0.000369	0.001269	0.000637
$\frac{N}{10} = 110$	0.000369	0.001269	0.000637

Tabulka 2: Ohodnocení jednotlivých konceptů pro různý počet iterací – C prostor konceptů

Ohodnocení vybraných konceptů z prostoru konceptů C++ se nachází v tabulce 3. Prostor konceptů C++ obsahuje 688 konceptů a tedy  $\frac{N}{10} = 68$  iterací.

počet iterací	ExpressionStatement	UnaryExpression	CaseStatement
1	0.001512	0.003948	0.002586
2	0.000439	0.007609	0.000946
5	0.000425	0.004502	0.000789
10	0.000423	0.004105	0.000779
20	0.000423	0.004105	0.000779
$\frac{N}{10} = 68$	0.000423	0.004105	0.000779
100	0.000423	0.004105	0.000779

Tabulka 3: Ohodnocení jednotlivých konceptů pro různý počet iterací – C++ prostor konceptů

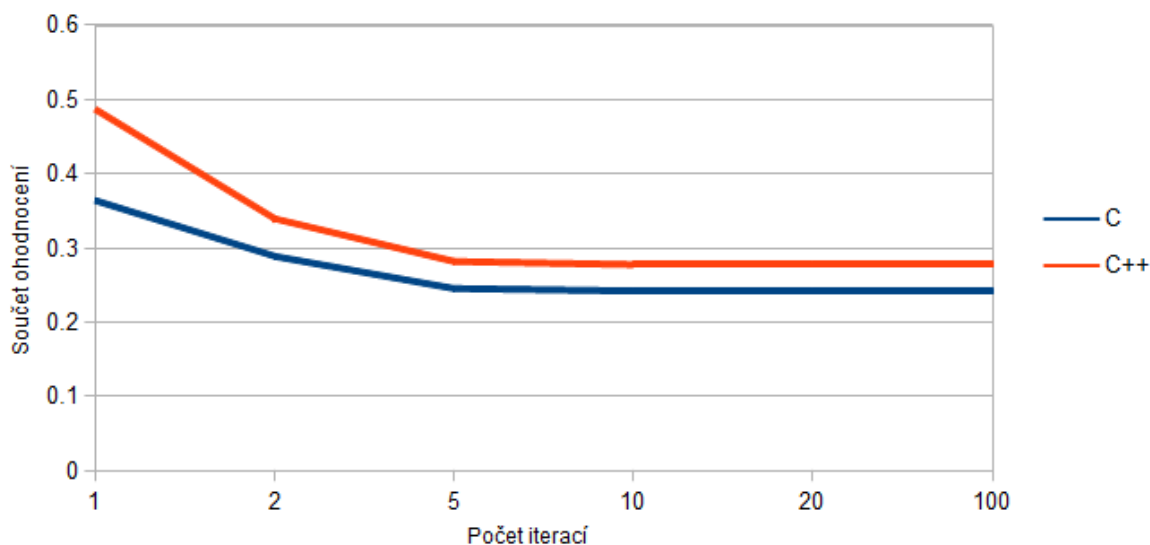
Jako příklad konceptů z prostoru C a z prostoru C++ jsme úmyslně zvolili koncepty,

kteřé mají stejné názvy. Podle znalosti problematiky a také podle volby stejného jména můžeme usoudit, že se jedná o stejné koncepty. Jejich ohodnocení je však odlišné a to z toho důvodu, že příslušný prostor konceptů obsahuje rozdílný počet konceptů a je odlišně uspořádán.

V případě prostoru konceptů Lispu odpovídá 10 iterací  $\frac{N}{10}$ , protože prostor konceptů obsahuje 107 konceptů. Příklad ohodnocení konceptů se nachází v tabulce 4.

počet iterací	CloseFunction	number	AugmentingRecursion
1	0.025848	0.073986	0.049614
2	0.006051	0.015504	1.657723
5	0.005821	0.013063	0.933437
$\frac{N}{10} = 10$	0.005818	0.013057	0.907895
20	0.005818	0.013057	0.907895
100	0.005818	0.013057	0.907895

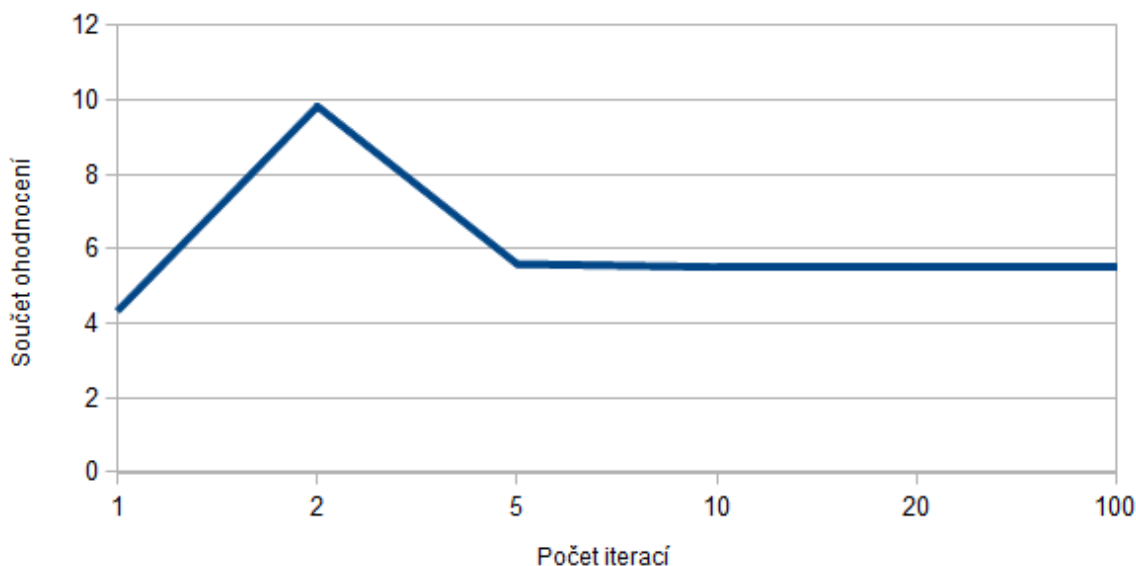
Tabulka 4: Ohodnocení jednotlivých konceptů pro různý počet iterací – Lisp prostor konceptů



Obrázek 5: Součet ohodnocení konceptů v prostorech konceptů C a C++

Na obrázcích 5 a 6 (vlevo součet ohodnocení pro celý prostor konceptů, dole počet iterací) je zobrazen součet ohodnocení konceptů pro prostor konceptů pro různý počet iterací. Grafy zřetelně ukazují, že po 10 iteracích se celkový součet ohodnocení konceptů

mění jen minimálně.



Obrázek 6: Součet ohodnocení konceptů v prostoru konceptů Lisp

Důvod proč je součet ohodnocení prostoru konceptů Lisp o řád vyšší než součet ohodnocení prostoru konceptů C a C++ je v množství konceptů a množství relací, které se v prostoru konceptů nacházejí a to především v závislostní relaci *definedBy* (viz tabulka 1), která má v prostoru konceptů Lisp značně vyšší ohodnocení. Její hodnota přesahuje 1, což znamená, že v daném prostoru konceptů je vyšší počet daných relací než počet samotných konceptů. Složitost konceptů je tedy vyšší, což můžeme také pozorovat na konceptu *AugmentingRecursion* v tabulce 4.

## 5.5 Shrnutí

Navrhli jsme rozšíření PageRank algoritmu reflektující relace různých typů a jejich ohodnocení pro výpočet ohodnocení konceptů z prostoru konceptů. Na dvou rozsáhlých (prostor konceptů C 1100 a prostor konceptů C++ 688 konceptů) a jednom menším (prostor konceptů Lisp 107 konceptů) jsme si experimentálně ověřili počet iterací, který je nutný pro ustálení ohodnocení konceptů. V další kapitole ukážeme, že získané hodnoty jsou dobře použitelné při personalizované adaptaci navigace postavené na použití prostoru konceptů a reflektující model uživatele.



## Kapitola 6

# Navigace

Navigace uživatele představuje nejdůležitější z funkcí adaptivních systémů. Každý z adaptivních systémů k tomuto úkolu přistupuje trochu odlišně dle zvoleného způsobu navigace. Jak bylo diskutováno v dříve v textu může být navigace postavena na přímé navigaci uživatele k požadovanému cíli na základě uživatelského požadavku; předchozích zkušenostech jiných uživatelů, kteří měli stejný či podobný cíl; na využití znalostních map, prostoru konceptů či ontologií pro určení znalostí a cest k nim vedoucích; na splnění prerekvizit pro to aby se určitá informace zobrazila.

Poskytovat uživateli informace, které pro něj mají význam, nebo které jej navedou k požadovanému cíli je i cílem našeho principu navigace [59, 61, 75]. Způsob navigace, který jsme navrhli je založen na použití prostoru výukových objektů a prostoru konceptů a jeho ohodnocení, které byly popsány dříve. V této kapitole se budeme držet označení výukový objekt místo označení dokument, tak jak byl definován v úvodní části „Základní pojmy“ (viz 2.1, strana 8).

Navigace uživatele probíhá ve třech krocích, kdy uživateli je nejprve prezentován nějaký výukový objekt společně s jeho výukovým materiálem. V dalším kroku uživatel provede ověření znalostí nebo pokračování na jinou stránku, kdy ve třetím kroku dojde k úpravě nastavení adaptace a k úpravě menu uživatele.

Všechny informace o uživateli, které jsou potřeba pro jeho úspěšné navigování se uchovávají v modelu uživatele, který slouží jednak jako zdroj pro adaptaci, současně také je zpětně ovlivňován volbami a chováním uživatele. V rámci modelu uživatele se sleduje jednak pohyb uživatele logováním návštěv jednotlivých výukových objektů, tak také dosažené znalosti uchováváním konceptů, které se uživatel již naučil.

## 6.1 Propojení prostoru konceptů a výukových objektů

Aby mohla navigace fungovat je nutné propojit prostor konceptů s prostorem výukových objektů. Pro každý výukový objekt je definována množina konceptů, které daný výukový objekt popisují (viz obrázek 1, strana 10).

Na rozdíl od jiných přístupů nerozlišujeme pro výukový objekt množinu prerekvizit a množinu naučené informace. Nerozlišujeme také míru příslušnosti daného konceptu k výukovému objektu – všechny koncepty, které jsou obsaženy v množině konceptů výukového objektu považujeme za stejně významné pro znalost výukového objektu.

Více výukových objektů se může vázat na stejné koncepty. Vyjadřuje se tím, že daný koncept může být obsažen ve více výukových objektech, bez ohledu na formu či rozsah takového výskytu.

### Příklad 6.1.1

Jako příklad uvádíme výukové objekty z vytvořeného zkráceného kurzu Lisp a jejich navázání na koncepty z prostoru konceptů Lispu.

výukový objekt	koncepty
Seznamy	APPEND, AppendFunction, CONS, ConsFunction, FIRST, FirstFunction, list, LIST, ListFunction, NIL, NULL, REST, RestFunction
Rekurze	AND, AndFunction, COND, CondFunction, OR, OrFunction, Recursion, s-expression
Rekurze na konci	AND, AndFunction, COND, CondFunction, OR, OrFunction, Recursion, s-expression, TailRecursion

Příklad ukazuje násobnost výskytu některých konceptů ve více výukových objektech, jak bylo předesláno. ■

Každý výukový objekt může obsahovat výukový materiál ve více jazycích. Uživateli je při navigaci zobrazován výukový materiál v jazyce, který si vybral. Navíc každý výukový objekt obsahuje anotaci, která ve zkratce popisuje jeho obsah. Anotace se používá pro snadnější orientaci uživatele v prezentované navigaci.



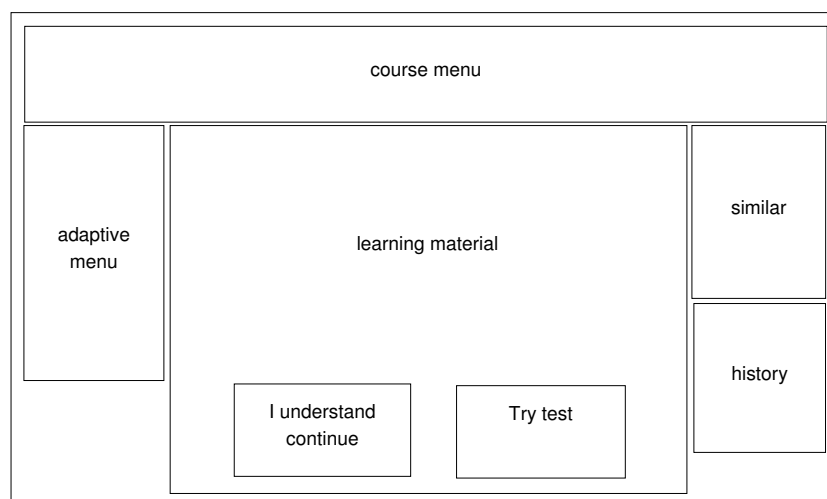
## 6.2 Model uživatele

Model uživatele, který využíváme v našem způsobu navigace vychází z překryvného modelu přesněji rozdílového modelu uživatele (viz 4.4). To znamená že pro každého uživatele je v rámci modelu uživatele uchovávána *množina dosažených znalostí* (achieved knowledge set). Znalost konceptu uživatelem je v našem způsobu navigace brána jako binární – znalost konceptu je dána tím, zda se koncept nachází v množině dosažených znalostí či nenachází. Koncept se v množině dosažených znalostí může nacházet nejvýše jednou, takže pokud jsou do množiny znalostí přidávány koncepty, které již tam uživatel má, nemá to na obsah množiny znalostí žádný vliv. Koncepty mohou být na základě adaptace do množiny naučené znalosti jednak přidávány ale také odebírány, například na základě výsledku testu.

Součástí uživatelského modelu je množina navštívených výukových objektů a log uživatele, do kterého se zaznamenávají informace o návštěvách výukových objektů. Toto je důležité jednak pro účely rekonstrukce uživatelské cesty kurzem, a také pro možnost zjištění stráveného času uživatele nad jednotlivými výukovými objekty.

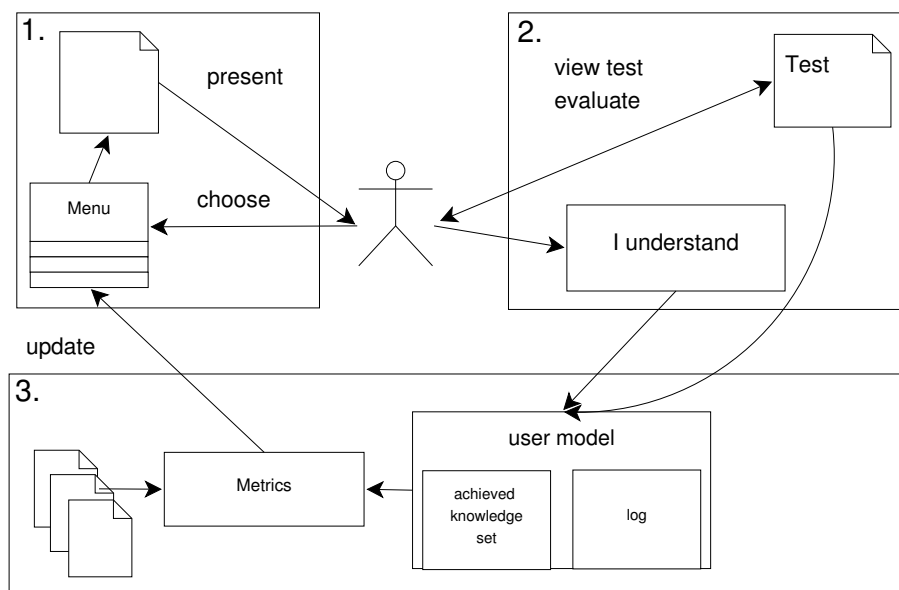
## 6.3 Navigace uživatele

Způsob navigace, který jsme navrhli je postaven na interakci uživatele a systému, který pro uživatele na základě modelu uživatele a dostupných výukových objektů připravuje vizuální rozhraní s adaptovanou navigací nad výukovými objekty.



Obrázek 7: Návrh uživatelského GUI

Uživateli je prezentováno GUI (viz obrázek 7), kde na levé straně je mu zobrazeno adaptované menu a na pravé straně dostupné podobné výukové objekty a jeho historie. V horní liště je mu pro přehlednost zobrazována kompletní navigace nad celým kurzem. Hlavní část obrazovky zabírá zobrazený výukový materiál. Na konci výukového materiálu se nacházejí dvě tlačítka a to „všemu rozumím, chci pokračovat dále” a tlačítko „Chci zkusit test”.



Obrázek 8: Způsob navigace uživatele nad obsahem

Navigaci a interakci uživatele můžeme popsat cyklicky pomocí těchto třech kroků (viz obrázek 8):

1. volba z menu a zobrazení výukového materiálu;
2. rozhodnutí o způsobu pokračování;
3. úprava modelu a menu;

### Volba z menu a zobrazení výukového materiálu

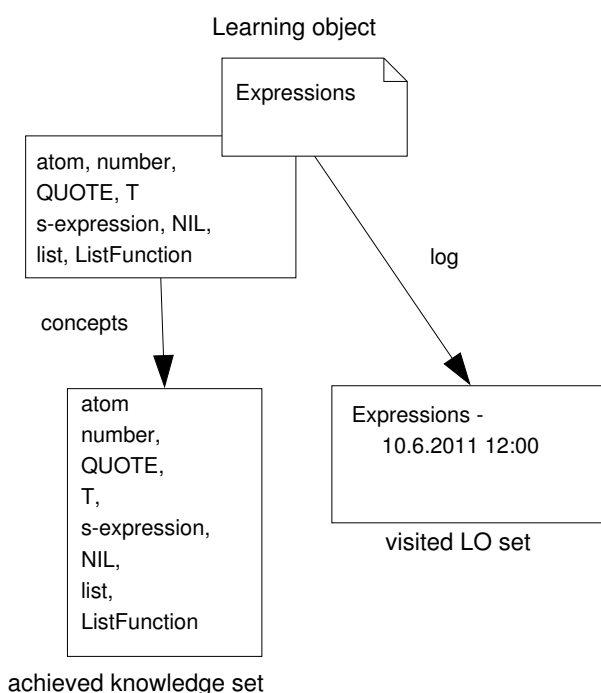
V prvním kroku navigace je uživateli prezentováno menu s výukovými objekty ze kterého si uživatel vybírá. V menu je zobrazeno více výukových objektů, které jsou zde reprezentovány svým názvem a anotací. Setřídění výukových objektů uživateli naznačuje, který výukový objekt je v dané chvíli pro něj nejvhodnější.

Podle toho, který výukový objekt si uživatel v menu zvolí je mu zobrazen výukový materiál daného výukového objektu. Výukový materiál může být představován samostatnou stránkou doplněnou o multimediální obsah případně více stránek tvořící ucelený celek. Při přístupu uživatele na výukový objekt je do množiny navštívených objektů uloženo časové razítko, které představuje počátek navštívení výukového objektu.

Položka, kterou si uživatel z menu vybral je ze zobrazeného menu odebrána.

### Rozhodnutí o způsobu pokračování

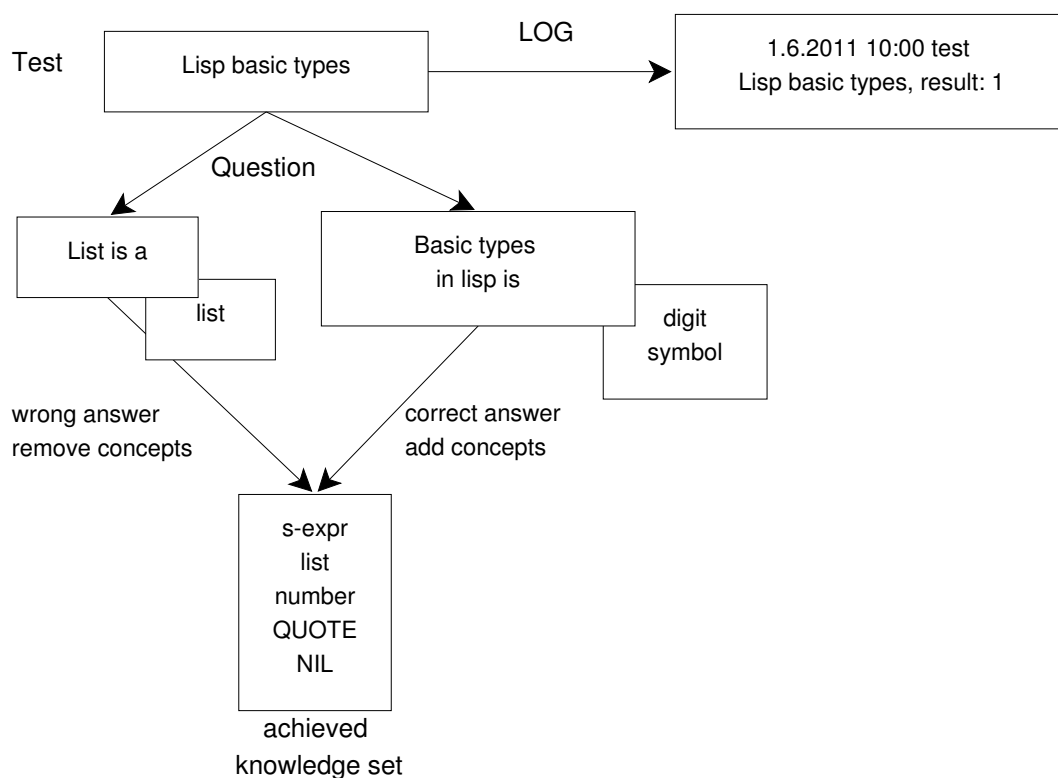
Když uživatel dokončí příslušný výukový materiál má tři možnosti jak pokračovat. Při každé z těchto možností se zaznamenává do uživatelské množiny navštívených objektů další časové razítko představující ukončení návštěvy výukového objektu.



Obrázek 9: Aktualizace uživatelského modelu a logu

1. Uživatel si přímo vybere v menu jinou položku. V této chvíli se navigace vrací do kroku „volba z menu a zobrazení“. Do uživatelské množiny dosažených znalostí se nezapíše žádný koncept z množiny konceptů daného výukového objektu.

2. Pokud si je uživatel jistý tím co se ve výukovém objektu naučil může pokračovat k dalšímu doporučenému výukovému objektu. Uživatel má k dispozici tlačítko „všemu rozumím, chci pokračovat dále“, po jehož stisknutí dojde k doplnění jeho množiny dosažených znalostí o koncepty daného výukového objektu viz obrázek 9.
3. Uživatel si své znalosti z výukového objektu chce otestovat – zvolí si tlačítko „chci zkusit test“ – nebo test je určen jako povinný. Je mu tedy nabídnut test vztahující se k probírané látce. Každá otázka v testu pokrývá určitou část problému daného výukového objektu a je jí přiřazena množina konceptů. Podle správnosti uživatelských odpovědí na jednotlivé otázky jsou mu koncepty do množiny znalostí přidány nebo naopak odebrány, viz obrázek 10.



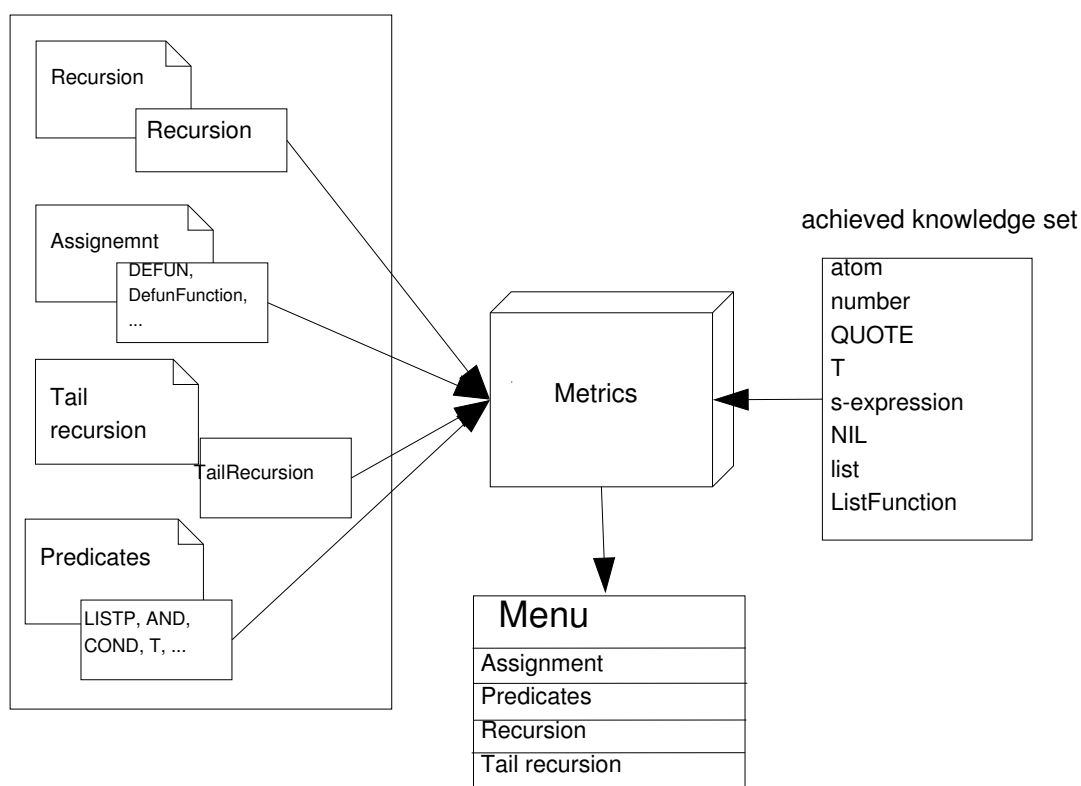
Obrázek 10: Vyhodnocení testu a aktualizace uživatelského modelu

Do logu uživatele je uložen záznam s časovým razítkem o začátku testu. Po jeho odeslání zpět a jeho vyhodnocení je do logu uložen další záznam s časovým razítkem určující ukončení testu. Společně s tím je uloženo také vyhodnocení testu. Uživatel

může test provést vícekrát. Pokaždé je znovu vyhodnocen a výsledek uložen.

## Úprava modelu a menu

Uživatelovo menu je aktualizováno podle jeho množiny znalostí a dostupných výukových objektů. Pro vlastní úpravu menu se používají metriky (viz obrázek 11), které reflektují odlišné aspekty a přístupy k navigaci uživatele. Definovali jsme několik základních metrik, které lze využít:



Obrázek 11: Přizpůsobení menu za použití metrik

**Počet prvků** – výchozí metrika, která slouží spíše pro otestování funkčnosti než pro vlastní navigaci. Výukové objekty jsou seříděny podle počtu prvků v množině konceptů, a to buď podle počtu všech prvků anebo jen podle počtu prvků, které zatím nejsou v množině dosažených znalostí.

**Ohodnocení neznámých konceptů neznámých výukových objektů** – pro každý dosud nenavštívený výukový objekt se vypočte aktuální ohodnocení z jeho množiny konceptů a množiny znalostí uživatele. Pro výpočet ohodnocení se vytvoří komplement těchto dvou množin. Vlastní ohodnocení se pak vypočte jako součet ohodnocení konceptů, které jsou obsaženy v získaném komplementu. Výukové objekty jsou setříděny podle vypočteného ohodnocení vzestupně.

**Ohodnocení všech konceptů** – výpočet probíhá stejně jako u předchozí metriky pro dosud nenavštívené výukové objekty. V tomto případě se však u každého výukového objektu bere v úvahu každý prvek z jeho množiny konceptů. Koncepty, které jsou již obsaženy v uživatelské množině znalostí přispívají k celkovému ohodnocení výukového objektu jen minimální částí. V našem případě jsme zvolili koeficient 0.1, kterým se násobí ohodnocení konceptu.

**Ohodnocení neznámých konceptů všech výukových objektů** – setřídění se provádí pro všechny výukové objekty ať navštívené nebo dosud nenavštívené. Pro každý výukový objekt se vytvoří komplement jeho množiny konceptů a uživatelské množiny dosažených znalostí. Pokud je komplement prázdný, s daným výukovým objektem se dále nepočítá. Pokud není, vypočte se ohodnocení výukového objektu jako součet všech ohodnocení konceptů ze získaného komplementu. Podle vypočteného ohodnocení jsou výukové objekty setříděny vzestupně. Tato metrika má význam v situaci, kdy jsou v rámci navigace využívány testy. Tím, že je možné koncepty v závislosti na správnosti odpovědi v testu z množiny dosažených znalostí odebírat, ty výukové objekty, které dané koncepty obsahují se zpětně začnou nabízet uživateli. Reflektuje tento přístup situaci kdy uživatel nepochopil správně nebo zapoměl předchozí látku a na základě této neznalosti neuspěl – jsou mu tedy nabídnuty znovu výukové objekty pojednávající o této látce.

**Metriky se zapomínáním** – setřídění se provádí opět pro všechny výukové objekty, ovšem pro již navštívené výukové objekty se započte „míra zapomínání“, která určí nakolik uživatel daný výukový objekt a jeho koncepty zapoměl. Velký význam pro stanovení tohoto faktoru má čas, kdy u dříve naučených výukových objektů se míra zapomínání zvyšuje. Tento přístup ovšem předpokládá, že se bude do logu uživatele ukládat časové razítko pro přístup ke každému konceptu, nejen k výukovému objektu.

**Podobnostní metriky** – výukové objekty jsou zvoleny na základě své podobnosti aktuálnímu výukovému objektu. Podobnost výukových objektů se stanovuje na základě porovnání jejich množin konceptů.

Předpokládáme, že výsledkem podobnostní metriky může být větší množství výukových objektů. Máme v plánu zjemnit uspořádání výukových objektů zavedením přídatné metriky vyhodnocující složitost obsahu výukového objektu. V případě zdrojových kódů jako obsahu výukových objektů předpokládáme jejich kategorizaci pomocí lexikální analýzy, vytvoření vektorové reprezentace a výpočtu míry příslušnosti. Tato budoucí práce přesahuje rozsah disertace a věnuje se jí jiný kolega v týmu školitele.

Nalezené výukové objekty jsou seříděny podle zvolené metriky. Pro uživatele je vytvořeno aktualizované menu reflektující jeho aktuální znalosti. Navigace následně probíhá krokem *Volba z menu a zobrazení výukového materiálu*.

## 6.4 Shrnutí

Způsob navigace, který jsme zde představili umožňuje provádět navigaci uživatele na základě metadat problémové domény, tak jak jsou zachyceny v prostoru konceptů. Ohodnocení prostoru konceptů, tak jak bylo prezentováno v předchozí kapitole, umožňuje porovnávání výukových objektů na základě jejich množin konceptů a jejich ohodnocení. Interakce uživatele se systémem a jeho volby přímo ovlivňují další nabízenou navigaci. Tento trend můžeme především pozorovat při volbě uživatele zda absolvuje či neabsolvuje test. Úspěšnost uživatele v testu má přímý dopad na další postup v navigaci.

Použitím různých metrik pro seřídění výukových objektů určuje způsob jakým se bude navigace uživatele ubírat. Flexibilitnost našeho přístupu spočívá v možnosti kdykoliv zvolenou metriku nahradit libovolnou jinou metrikou. Rozhodnutí o zvolené metrice může být dáno autorem kurzu, možnostmi systému nebo volbou dle předem definovaného klíče.

V následující kapitole představíme ověření našeho způsobu navigace v adaptivním webovém systému XAPOS na kurzu Lispu.





## Kapitola 7

# Ověření výsledků

Funkčnost našeho přístupu jsme ověřili experimentem [62, 63]. Pro tento účel jsme vytvořili zkrácenou verzi kurzu „Programování v Lispu“, zmíněnou již v předchozí kapitole. Kurz obsahuje 11 výukových objektů a prostor konceptů se skládá ze 107 konceptů. Rozsah testu byl vymezen na maximální dobu 60 minut.

Výpočet ohodnocení konceptů byl v souladu s návrhem (viz kapitola 5) proveden právě jednou při nahrávání prostoru konceptů do systému XAPOS, ve kterém jsme experiment prováděli. Popis XAPOSu je uveden v příloze B. Výukové objekty byly v průběhu experimentu dostupné v českém, anglickém a tureckém jazyce. Na konci každého výukového objektu byl pro uživatele připraven nepovinný test, ve kterém si mohli ověřit své znalosti z daného výukového materiálu. V rámci kurzu byl připraven i závěrečný test shrnující látku obsaženou v celém experimentu.

Experiment proběhl na PřF Ostravské univerzity, kde se ho zúčastnilo 35 studentů, a na univerzitě v Ankaře, kde se ho zúčastnilo 89 studentů. Do výsledků experimentu není započteno 21 studentů, kteří sice experiment započali ale nedokončili.

S ohledem na rozšíření PageRanking algoritmu o strukturální vazby v prostoru konceptů neočekáváme horší výsledky, než které dosahují jiné výzkumné týmy. Protože nejsou k dispozici standardizovaná data, umožňující případné srovnání napříč skupinami, zvolili jsme postup vyhodnocení rozdělením testerů do dvou respektive čtyř skupin pro vyhodnocení výsledků experimentu. Rozdělení do dvou skupin proběhlo podle příslušnosti k univerzitě – CZ a TR, rozdělení do čtyř skupin podle použitého algoritmu na ohodnocení konceptů a podle příslušnosti k univerzitě – Ranking CZ, Ranking TR, PageRank CZ, PageRank TR. Rozdělení uživatelů podle algoritmu nebylo za účelem porovnání obou algoritmů, ale za

účelem ověření funkčnosti navigace založené na ohodnoceném prostoru konceptů.

V rámci vyhodnocení experimentu jsme se zajímali o to, jak uživatelé přistupovali v experimentu k výukovým objektům, a to jak z hlediska uživatelů a počtu jejich zobrazení výukových objektů, tak z hlediska výukových objektů, kde jsme sledovali jejich návštěvnost. Další vyhodnocení experimentu, které nás zajímalo, bylo kolik času uživatelé strávili v experimentu jako celku, a kolik času nad jednotlivými výukovými objekty.

Důležitou částí experimentu, která má přímý vliv na způsob navigace, jsou testy dostupné ve výukových objektech. Sledovali jsme, jak uživatelé přistupovali k nepovinným testům – jejich ochota otestovat své znalosti a také jejich úspěšnost a vyhodnocení závěrečného testu, který souhrmně testoval dosažené znalosti v celém experimentu.

Zajímavou částí experimentu se ukázalo vyhodnocení logů uživatelů z hlediska cesty uživatele, kterou volil mezi jednotlivými výukovými objekty a jak se volba shodovala s nabízenou navigací. Vyhodnocení cest pro nás představuje i zpětnou vazbu, zda námi doporučená navigace odpovídá reálně aplikované navigaci uživatelem.

V další části vyhodnocení experimentu jsme se zaměřili na zjištění a potvrzení předpokládaných vzorů chování uživatelů. Vzory chování ukazují, jak se uživatelé při přechodu mezi jednotlivými výukovými objekty rozhodovali pro další navigaci, zda postupovali sekvencně podle námi navržené navigace, či zda přeskakovali na výukové objekty dle svého zájmu, či se vraceli nebo opakovaně přistupovali k již navštíveným výukovým objektům.

## 7.1 Statistiky

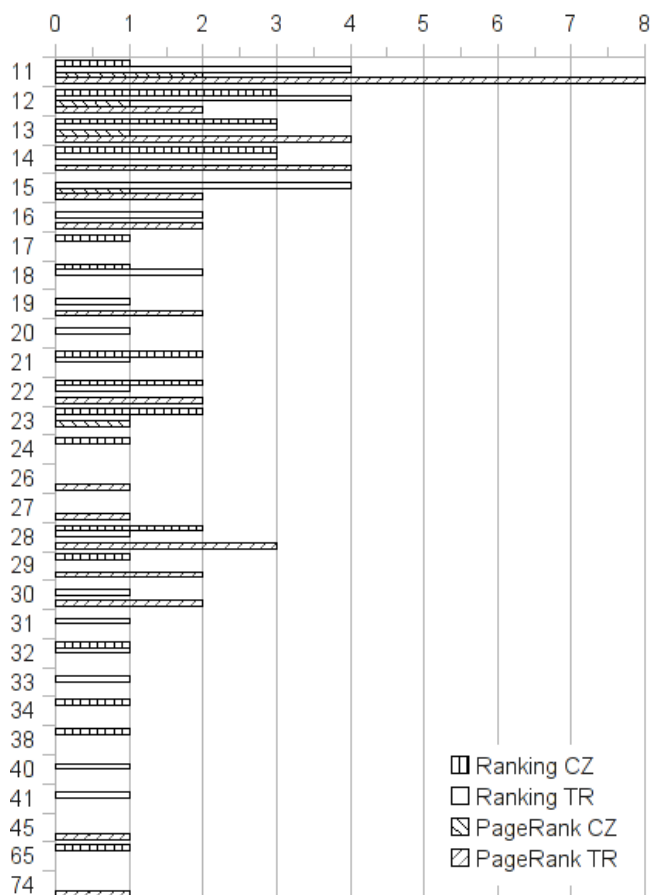
Jako podklad pro zpracování výsledků z experimentu byly použity logy, výsledky testů a množiny dosažených znalostí v modelu uživatele jednotlivých uživatelů.

Uživatelé nebyli ve svém pohybu v rámci testu nijak limitováni. Výukové objekty mohli navštívit opakovaně, testy umístěné na konci každého výukového objektu nemuseli spustit, nebo si je mohli vyzkoušet vícekrát.

Množství přístupů k výukovým objektům se z pohledu uživatelů vyskytovalo v rozmezí 11 přístupů, což představuje přístup ke každému výukovému objektu právě jednou, až po 74 přístupů (viz obrázek 12).

Průměrný počet přístupů k výukovým objektům byl 19.4 přístupů za uživatele, medián 15 přístupů za uživatele. V tabulce 5 je rozdělení pro čtyři skupiny uživatelů:

Na počty přístupů jsme se zaměřili z opačné strany, a to z hlediska výukových objektů.



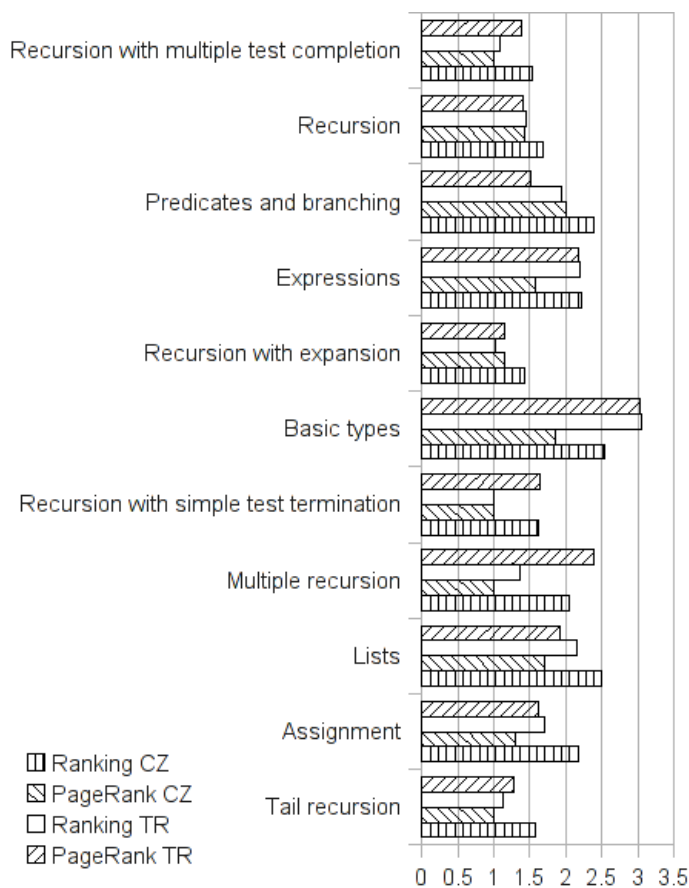
Obrázek 12: Počty přístupů uživatelů k dokumentům v rámci experimentu

skupina uživatelů	průměr	medián
Ranking CZ	22	21
Ranking TR	18	15
PageRank CZ	14	13
PageRank TR	19	15

Tabulka 5: Počty přístupů uživatele

Na obrázku 13 jsou zobrazeny průměrné hodnoty přístupů na jednotlivé výukové objekty. Výukový objekt byl v rámci experimentu navštíven každým uživatelem v průměru minimálně 1.2krát a maximálně 2.9krát. Mediánová hodnota je 1.6 přístupů na výukový objekt. Počty pro jednotlivé skupiny uživatelů jsou v tabulce 6.

Rozdíly mezi jednotlivými skupinami uživatelů jsou zanedbatelné.



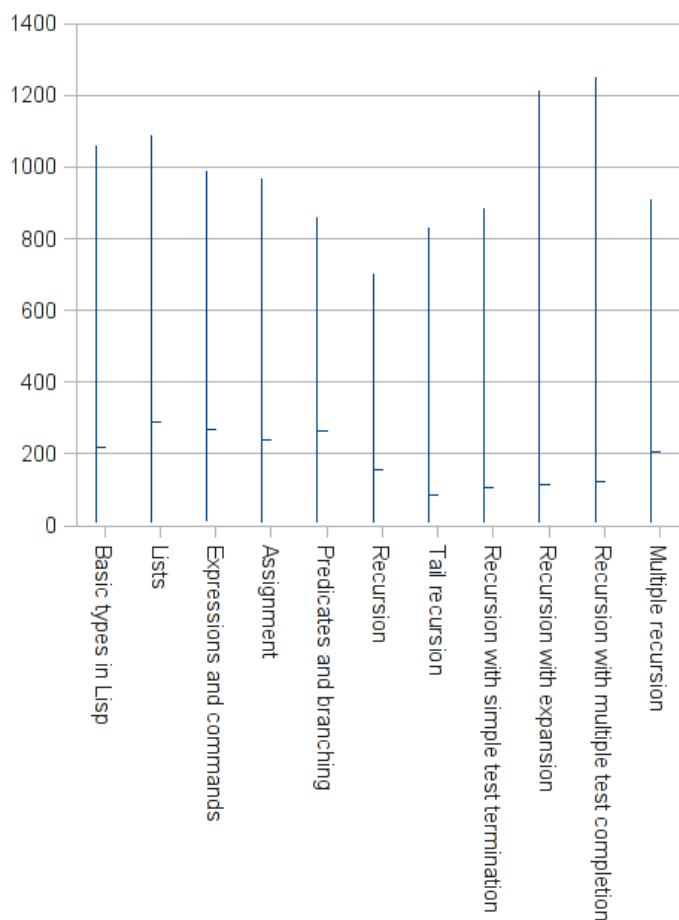
Obrázek 13: Průměrné počty přístupů uživatelů k výukovým objektům

skupina uživatelů	min	max	medián
Ranking CZ	1.4	2.5	2.1
Ranking TR	1.0	3.1	1.6
PageRank CZ	1.0	2.0	1.4
PageRank TR	1.2	3.0	1.6

Tabulka 6: Počty navštívení dokumentu uživatelem

## 7.2 Vyhodnocení stráveného času

Vyhodnocení času stráveného uživateli v rámci experimentu ukázalo některé zajímavé výsledky. 5 % uživatelů bylo schopno experiment zvládnout v čase pod čtyři minuty. Tito uživatelé pravděpodobně rychle „proklikali” všechny výukové objekty, tak aby experiment co nejrychleji ukončili. Na druhou stranu extrémní čas strávený v kurzu v podobě více než



Obrázek 14: Intervaly časů strávených uživateli na výukových objektech (v sekundách)

12 tisíc minut jsme brali jako chybnou hodnotu a nezahrnovali jsme jej do vyhodnocení.

Ze získaných dat jsme vypočetli průměrný čas uživatele strávený na jednom výukovém objektu jako 174 sekund. Medián byl 108 sekund.

Na obrázku 14 vidíme intervaly časů strávených uživateli na jednotlivých výukových objektech s vyznačenými mediánovými hodnotami. Rozmezí intervalů se pohybuje od několika sekund do několika set sekund. Velmi nízké hodnoty – pod 20 sekund si vysvětlujeme:

- uživatel v daném výukovém objektu jen hledal konkrétní informaci, jejíž ověření nezabralo moc času;
- uživatel patřil do skupiny uživatelů, kteří experiment pojali jako ztrátu času a jen příslušné výukové objekty proklikali;

- uživatel si chtěl znovu projít test, který je asociován s výukovým objektem. Protože test je umístěn na konci výukového objektu a spouští se až explicitním požadavkem, je nutné daný výukový objekt navštívit, což systém započítává jako přístup.

### 7.3 Vyhodnocení testů

V rámci experimentu jsme připravili pro uživatele možnost otestovat si své zkušenosti z probírané látky. Připravili jsme testy v každém výukovém objektu, kde si uživatel mohl otestovat znalosti z výukového objektu a závěrečný test, který obsahoval otázky pokrývající celý kurz. Otázky v testech byly definovány jako *otázka s výběrem odpovědi*, kde uživateli bylo prezentováno několik možností volby. Právě jedna odpověď v otázce byla správná.

#### Testy výukových objektů

Každý výukový objekt obsahoval na svém konci test určený k otestování znalostí z výukového materiálu. Test byl pro uživatele nepovinný. Pokud jej však uživatel odeslal k vyhodnocení, jeho výsledek měl vliv na postup v další navigaci. Na základě výsledku testu byla upravena uživatelova množina dosažených znalostí. Každá otázka testu měla definovanou svou množinu konceptů, kterou pokrývá z dané problémové domény a při správné odpovědi byly tyto koncepty do množiny dosažených znalostí přidány, při nesprávné odpovědi byly naopak odebrány, viz obrázek 10, strana 60.

Vyhodnocení testů z výukových objektů ukazují, že test v rámci výukového objektu byl využit v 17 % případů. Nejméně se uživatelé rozhodli absolvovat test u výukového objektu *Základní typy*, a to pouze v 11 % případů, nejvíce u výukového objektu *Přirazení* ve 23 % případů. Situace, kdy si uživatel test spustil, ale neodeslal jej ke kontrole jsme do těchto výsledků nezapočítávali.

Z celkových výsledků lze odvodit, že test jako navigační prvek má své opodstatnění. Z logů jsme zjistili, že uživatelé, kteří neuspěli v testu se k danému výukovému objektu vraceli. Všem uživatelům, kteří v testu v odpovědi chybovali se z množiny dosažených znalostí přidružené koncepty odstranily. Uživatel tak „ztratil“ znalost daných konceptů a musel se je průchodem příslušnými výukovými objekty naučit znovu.

### Závěrečný test

Závěrečný test byl připraven v rámci kurzu pro otestování znalostí uživatelem na závěr experimentu. Omezení byla v testu následující:

- Test nebyl povinný – někteří uživatelé jej tedy v rámci experimentu vynechali úplně.
- Test bylo možné spustit kdykoliv v průběhu experimentu – někteří uživatelé test vyzkoušeli na začátku experimentu, případně v průběhu. Většina uživatelů však test absolvovala na konci experimentu.
- Test bylo možno vyzkoušet vícekrát. Z vyhodnocení výsledků vyplývá, že tohoto důvodu, a z důvodu nepovinnosti testu uživatel v průměru spustil test 1.4krát. Zajímavým zjištěním bylo, že více než dvakrát si test vyzkoušelo 10 % uživatelů.

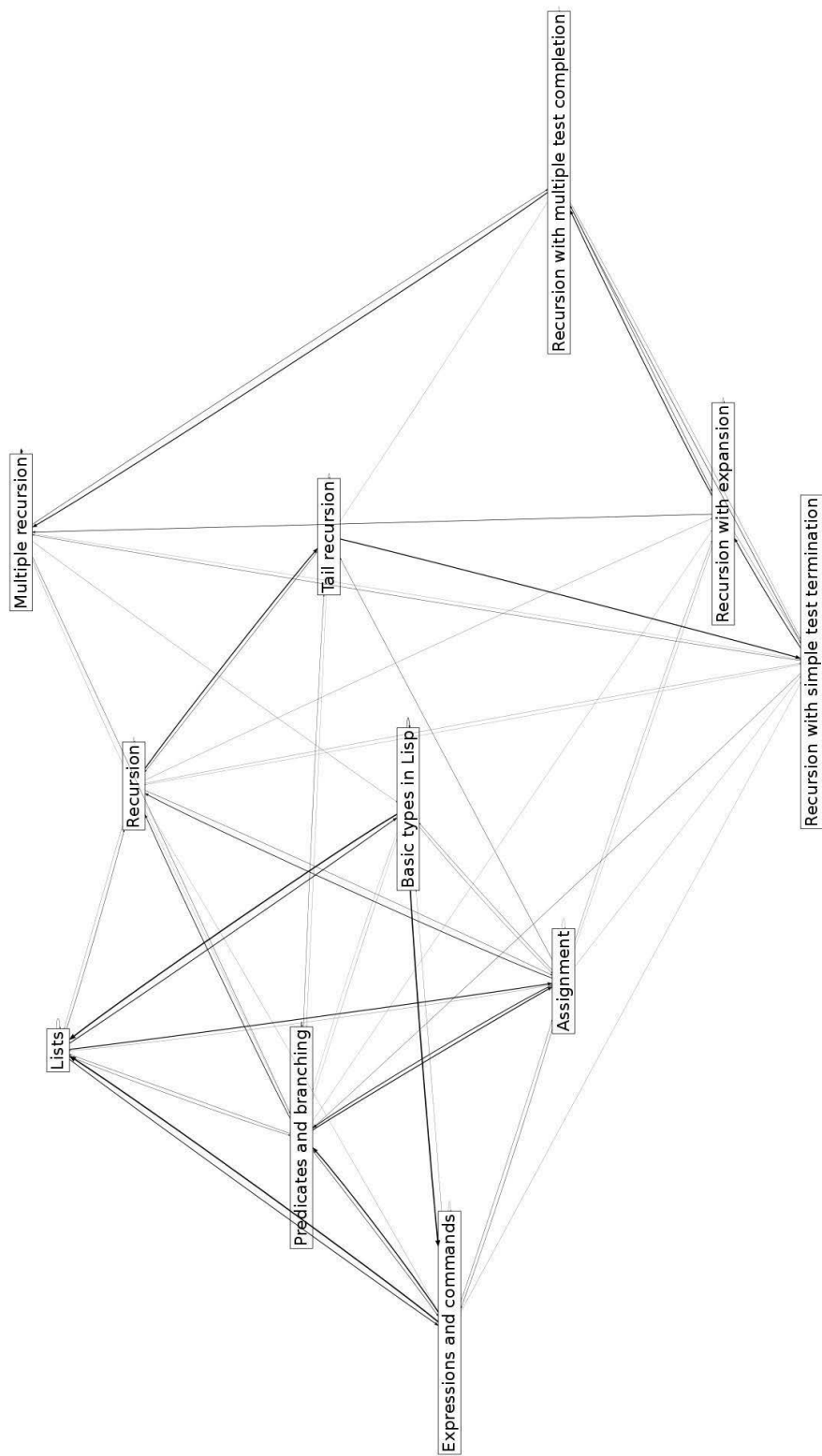
Výsledky testů ukazují, že celkově 26 % studentů mělo v závěrečném testu hodnocení 0 (Ranking CZ: 27 %, Ranking TR: 23 %, PageRank CZ: 25 %, PageRank TR: 29 %). 90 % z těchto studentů se o závěrečný test pokusilo znovu s lepším výsledkem. Na druhé straně výsledek s plným hodnocením nebo s jednou špatnou odpovědí dosáhlo 32 % studentů (Ranking CZ: 33 %, Ranking TR: 34 %, PageRank CZ: 16 %, PageRank TR: 30 %).

Dosažené výsledky ukázaly, že i závěrečný test je důležitou součástí experimentu. To že jsme umožnili uživatelům spustit si test kdykoliv v průběhu experimentu ukázalo, že mnoho uživatelů se chtělo experimentu vyhnout či si své znalosti potvrdit hned na začátku experimentu. Po neúspěchu v tomto testu kurz absolvovali a své znalosti následně potvrdili v opakovaném testu.

## 7.4 Vyhodnocení cest

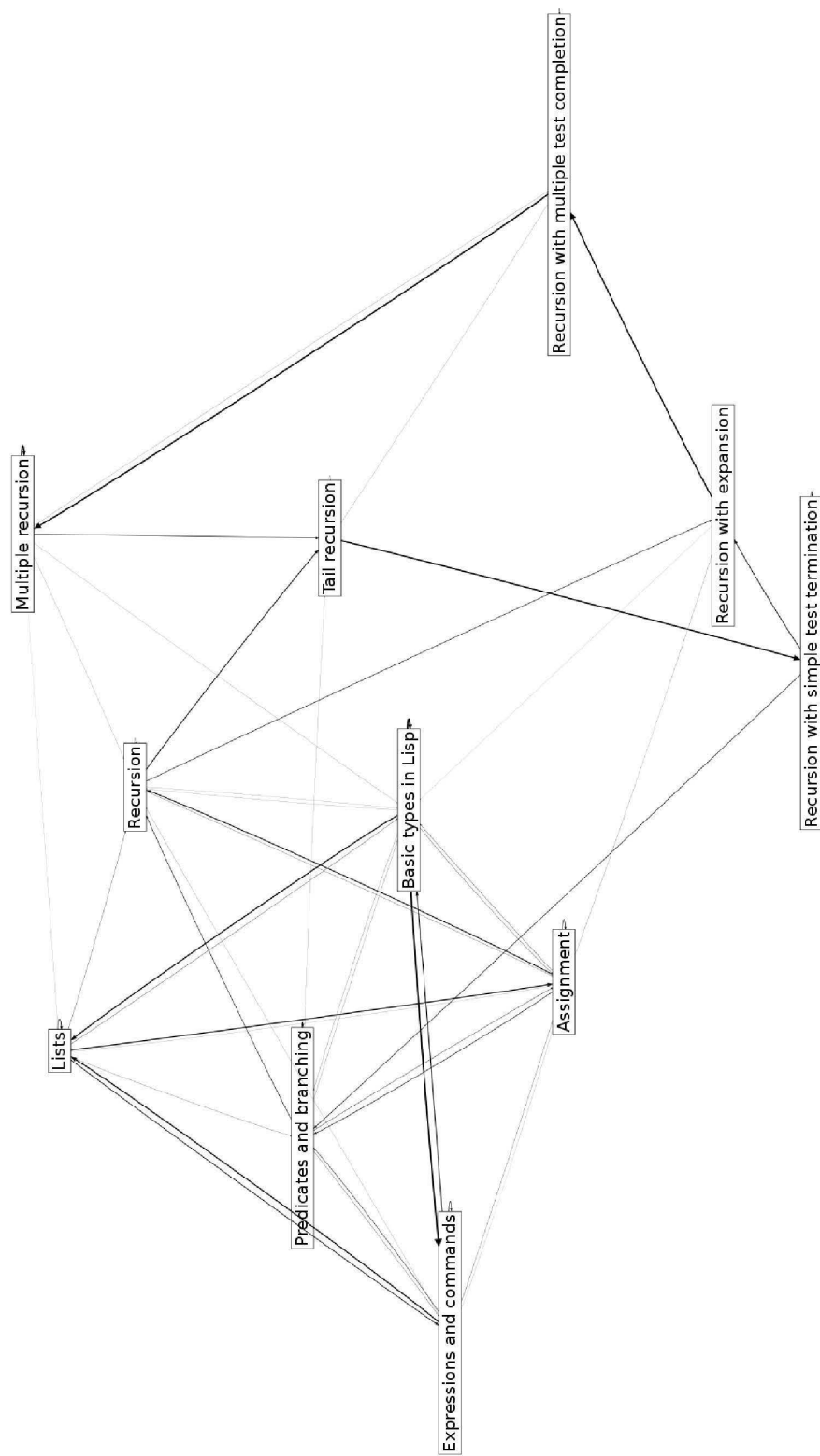
Vyhodnocení cest uživatelů v rámci experimentu jsme provedli za účelem zkoumání, mezi kterými výukovými objekty byly přechody uživatelů nejčastější. Pro tento typ analýzy jsme uživatele rozdělili do dvou skupin a to na CZ a na TR uživatele.

Analýzou logů uživatelů jsme získali přechody uživatelů mezi jednotlivými výukovými objekty. Započítávali jsme přechody, které byly uskutečněny jak na základě navržené navigace, tak také ty, kdy se uživatelé pohybovali na základě svého rozhodnutí, a také návraty na stejný výukový objekt.



Obrázek 15: Graf průchodů uživatele CZ kurzem





Obrázek 16: Graf průchodů uživateli TR kurzem

Na obrázcích 15 a 16 můžeme vidět přechody mezi všemi výukovými objekty v rámci experimentu. Na obrázku 15 jsou zobrazeni uživatelé CZ a na obrázku 16 uživatelé TR. Síla čáry mezi jednotlivými výukovými objekty určuje množství přechodů, které se mezi těmito výukovými objekty odehrály – přechody s minimální hodnotou byly z obrázků vypuštěny pro lepší přehlednost.

Zajímavým zjištěním byly opakované návštěvy stejného výukového objektu. Počet těchto návštěv představuje 18 % všech návštěv výukových objektů. Ačkoliv se toto zdá jako velké číslo, jeho vysvětlení může ležet v principu uspořádání výukových objektů. Jak již bylo vícekrát uvedeno na konci výukového objektu se nachází test, který může student využít k ověření svých získaných znalostí. Test se dá spustit pouze ze stránky příslušného výukového objektu a tedy pokud uživatel má zájem jen o test – například si jej jde znovu zkusit po tom co prvně neuspěl nebo má zájem o lepší výsledek – musí daný výukový objekt navštívit, aby mohl spustit test. Dojde tak k zapsání informace o přístupu uživatele k danému výukovému objektu do jeho logu. Podrobnější analýzou jsme zjistili, že tento předpoklad byl správný a mnoho z těchto opakovaných přístupů je pouze v řádu několika sekund, které stačí k otevření výukového objektu, posunutí na jeho konec a spuštění testu.

Ze získaných dat můžeme také vyčíst že nejsilnější přechody mezi výukovými objekty odpovídají přechodům v „ideálním průchodu“, který byl získán automatizovaným výpočtem cesty, kdyby uživatel vždy postupoval podle navržené navigace a zvolil možnost „všemu rozumím, chci pokračovat dále“. Nejvíce je tento trend pozorovatelný mezi výukovými objekty „Recursion with expansion“ a „Recursion with multiple test completion“ a mezi výukovými objekty „Recursion with multiple test completion“ a „Multiple recursion“ kde tyto přechody použilo 65 % všech uživatelů.

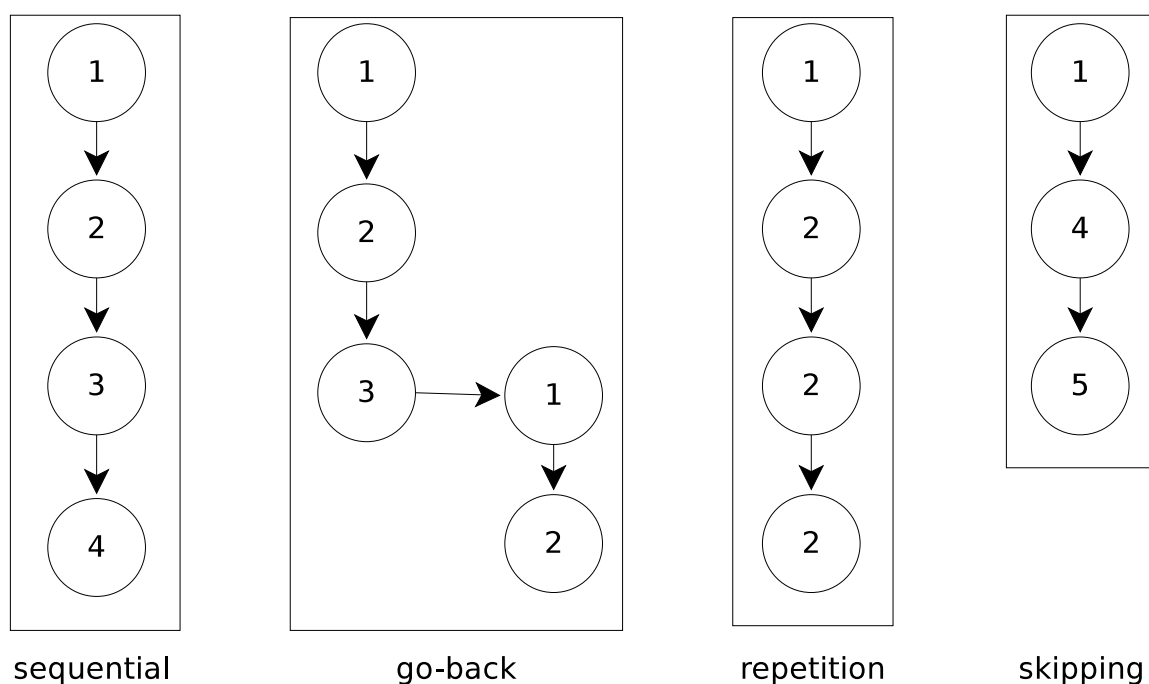
Analýza průchodů kurzu uživateli může ukázat na možné problematické místa kurzu, jako jsou příliš obtížný výukový materiál nebo naopak snadný, nebo také nevhodné rozdělení výukového materiálu do výukových objektů. Tyto vazby jsme nehledali a nezkoumali s ohledem na charakter omezeného obsahu kurzu v experimentu i s ohledem na fakt, že disertace a náš výzkum jsou orientovány informaticky.

## 7.5 Vzory chování

V získaných datech od uživatelů jsme se zaměřili jako na jednu část experimentu na identifikaci případných vzorů chování, které by byly využitelné pro navigaci. V navigaci je

možné následně vzory chování využít jako doplňkovou informaci při vytváření nabídky pro uživatele. Pro analýzu vzorů chování uživatelů jsme zvolili rozdělení uživatelů do tří skupin a to na uživatele CZ, uživatele TR a všechny uživatele.

V datech uživatelů jsme identifikovali čtyři očekávané vzory chování, viz obrázek 17:



Obrázek 17: Vzory chování uživatelů

- *Sekvenční (sequence)* – uživatel se pohybuje mezi výukovými objekty podle jemu nabídnuté navigace. Tento vzor chování je typický u uživatelů, kteří se s kurzem a probíranou látkou obvykle setkávají poprvé a nemají tudíž o ní dosud žádné znalosti. Druhou skupinou, která obvykle volí tento vzor chování jsou uživatelé, kteří si definovali svůj cíl a systém jim připravil cestu jak se k němu dostat.
- *Opakování (repetition)* – uživatel navštíví výukový objekt opakovaně bez toho aby navštívil nějaký jiný. Toto chování se obvykle vyskytne v případě, že uživatel se pokusí o zvládnutí testu, který je na konci výukového objektu. Pokud v něm neuspěje nebo uspěje za svým očekáváním, zkusí si projít výukový materiál znovu.
- *Návrat (go-back)* – uživatel se rozhodne vrátit k nějakému dříve navštívenému výukovému objektu. Toto chování je obvyklé v situaci, kdy si uživatel chce ověřit již

dříve prezentované informace, a proto navštíví výukový objekt, kde se informace nacházejí. Obvykle se pak uživatel vrací k výukovému objektu ze kterého vyšel případně pokračuje nějakým následujícím.

- *Přeskakování (skipping)* – uživatel se rozhodne, že si zvolí cestu, bez ohledu na to co mu systém nabízí. Postupuje mezi výukovými objekty takovou cestou a způsobem, který vyhovuje jemu.

vzor	uživatelé	1/10	1/3	1/2	2/3
sekvenční	ALL	94	79	59	42
	CZ	78	74	41	29
	TR	100	81	65	46
opakování	ALL	49	5	3	1
	CZ	22	0	0	0
	TR	59	7	4	1
návrat	ALL	60	28	7	0
	CZ	67	44	22	0
	TR	57	23	1	0
přeskakování	ALL	38	6	4	3
	CZ	41	22	15	11
	TR	37	0	0	0

Tabulka 7: Vzory chování uživatelů

Pro analýzu vzoru chování se používá každý přechod uživatele mezi dvěma výukovými objekty, který je klasifikován jako instance jednoho z definovaných vzorů chování. Výsledky, které jsme získali z dat uživatelů ukazují, že pouze 5 % studentů používá čistě sekvenční vzor chování. Ostatní uživatelé se od daného vzoru méně či více odkloňují. V tabulce 7 jsou výsledky pro jednotlivé vzory chování s ohledem na procentuální četnost jejich používání. Sloupec 1/10 představuje, že daný vzor byl použit v 1/10 průchodů kurzem, 1/3, 1/2 a 2/3 respektive.

## 7.6 Shrnutí

Cílem experimentu nebylo porovnání PageRank a našeho Ranking algoritmu. Naším cílem bylo ověření, zda námi navržený adaptační navigační mechanismus má reálnou šanci v praxi. Celkově experiment potvrdil, že je princip navigace implementovatelný a použitelný. Získané výsledky z průchodu zkrácenou verzí kurzu Lisp ukázaly limitující faktory řešení, které

se skrývají především v kvalitním zpracování prostoru konceptů a v navázání prostoru výukových objektů na koncepty.

Námi navržená navigace v ideálním průchodu, tak jak jsme ji zjistili vyhodnocením ze systému odpovídá návrhu navigace autorem kurzu. V přípravné fázi jsme výukové objekty z kurzu Lispu setřídili podle zamýšleného pořadí studia. Ze zjištěných cest mezi výukovými objekty jsme zjistili, že nejvíce preferované přechody mezi dvěma výukovými objekty odpovídají zamýšlené navigaci.

Ačkoliv test v rámci výukových objektů nebyl vždy uživateli využit, myslíme si, že má své opodstatnění v principu navigace. Jeho výsledek zásadně ovlivňuje další směřování navigace. Možnou úpravou, aby test získal větší roli v navigaci, je udělal jej povinným. V našem experimentu jsme k tomuto opatření nesáhli.



## Kapitola 8

# Zhodnocení a závěr

Naším cílem je zkvalitnění navigace pro uživatele. Zaměřili jsme se na princip navigace nad dokumenty postavený na využití jejich metadat. Předcházející text popsal současný stav poznání v oblasti sémantického webu a adaptace. Podrobně jsme se věnovali našim východiskům, přístupu, návrhu a realizaci řešení a experimentálnímu ověření. Jádro práce bylo publikováno na lokálních i renomovaných mezinárodních konferencích, vždy tématicky zaměřených tak, aby nám poskytly prostor pro diskusi a získání zpětné vazby a umožnily nám i sledování aktuálního vývoje v oblasti personalizace a adaptace.

K hlavním výsledkům naší práce patří návrh algoritmu pro ohodnocení existující množiny konceptů. Každému konceptu z množiny konceptů je vypočteno ohodnocení na základě jeho pozice, relací a ohodnocení jeho sousedů. Propojením konceptů a výukových objektů navíc získáváme možnost výpočtu ohodnocení výukového objektu.

Navigační mechanismus, který jsme představili, využívá propojení mezi výukovými objekty a koncepty k vytváření navigace pro uživatele. Uživateli jsou nabídnuty takové výukové objekty, které odpovídají jeho aktuálnímu znalostem. Výběr výukových objektů pro uživatele probíhá za použití představených metrik. Přímý vliv na možný směr navigace má chování a reakce uživatele na předkládanou navigaci. To představuje další hlavní výsledek.

Představený princip navigace jsme implementovali do webového systému XAPOS. Zvolili jsme variantu vytvoření vlastního systému po praktických zkušenostech s jinými systémy. Tak jsme mohli experimentálně ověřit naše návrhy na reálných skupinách uživatelů. Praktické testování našeho způsobu navigace jsme realizovali na studentech z Ostravské univerzity a studentech z univerzity v Ankaře. Tyto odlišné skupiny uživatelů nám umožnily ověřit si princip navigace na kulturně odlišných skupinách i funkční vícejazyčnost našeho

řešení. Každý výukový objekt může být definován více jazykových variant, ze kterých si uživatelé mohou vybrat. Jedná se však o vícejazyčnost obsahu výukového objektu. Metadata daného výukového objektu i jejich vazby v prostoru konceptů jsou pro všechny jazyky společná. Princip navigace ve vícejazyčném prostředí je tedy pro uživatele identický a nezáleží jakou jazykovou variantu si zvolí. Umožňuje to například možnost studia materiálů v anglickém jazyce s možností přepnout si do češtiny tam, kde si uživatel není jistý. Tento princip přináší i další zajímavou variantu a to s rozvojem služeb automatizovaného překladu možnost vytvářet nové jazykové varianty za běhu na základě požadavku. Další variantou je vyhledávání odpovídajícího obsahu na základě metadat v požadovaném jazyce. Tuto oblast jsme zatím nezkoumali, ale je to jeden z možných budoucích směrů vývoje.

Testování na uživateli ukázalo, že námi navržený princip navigace je prakticky realizovatelný a je možné jej používat pro navigaci uživatelů. Získané výsledky ukázaly možné nedostatky našeho řešení a možnosti dalšího rozvoje principu navigace. Jedním z velmi důležitých poznatků je, že celá úspěšnost navigace je velmi úzce spjata s kvalitou návrhu prostoru konceptů a s propojením konceptů a výukových objektů. Vytváření prostoru konceptů je časově i znalostně náročný proces, který se v současné době snaží mnoho autorů zautomatizovat. Způsob vytváření prostoru konceptů z obsahu představuje komplexní problém a přesahuje rámec této práce. Metodou poloautomatizovaného vytváření prostoru konceptů z existujícího obsahu za pomoci lexikální analýzy a tagování a parsování se zabývá kolega z týmu řešitele. Částečně jsem se na tomto řešení podílel viz [64]. Předběžné výsledky jsou slibné a ukazují že metoda poloautomatizovaného zpracování obsahu výrazně předzpracuje základ budoucích konceptů a navrhne prototypy vztahů různých typů mezi nimi.

Námi navržený algoritmus pro ohodnocení prostoru konceptů a navigace založená na tomto ohodnocení má potenciál reálného uplatnění. Přidání metadat ve formě konceptů umožňuje automatizovanou navigaci nad prostorem výukových objektů bez zásahů autora obsahu. Protože v našem navigačním mechanismu nepoužíváme autorem definována pravidla pro zobrazení výukových objektů ani předem určené pořadí nebo závislosti mezi výukovými objekty eliminujeme tímto podstatně vliv a zvyklosti autora. Velkou výhodou je možnost vytvoření kurzu z více různých zdrojů od více autorů. Podmínkou je, že výukové objekty budou popsány pomocí konceptů ze stejného prostoru konceptů, což je pro stejnou aplikační doménu splnitelné.



## 8.1 Budoucí práce

Představený princip navigace je možné dále rozvíjet například v naznačených oblastech.

### Míra znalosti

Navázání mezi prostorem výukových objektů a prostorem konceptů je určeno množinou konceptů příslušných k danému výukovému objektu. Informace o příslušnosti zaznamenáváme v současné době v dvouhodnotové podobě. To znamená, že buď koncept je nebo není uveden v množině konceptů výukového objektu. Jako vhodné rozšíření se jeví zavedení míry znalosti jako procentuálního vyjádření příslušnosti konceptu k výukovému objektu nebo testovací otázce. Ekvivalentně s tím pak zavedení míry znalosti konceptu do množiny znalostí uživatele.

Tato změna přináší lepší a „jemnější“ možnost navigace uživatele. Princip navigace je možné rozšířit o další metriky, počítající s vyjádřením znalosti konceptu v procentuálním poměru. Například by bylo možné brát v metrice v úvahu jen takové koncepty, které uživatel již zná s definovanou mírou znalosti nebo naopak nezná.

V principu navigace by byla nutná úprava způsobu aktualizace množiny znalostí uživatele. Pro nově přidávané objekty do množiny znalostí uživatele by platilo pravidlo že se do ní přidají se svou mírou znalosti. Pokud by koncept v množině znalostí již obsažen byl existuje více možností aktualizace míry znalostí a to například součtem obou hodnot míry znalosti, jejich průměrem či zvolením vyšší míry znalosti. V případě testovacích otázek u správných odpovědí je princip stejný, v případě nesprávných odečtem hodnot, jejich průměrem či volbou nižší míry znalosti.

Pro každý koncept v množině znalostí uživatele je pak nutné sledovat aby jeho míra znalostí nepřekročila maximální hranici (100 %). V případě spadnutí pod spodní hranice do negativních hodnot je možné tuto informaci využít pro navigaci.

### Podobnost a složitost

Podobnost výukových objektů je možné definovat více způsoby. Pro zlepšení navigace je vhodná definice podobnosti výukových objektů na základě podobnosti jejich množin konceptů. Porovnáním množin konceptů výukových objektů s aktuálně zobrazeným výukovým objektem je možné nalézt takové výukové objekty, které se podobají z hlediska obsažených znalostí. Plánujeme použít tento princip nejen jako metriku pro výběr dalších výukových

objektů ale hlavně jako doplňkovou informaci pro aktuálně zobrazený výukový objekt, kdy se k němu budou nabízet podobné výukové objekty typu ukázkový příklad, případně doplňkový cizojazyčný obsah.

Zavedením složitosti výukového objektu jako dalšího kritéria při navigaci nám pak umožní setřídění výukových objektů, které jsou si z hlediska obsažených konceptů blízké. V případě výukových objektů typu ukázkový příklad předpokládáme jejich kategorizaci vzhledem k výkladovým výukovým objektům a následně zjemnění jejich uspořádání pomocí vybraných metrik (známých ze softwarového inženýrství) charakterizujících například složitost, pokročilost, rozsah, ... Těto oblasti výzkumu se věnuje jiný kolega v týmu školiče [52].

### **Další možnosti rozšíření**

V rámci prováděného experimentu se studenty jsme zkoumali zda se v datech uživatelů nenacházejí vzory opakujícího se chování, které by bylo možné využít pro navigaci. Podařilo se nám v datech identifikovat čtyři očekávané vzory. Identifikací vzoru chování uživatele je možné získat další informaci vhodnou při tvorbě navigace. Sledováním chování uživatele je možné na základě předem definovaných vzorů odhadovat další kroky uživatele a přizpůsobovat mu tak navigaci. Například je možné, pokud se uživatel neustále vrací k nějaké látce nabídnout mu její hlubší vysvětlení či více příkladů.

Další možnosti rozvoje vidíme v definování cíle či cílů, kde si uživatel zvolí přímo koncepty, které se chce naučit. V navigaci pak mohou brát v úvahu jen ty výukové objekty, které zvolené koncepty obsahují. Vhodnou úpravou se pak jeví i změna zobrazovaného uživatelského rozhraní, kde se uživateli budou zobrazovat informace o jeho postupu, případně přímé zobrazení konceptů (jejich vhodná reprezentace) místo menu s výukovými objekty.

Automatizované nabízení externích odkazů představuje další logický krok v možnostech navigace. Doplněním o odkazy na externí zdroje je možné uživatelům zpřístupnit doplňkový rozšiřující zdroj informací k danému výukovému materiálu. Vyhledávání odpovídajících externích odkazů je možné na základě konceptů a klíčových slov a s případným využitím folksonomie a principů Webu 3.0 a eLearnigu 2.0. Velmi vhodné se toto jeví zejména pro vyhledávání dalších ukázkových příkladů vysvětlujících danou látku.

## Kapitola 9

# Conclusions and future work

The goal of our work is to improve navigation for users. We focused on navigation scheme over document based on their metadata. One of the main results of our work is design of the algorithm for evaluating the existing concepts set. For each concept from the concepts set an evaluation is calculated based on its position, relations to other concepts and their evaluations.

The navigation scheme we presented uses the connection between learning objects and concepts to personalize navigation for a user. Users are offered learning objects that are chosen according to their actual knowledge. For choosing suitable learning objects metrics are employed. User's responses to the presented learning object and their choices influence the future navigation. This represents another major outcome.

We implemented a web-based system XAPOS which uses described navigation scheme. So we were able to experimentally validate our navigation scheme. We performed a practical testing with students from the University of Ostrava and students from the university in Ankara. These different groups of users allowed us to verify the principle of navigation on the culturally diverse groups and internationalization of our solution.

Testing on users showed us that the proposed navigation scheme is practically feasible. The obtained results showed the possible shortcomings of the solution as well as opportunities for further development. A very important finding was that the whole success of the navigation is very closely connected with the quality of the concepts space and linkage to the learning objects.

The navigation scheme is usable in real life navigation of a user. Adding metadata to learning objects allows for navigation over the learning objects space without the interventions

of the author of the material. This allow us to include learning material from several authors covering the same problem domain. The only requirement is for the learning objects to be described by the same concept space.

The navigation scheme is possible to extend in such areas as:

- quantitative measure of knowledge – information about membership of a concept with learning object could not be only in binary form but also as a quantitative measure. The value will show how much the learning object covers the concept. In the navigation scheme this information could be used to store user’s knowledge of a particular concept also as quantitative measure. This approach will allow to define new metrics using the information for selecting appropriate learning objects, for example only such learning object where user knows concepts with at least predefined presentage.
- similarity and complexity – users can be provided with learning objects which are similar to the current learning object in the meaning of similarity of concepts. Using complexity of a learning object provides a possibility to order the similar learning objects. This specially applys to the learning objects of the type example where more than one learning object can be similar. The learning objects have to be evaluated in the meaning of complexity – a field of study which another student of the supervisor is focusing on [52].
- other further possibilities include using navigation patterns for navigation; defining goal or goals in the meaning of concepts which users wants to learn and provide them with appropriate learning objects; presenting users with links to corresponding external material, which will be searched using the concepts of the learning object.

# Literatura

- [1] Alani H., Brewster Ch. Metrics for Ranking Ontologies. In *Proceedings of 4<sup>th</sup> Int. EON Workshop, 15<sup>th</sup> Int. World Wide Web Conf.* Edinburgh, 2006. pp. 26–33.
- [2] Andrejko A., Barla M., Bieliková M. Ontology-based user modeling for web-based information systems. In *Wita Wojtkowski, W. Gregory Wojtkowski, Jože Zupancic, Gabor Magyar, and Gabor Knapp, editors, Advances in Information Systems Development, New Methods and Practice for the Networked Society*, Vol. 2, 2007. Springer Science+Business Media, New York. pp. 457–68.
- [3] Andrejko A., Bieliková M. Investigating Similarity of Ontology Instances and Its Causes. In *Proceedings of the 18th international conference on Artificial Neural Networks, Part II, ICANN '08*. Prague, Czech Republic. Springer, 2008. pp. 1–10.
- [4] Andrejko A., Bieliková M. Comparing Instances of Ontological Concepts for Personalized Recommendation in Large Information Spaces. In *Computing and Informatics*, Vol. 28, No. 4. 2009. pp. 427–450.
- [5] Aroyo L., Dicheva D. AIMS: Learning and Teaching Support for WWW-based Education. In *Int. J. for Continuous Eng. Education and Life-long Learning* Vol. 11 No. 1/2. 2001. pp. 152–164
- [6] Aroyo L., Dicheva D., Cristea A. Ontological support for Web Courseware Authoring. In *Cerri, S.A., Gouardéres, G., Paraguaçu, F. (eds.) ITS 2002*. LNCS, vol. 2363, Springer. 2002. pp. 270–280.
- [7] Aroyo L., Dolog P., Houben G., Kravcik M., Naeve A., Nilsson M., Wild F. Interoperability in Personalized Adaptive Learning. In *Educational Technology & Society*, Vol. 9 No. 2 2006. pp. 4–18.

- [8] Barla, M., Bieliková M. Ordinary Web Pages as a Source for Metadata Acquisition for Open Corpus User Modeling. In *Proceedings of IADIS WWW/Internet 2010*. IADIS Press. 2010. pp. 227–233.
- [9] Bernes-Lee T., Hendler J., Lassila O. The Semantic Web. *Scientific American*. May 2001. pp. 29–37.
- [10] Bernes-Lee T. Semantic Web. *XML2000*, on <http://www.w3.org/2000/Talks/1206-xml2k-tbl/> [online: 8. 6. 2011]
- [11] Bieliková M., Moravčík M. Modeling the Content of Adaptive Web-Based System Using an Ontology. In *proceedings of 1<sup>st</sup> International Workshop on Semantic Media Adaptation and Personalization SMAP06*. IEEE Computer Society. 2006. pp. 115–120
- [12] Bober M., Šaloun P., Velart Z. Knowledge Mining from Adaptive Course, In *Conference Proceedings from 5<sup>th</sup> International Conference on Emerging e-Learning Technologies and Applications - ICETA 2007*. Elfa, s.r.o. 2007. pp. 355–360.
- [13] Breslin J. G., Passant A., Decker S. The Social Semantic Web. Springer 2009. ISBN 978-3-642-01171-9
- [14] Brin S., Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Seventh International World-Wide Web Conference (WWW 1998)*. Elsevier Science Publishers B. V. 1998. pp. 107–117.
- [15] Brusilovsky P. Methods and Techniques of Adaptive Hypermedia. In *Adaptive Hypertext and Hypermedia*, Kluwer Academic Publisher. 1998. p. 1–44.
- [16] Brusilovsky P. Adaptive hypermedia. In *User Modelling and User Adapted Interaction*. Vol. 11, No. 1. 2001. pp. 87–110.
- [17] Brusilovsky P., Sosnovsky S., Yudelso M. Accessing Interactive Examples with Adaptive Navigation Support. In *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'04)*. IEEE Computer Society. 2004. pp. 842–843.
- [18] Cristea A., De Mooij A. Adaptive Course Authoring: MOT, My Online Teacher. In *Proceedings of ICT-2003, IEEE LTF International Conference on Telecommunications, Telecommunications + Education Workshop*. IEEE Computer Society. 2003 pp. 1762–1769.

- [19] Cristea A., De Mooij A. LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. In *Proceedings of The Twelfth International World Wide Web Conference*. ACM New York. 2003.
- [20] De Bra P., Calvi L. Aha! an open adaptive hypermedia architecture. In *The New Review of Hypermedia and Multimedia*. Vol. 1, No. 4. 1998. pp. 115–139.
- [21] De Bra P., Houben G.J., Wu H. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*. ACM New York. 1999. pp. 221–239.
- [22] De Bra P., Aerts A., Smits D., Stash N. AHA! meets AHAM. In *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer LNCS 2347. 2002. pp. 381–384.
- [23] 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 fourteenth ACM conference on Hypertext and hypermedia. HYPERTEXT '03*. ACM New York. 2003. pp. 81–84.
- [24] De Bra P., Ruiter J. AHA! Adaptive Hypermedia for All. In *Proceedings of the AACE WebNet Conference*. 2002. pp. 262–268.
- [25] Dicheva D., Dichev C. Helping Courseware Authors to Build Ontologies: the Case of TM4L. In *Proceeding of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*. 2007. pp. 77–84.
- [26] Doan A., Madhavan J., Domingos P., Halevy A. Ontology Matching: A Machine Learning Approach. In *Handbook on Ontologies in Information Systems*. Springer. 2003. pp. 397–416.
- [27] Ehrig M., Staab S. QOM – Quick ontology mapping. In *Proceedings of 3rd International Semantic Web Conference (ISWC04)*. Springer. 2004. pp. 683–697.
- [28] Fortuna B., Grobelnik M., Mladenic D. Semi-automatic Construction of Topic Ontology. In *Semantics, Web and Mining, Joint International Workshop, EWMMF 2005 and KDO 2005*. 2005. pp. 121–131.

- [29] Giunchiglia F., Yatskevich M., Shvaiko P. Semantic matching: Algorithms and implementation. In *Journal on Data Semantics IX*. Vol. 1. Springer Berlin/Heidelberg. 2007. pp. 1–38.
- [30] Gruber T.R. Toward principles for the design of ontologies used for knowledge sharing. In *International Journal of Human-Computer Studies - Special issue: the role of formal ontology in the information technology*. Vol. 43, No. 5–6. Academic Press. 1995. pp. 907–928.
- [31] Hazman M., El-Beltagy S.R., Rafea A. Ontology Learning from Domain Specific Web Documents In *the International Journal for Metadata Semantics and Ontologies*. Vol. 4 No. 1/2, 2009. pp. 24–33.
- [32] Heckmann D., Schwartz T., Brandherm B., Schmitz M., von Wilamowitz-Moellendorff M. GUMO — the general user model ontology. In *Proceedings of the 10th International Conference on User Modeling (UM'2005)*. LNAI 3538. Springer-Verlag. 2005. pp. 428–432.
- [33] Henze N. Personal Readers: Personalized Learning Object Readers for the Semantic Web. In *Proceeding of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*. IOS Press Amsterdam. 2005. pp. 274–281.
- [34] Hofmann T. Learning What People (Don't) Want. In *Proceedings of the 12th European Conference on Machine Learning*. Springer-Verlag. 2001. pp. 214–225.
- [35] Holohan E., Melia M., McMullen D., Pahl C. Adaptive E-Learning Content Generation based on Semantic Web Technology. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL 2005) – at the 12<sup>th</sup> International Conference on Artificial Intelligence in Education AIED 2005*. 2005. pp. 29–36.
- [36] Jeh G., Widom J. SimRank: A Measure of Structural-Context Similarity. In *ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM New York. s2002. pp 538–543.
- [37] Jovanovic J., Gašević D., Devedžić V. TANGRAM for Personalized Learning Using the Semantic Web Technologies. In *Journal of Emerging Technologies in Web Intelligence*. Vol 1, No 1. Academy Publisher. 2009. pp. 6–21.



- [38] Kalfoglou Y., Schorlemmer M. Information-flow-based ontology mapping. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*. Lecture Notes in Computer Science 2519. Springer. 2002. pp. 1132–1151.
- [39] Kalfoglou Y., Schorlemmer M. Ontology mapping: The state of the art. In *The Knowledge Engineering Review*. Vol. 18, No. 1. 2003. pp. 1–31.
- [40] Kavcic A. The role of user models in adaptive hypermedia systems. In *10<sup>th</sup> Mediterranean Electrotechnical Conference MEleCon 2000*. IEEE Computer Society. 2000. pp. 119–122.
- [41] Kay J. The um toolkit for cooperative user modelling. In *User Modelling and User-Adapted Interaction*. Vol. 4, No. 3. Springer. 1995. pp. 149–196.
- [42] Kay J. Stereotypes, student models and scrutability. In *Intelligent Tutoring Systems*. LNCS 1839. Springer Berlin / Heidelberg. 2000. pp. 19–30.
- [43] Kay J., Lum A. Ontology-based user modeling for the semantic web. In *10<sup>th</sup> International Conference on User Modeling (UM'05): Workshop 8*. 2005. pp. 11–19.
- [44] Kobsa A. User modeling: Recent work, prospects and hazards. In *M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, editors, Adaptive user interfaces: Principles and practice*. 1993. pp. 111–128.
- [45] Maedche A., Staab S. Measuring Similarity between Ontologies. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*. LNCS 2472. Springer. 2002. pp 15–21.
- [46] McGuinness D. L., Fikes R., Rice J., Wilder S. The Chimaera Ontology Environment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press. 2000. pp. 1123–1124.
- [47] Melnik S., Garcia-Molina H., Rahm E. Similarity Flooding: A Versatile Graph Matching Algorithm (Extended Technical Report). Stanford InfoLab. 2001. [online 11. 6. 2011] <http://ilpubs.stanford.edu:8090/497/>
- [48] Mitra M., Noy N. F., Jaiswal A.R. OMEN: A Probabilistic Ontology Mapping Tool. In *The Semantic Web – ISWC 2005*. LNCS 3729. Springer. 2005. pp 537–547.

- [49] Morville P. *Ambient Findability: What We Find Changes Who We Become*. O'Reilly Media, 2005, ISBN 978-0596007652
- [50] Niles I., Pease A. Towards a Standard Upper Ontology. In *Proceedings of FOIS '01 Proceedings of the international conference on Formal Ontology in Information Systems*. ACM New York. 2001. pp. 2–9.
- [51] Nebel I., Smith B., Paschke R. A user profiling component with the aid of user ontologies. In *Proceedings of Workshop Learning - Teaching - Knowledge - Adaptivity (LLWA 03)*. 2003.
- [52] Nekula J., Šaloun P., Velart Z., Klimánek P. Učební příklady – personalizovaný adaptivní web. In *Sborník konference WIKT2010*. ÚI SAV Bratislava. 2010. pp. 22–25.
- [53] Newcomb S. A Perspective on the Quest for Global Knowledge Interchange. In *XML Topic Maps – Creating and Using Topic Maps for the Web (Jack Park and Sam Hunting, eds.)*. Addison-Wesley. 2003. pp. 31–50.
- [54] Noy N.F., Musen M. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the 17<sup>th</sup> National Conference on Artificial Intelligence (AAAI'00)*. 2000. pp. 450–455.
- [55] Noy N.F., Musen M. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*. 2001. pp 63–70.
- [56] Page L., Brin S., Motwani R., Winograd T. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford InfoLab. 1999.
- [57] Philpot A., Fleischman M., Hovy E.H. . Semi-Automatic Construction of a General Purpose Ontology. In *Proceedings of the International Lisp Conference*. 2003. pp. 1–8.
- [58] Rich E. User modeling via stereotypes. In *Readings in intelligent user interfaces*. Morgan Kaufmann Publishers Inc. 1998. pp. 329–342. ISBN 1-55860-444-8.
- [59] Šaloun P., Velart Z. Adaptive ontology-based navigation. In *Proceedings of A3H: 6<sup>th</sup> International Workshop on Authoring of Adaptive and Adaptable Hypermedia*. Springer. 2008. pp. 1–4.

- [60] Šaloun P., Velart Z. Personalizace a navigace založena na ohodnocení prostoru konceptů. In *WIKT 2009 - 4<sup>th</sup> Workshop on Intelligent and Knowledge oriented Technologies 2009*. 2009. pp. 44–48.
- [61] Šaloun P., Velart Z. Concept Space Rating for Personalization of Learning Materials Based on Relations. In *Proceedings of 4<sup>th</sup> International Workshop on Semantic Media Adaptation and Personalization*. IEEE Computer Society. 2009. pp. 67–72.
- [62] Šaloun P., Velart Z. Evaluation of Concept Space Rating Algorithm and Navigation Scheme. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE Computer Society. 2010. pp. 288–291.
- [63] Šaloun P., Velart Z., Nekula J. Navigation over Multilingual Content using one Concept Space: Controlled Experiment. In *Fifth International Workshop on Semantics Media Adaptation and Personalization SMAP 2010*. IEEE Computer Society. 2010. pp. 1–6.
- [64] Šaloun P., Klimánek P., Velart Z. Semiautomatic domain model building from text-data. In *SMAP 2011 - 6th International Workshop on Semantic media adaptation and personalization*. 2011.
- [65] Šimko M., Bieliková M. Automatic Concept Relationships Discovery for an Adaptive E-course. In *EDM 2009 - Educational Data Mining 2009, 2nd International Conference On Educational Data Mining*. 2009. pp. 171–179.
- [66] Šimko M., Bieliková M. Automated Educational Course Metadata Generation Based on Semantics Discovery. In *Proceedings of 4<sup>th</sup> European Conference on Technology Enhanced Learning, ECTEL 2009*. LNCS 5794. Springer. 2009. pp. 99–105.
- [67] Šimko M., Barla M., Bieliková M. ALEF: A Framework for Adaptive Web-Based Learning 2.0. In *Proceedings of IFIP Advances in Information and Communication Technology* Vol. 324/2010. Springer. 2010. pp. 367–378.
- [68] Sosnovsky S., Brusilovsky P. Layered Evaluation of Topic-Based Adaptation to Student Knowledge. In *Proceedings of Fourth Workshop on the Evaluation of Adaptive Systems at 10th International User Modeling Conference, UM 2005*, 2005. pp. 47–56.

- [69] Studer R., Benjamins V.R., Fensel D. Knowledge engineering: Principles and methods. In *Data Knowledge Engineering*. Vol. 25, No. 1–2. 1998. pp. 161–197.
- [70] Tiňo P., Polčicová G. Topographic organization of user preference patterns in collaborative filtering. In *Neural network world*. Vol. 13, No. 3. 2003. pp. 311–324.
- [71] Understanding Metadata. *published by NISO (National Information Standards Organization)*. 2004. ISBN 1-880124-62-9.
- [72] van Meteren R., van Someren M. Using content-based filtering for recommendation. In *G. Potamias, V. Moustakis, and M. van Someren, editors, ECML/MLNET Workshop on Machine Learning and the New Information Age*. 2000. pp. 47–56.
- [73] Velart Z., Šaloun P. Využití adaptivního hypermediálního systému AHA! při výuce programovacího jazyka C++. In *Objekty 2005*. 2005. pp. 280–288
- [74] Velart Z., Šaloun P. User Behavior Patterns in the Course of Programming in C++. In *Proceedings of the joint international workshop on Adaptivity, personalization & the semantic web*. ACM Digital Library. 2006. pp. 41–44.
- [75] Velart Z., Šaloun P. Ontology Based Course Navigation. In *HT'07 (Hypertext 2007) Proceedings*. ACM New York. 2007. p. 151–152.
- [76] Velart Z., Nekula J., Šaloun P. Experimentální adaptivní vícejazyčný webový systém. In *ITAT 2010 - Informačné Technológie – Aplikácie a Teória*. 2010. pp. 129–130.
- [77] Velart Z., Šalou, P., *Adaptivní systém XAPOS*. „Autorizovaný software”, podán k registraci do RIV, 2011, VŠB–TU Ostrava.  
<http://arg.vsb.cz/xapos/>.
- [78] Yudelso M., Brusilovsky P., Sosnovsky S. Accessing Interactive Example with Adaptive Navigation Support. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*. IEEE Computer Society. 2004. pp. 842–843.
- [79] Yudelso M., Gavrilova T., Brusilovsky P. Towards user modeling meta-ontology. In *Proceedings of the 10<sup>th</sup> International Conference on User Modeling*. Springer. 2005. pp. 448–452.

- [80] Yudelso M., Brusilovsky P. NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples. In *Proceedings of 12<sup>th</sup> International Conference on Artificial Intelligence in Education, AIED 2005*. Amsterdam: IOS Press. 2005. pp. 710–717.
- [81] Vander T.W. Understanding Folksonomy (Tagging that Works). Brighton England 2006, [online: 29. 5. 2011] [http://s3.amazonaws.com/2006presentations/dconstruct/Tagging\\_in\\_RW.pdf](http://s3.amazonaws.com/2006presentations/dconstruct/Tagging_in_RW.pdf)



Část III

**Přílohy**





## Příloha A

# Vlastní práce vztahující se k tématu disertace

- Velart Z. Utilization of adaptive hypermedia for learning programming. In *Proceedings of WOFEX 2006*. 2006. pp. 440–445. ISBN 80-248-1152-9
- Velart Z., Šaloun P., Kalinská M. Znalostmi řízený průchod kurzem. In *1<sup>st</sup> Workshop on Intelligent and Knowledge oriented Technologies - WIKT 2006 Proceedings*. 2006. pp. 78–80. ISBN 978-80-969202-5-9
- Velart Z., Šaloun P. Ontology Based Course Navigation. In *HT'07 (Hypertext 2007) Proceedings*. ACM New York. 2007. p. 151–152. ISBN 978-1-59593-820-6
- Velart Z. Ontology based course structure for adaptive web-based system. In *Proceedings of WOFEX 2007*. 2007. pp. 333–338.
- Šaloun P., Velart Z. Adaptive ontology-based navigation. In *A3H: 6<sup>th</sup> International Workshop on Authoring of Adaptive and Adaptable Hypermedia*. Springer. 2008. pp. 1–4.
- Velart Z., Šaloun P. Analýza obsahu kurzu, návrh navigace vytvořené z textu při použití pojmů ontologie. In *Proceedings of Znalosti 2008*. 2008. pp. 391–394. ISBN 978-80-227-2827-0
- Velart Z., Šaloun P. Personalized recommendation scheme based on content and knowledge hierarchy. In *Proceedings of WIKT 2008 – 3<sup>rd</sup> Workshop on Intelligent*

*and Knowledge Oriented Technologies*. 2008. pp. 9–12.

- Velart Z., Šaloun P. Navigace množinou výukových objektů s využitím prostoru konceptů. In *Proceedings of Znalosti 2009*. 2009. pp. 329–332.
- Šaloun P., Velart Z. Concept Space Rating for Personalization of Learning Materials Based on Relations. In *Proceedings of 4<sup>th</sup> International Workshop on Semantic Media Adaptation and Personalization*. IEEE Computer Society. 2009. pp. 67–72.
- Šaloun P., Velart Z. Personalizace a navigace založena na ohodnocení prostoru konceptů. In *WIKT 2009 - 4<sup>th</sup> Workshop on Intelligent and Knowledge oriented Technologies 2009*. 2009. pp. 44–48.
- Šaloun P., Velart Z., Nekula J. Navigation over Multilingual Content using one Concept Space: Controlled Experiment. In *Fifth International Workshop on Semantics Media Adaptation and Personalization SMAP 2010*. IEEE Computer Society. 2010. pp. 1–6.
- Šaloun P., Velart Z. Evaluation of Concept Space Rating Algorithm and Navigation Scheme. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE Computer Society. 2010. pp. 288–291.
- Velart Z., Nekula J., Šaloun P. Experimentální adaptivní vícejazyčný webový systém. In *ITAT 2010 - Informačné Technológie – Aplikácie a Teória*. 2010. pp. 129–130.
- Šaloun P., Velart Z., Nekula J. Towards automated navigation over multilingual content. Submitted to *Semantic Hyper/Multi-media Adaptation: Schemes and Applications*. Studies in Computational Intelligence. Springer. 2011.
- Velart Z., Šaloun P., *Adaptivní systém XAPOS*. „Autorizovaný software”, podán k registraci do RIV, 2011, VŠB–TU Ostrava.  
<http://arg.vsb.cz/xapos/>

## Příloha B

# XAPOS

Adaptivní webový systém XAPOS [76], který jsme vyvinuli pro ověření prezentovaného způsobu navigace je implementován a provozován jako webový. Uživatel k němu může přistupovat pomocí svého oblíbeného prohlížeče podporující současné webové standardy.

Systém XAPOS nabízí uživatelům výukové kurzy. Systém požaduje, aby každý uživatel byl přihlášen a je možné jej takto jednoznačně identifikovat. Navigace v systému je prováděna pro každého uživatele samostatně.

XAPOS implementuje architekturu AHAM a skládá se ze tří základních součástí:

- Model uživatele – slouží k uchování informací o uživateli systému. V XAPOSu je model uživatele reprezentován databázovým úložištěm, kde jsou uchovávány základní údaje o uživateli, jako jsou jméno a kontaktní údaje, jeho nastavení v systému a informace o aktivitě uživatele v jednotlivých kurzech (navštívení jednotlivých výukových objektů s časovým razítkem) a dosažených znalostech (množina dosažených znalostí obsahující naučené koncepty). Součástí modelu je také log aktivity uživatele – informace o přihlášení a odhlášení, absolvovaných testech apod. Pokud se uživatel v průběhu práce ze systému odhlásí, případně dojde k přerušení spojení po opětovném přihlášení, je automaticky nastaven na poslední navštívenou stránku v kurzu.
- Model domény – představuje problémovou doménu, která je prezentována uživateli systému. V našem případě model domény představuje prostor konceptů společně s ohodnocením jednotlivých konceptů a výukovými objekty. Výukový objekt je reprezentován HTML stránkou s odkazovanými obrázky či jiným multimediálním materiálem, respektive ukázkový příkladu zapsán jako XML soubor v námi definovaném

formátu. Model domény je v systému implementován kombinací databázového úložiště a souborového systému.

- Model adaptace – provádí adaptaci a úpravu navigace uživatele. V XAPOSu je implementován algoritmem pro ohodnocení prostoru konceptů a dříve popsaným způsobem navigace s metrikami používanými pro adaptaci navigace uživatele. Každému uživateli může být přiřazena jiná metria.

## Vícejazyčnost

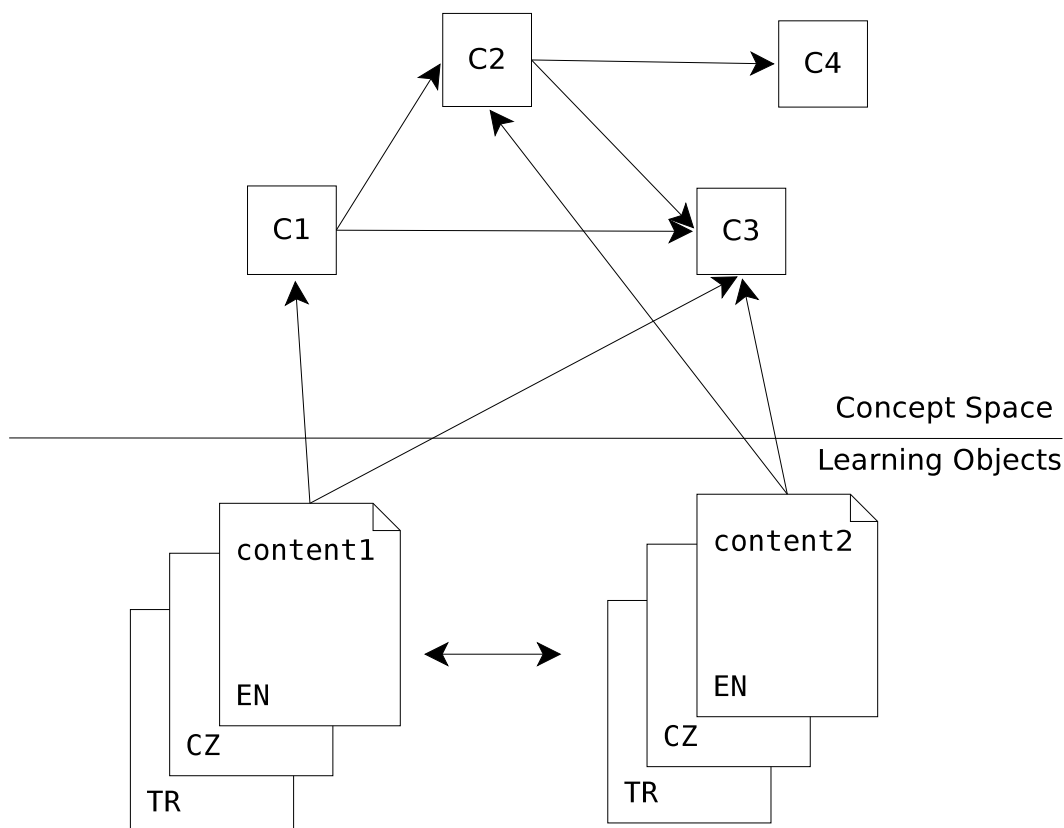
Vícejazyčnost je v XAPOSu řešena lokalizací a překladem obsahu i prostředí. Vícejazyčnost prostředí je řešena dostupnými prostředky frameworku Struts, ve kterém je XAPOS napsán. Pro každý jazyk, který systém podporuje, máme properties soubor s překladem. Struts zajišťuje automatické načtení properties, a podle zvoleného jazyka v GUI je mu zobrazována příslušná jazyková varianta. Pokud uživatel v GUI nevybere žádný jazyk automaticky se bere v úvahu nastavení jeho prohlížeče, podle kterého je mu zobrazován příslušný jazyk.

Výukový materiál může být vícejazyčný - jednotlivé jazykové varianty stejného obsahu jsou verzemi stejného výukového objektu (viz obrázek 18). Tímto způsobem je uživateli umožněno kurz procházet v libovolném dostupném jazyce, a také je mu nabídnuta možnost přepínání jazyka v průběhu kurzu - uživatel může například kurz studovat v anglickém jazyce, a pokud nerozumí některé látce, může se přepnout do jiného dostupného (českého) jazyka pro porovnání. Informace o zvoleném jazyce se ukládá do logu při navštívení výukového objektu. Pokud není výukový objekt v systému dostupný ve zvoleném jazyce, tak se uživateli zobrazí v implicitním jazyce, který je povinně určen pro celý kurz. Znamená to, že minimálně v daném jazyce musí být výukový obsah každého výukové objekty vytvořen.

Výukový obsah výukového objektu je ukládán do souborového systému. Každá jazyková mutace je uložena ve zvláštním souboru. V databázi jsou uloženy informace o vazbě výukových objektů na odpovídající soubory.

Každý výukový objekt má v databázi uložen název a anotaci v příslušných jazycích. V systému jsou tyto informace automaticky načítány podle zvoleného jazyka uživatelem společně s výukovým objektem. Název se uživateli zobrazuje v položkách v menu a anotace po najetí myši nad příslušnou položku.

Obdobně je v XAPOSu řešeno ukládání a práce s dalšími vícejazyčnými objekty, jako jsou testy, dotazníky a podobně.



Obrázek 18: Vícejazyčný obsah

## Testy

Testy jsou v procesu navigace jedním z důležitých rozhodovacích prvků. Každý výukový objekt může mít přiřazen test. Test je uživateli zpravidla prezentován na konci výukového objektu, tak aby bylo možné ověřit jeho dosažené znalosti. Test může být spuštěn na základě explicitního požadavku uživatele nebo může být definován jako povinný.

Každý test se skládá z více testovacích otázek. V současné době podporuje systém XA-POS otázky typu výběr odpovědi. Každá otázka má definovanou množinu konceptů, kterou pokrývá. Při vytváření testů by měl mít autor na paměti, aby koncepty, které jednotlivé otázky pokrývají byly z množiny konceptů daného výukového objektu.

Po zodpovězení testu uživatelem následuje jeho vyhodnocení. Na základě toho, jak uživatel odpověděl na jednotlivé otázky jsou asociované koncepty do jeho množiny dosažených znalostí přidány nebo odebrány.

## Logování a nastavení

Všechny akce uživatele v systému se ukládají do systémového logu. U každého záznamu se ukládají jednak informace o uživateli, který danou akci vyvolal, informace o aktuálně zvoleném jazyce a také časové razítko, aby bylo možno zrekonstruovat pohyb uživatele v systému. Současně jsou informace využívány pro navigaci uživatele.

Každý kurz v systému má definovány implicitní hodnoty základních nastavení, jako je například implicitní jazyk, počet zobrazovaných položek menu, implicitní metrika pro personalizaci apod. V profilu uživatele je možné některé z těchto nastavení individuálně upravit pro uživatele, jmenujme například počet zobrazovaných položek menu, zvolená metrika (uživatel nemůže měnit, nastaví administrátor) a další.

## Technologie

Systém je implementován jako webový systém a je postaven na MVC architektuře. XAPOS je implementován v Javě.

Pro zpracování komunikace uživatele se serverem a přípravu výsledných stránek, které se budou zobrazovat uživateli, používáme framework Struts<sup>1</sup> s rozšířeními Tiles – umožňuje skládáním částí stránek vytvořit výslednou stránku prezentovanou uživateli a Convention Plugin – usnadňuje automatizované mapování URL na akce pomocí zápisu konfiguračních informací pomocí anotací do příslušných tříd. Jednotlivé stránky jsou napsány v JSP za pomoci dostupných tagů z frameworku Struts. Při implementaci využíváme jednak standardních HTTP spojení tak také Ajax.

Prostor konceptů a průchod uživatele kurzem zobrazujeme pomocí projektu RaVis<sup>2</sup>, který umožňuje vizualizaci dat, jako jsou sítě, organizační struktury a podobně. RaVis prohlížeč je napsán jako Flash aplikace a vkládá se do kódu stránky. Data jsou dodána ve formě XML souboru, který má strukturu dle DTD dodávaného s projektem.

Systém XAPOS je možné provozovat na aplikačním serveru podporujícím technologie Servlet 5.2 a JavaServer Pages 2.1. V současné době systém XAPOS provozujeme na aplikačním serveru Tomcat 6.x<sup>3</sup>.

Pro persistenci dat používáme databázi. Pro přístup k datovému úložišti používáme

---

<sup>1</sup><http://struts.apache.org/>

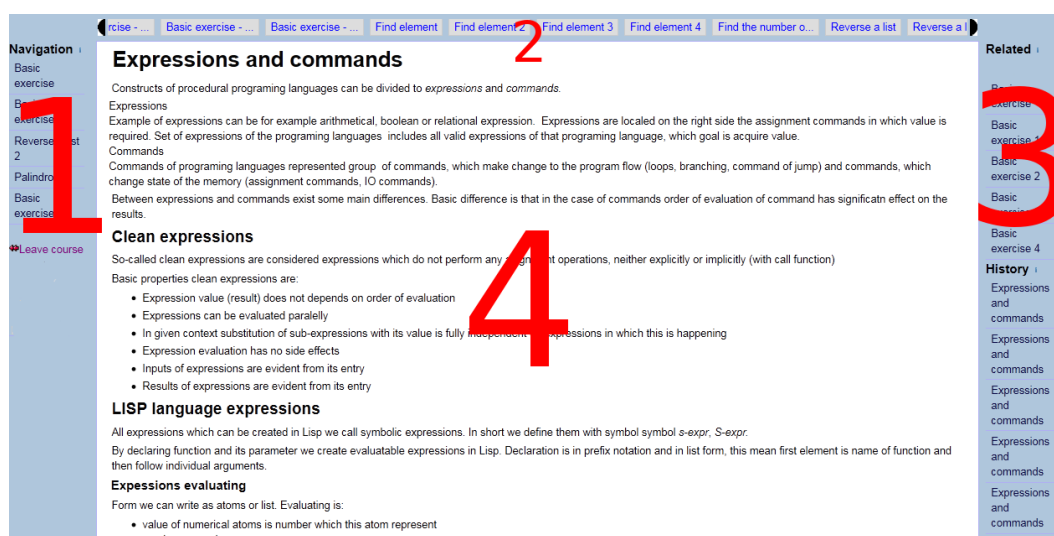
<sup>2</sup><http://code.google.com/p/birdeye/wiki/RaVis>

<sup>3</sup><http://tomcat.apache.org/>

ORM Hibernate<sup>4</sup>. V současné implementaci používáme databázový server MySQL. Díky použití Hibernate nepředstavuje případný přechod na jiný databázový server žádný zásadní problém. Pro uložení výukového obsahu (HTML stránky, související obrázky, ukázkové příklady, případně další multimediální materiál) používáme souborový systém.

## GUI

V rozhraní systému XAPOS (viz obrázek 19) se uživateli zobrazuje v části označené 1 menu, odpovídající personalizované navigaci – výukové objekty jsou seříděny systémem podle vhodnosti pro uživatele.



Obrázek 19: Rozvržení GUI systému

V části označené 2 se nachází menu s obsahem celého kurzu. Uživatel zde také nalezne přepínání jazyka, ve které se budou zobrazovat popisky v systému a také výukový materiál. Přepnutím jazyka dojde k aktualizaci právě zobrazované stránky do zvoleného jazyka. V pravé části označené 3 se nachází menu „Související“, kde se zobrazují související výukové objekty s daným výukovým objektem a menu „Historie“, kde se zobrazuje uživatelova historie v kurzu.

Výukový obsah, který je asociován s daným výukovým objektem se zobrazuje ve střední části obrazovky označené jako 4.

<sup>4</sup><http://www.hibernate.org/>

Všechny položky v menu jsou zobrazovány ve zvoleném jazyce. Pokud nějaká informace není dostupná ve vybraném jazyce, je nahrazena informací v implicitním jazyce kurzu. Najetím myši na libovolnou položku v menu se zobrazí její anotace - krátký popis daného výukového objektu. Pro zobrazení zdrojových kódů ukázkových příkladů je použita knihovna `google-code-prettify`<sup>5</sup>, určená ke zvýraznění syntaxe.

---

<sup>5</sup><http://code.google.com/p/google-code-prettify/>