

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5220-36263

Bc. Martin Labaj

**ODPORÚČANIE A KOLABORÁCIA NA ZÁKLADE
IMPLICITNEJ SPÄTNEJ VÄZBY**

Diplomová práca

Vedúci práce: prof. Ing. Mária Bieliková, PhD.

máj, 2011

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5220-36263

Bc. Martin Labaj

ODPORÚČANIE A KOLABORÁCIA NA ZÁKLADE IMPLICITNEJ SPÄTNEJ VÄZBY

Diplomová práca

Študijný program: Softvérové inžinierstvo

Študijný odbor: 9.2.5 Softvérové inžinierstvo

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva,
FIIT STU v Bratislave

Vedúci práce: prof. Ing. Mária Bieliková, PhD.

máj, 2011

Zadanie diplomovej práce

Meno študenta: **Bc. Labaj Martin**

Študijný program: Softvérové inžinierstvo

Študijný odbor: Softvérové inžinierstvo

Názov práce: **Odporúčanie a kolaborácia prostredníctvom implicitnej identifikácie v sociálnom kontexte**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie, zodpovedúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riadte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti alebo o priebežné výsledky Vášho riešenia alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnete zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva FIIT STU v Bratislave

Vedúci práce: **prof. Ing. Bieliková Mária, PhD.**

Termíny odovzdania:

podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti:
home > Informácie o > štúdiu > organizácia štúdia > diplomový projekt

V Bratislave dňa 15. 2. 2010



prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky a softvérového
inžinierstva

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce¹

Študent

Meno, priezvisko, tituly:	Martin Labaj, Bc.
Študijný program:	Softvérové inžinierstvo
Kontakt:	martin.labaj@computer.org, +421 902 237 968

Výskumník:

Meno, priezvisko, tituly:	Mária Bieliková, prof. Ing. PhD.
----------------------------------	----------------------------------

Projekt:

Názov:	Odporúčanie a kolaborácia prostredníctvom implicitnej identifikácie v sociálnom kontexte
Miesto vypracovania:	Ústav informatiky a softvérového inžinierstva, FIIT STU, Bratislava
Oblasť problematiky²:	Implicitná spätná väzba, prispôsobovanie obsahu na webe

Text zadania

Identifikácia významných fragmentov textu, čiže častí, ktoré sú pre používateľa zaujímavé alebo náročné, je dôležitý prvok pre činnosti, akými sú napríklad odporúčanie alebo revízie textu. Takéto informácie možno získať prostredníctvom explicitnej spätnej väzby, bežným spôsobom býva hodnotenie používateľom. Explicitné prostriedky však často narúšajú prirodzený priebeh práce používateľa a používateľ na ne nemusí reagovať pravdivo. Zaujímavými a aktuálnymi sú preto implicitné metódy založené na sledovaní konania používateľa.

Analyzujte problematiku získavania informácií o spôsobe práce používateľa s predloženým dokumentom s využitím implicitnej spätnej väzby. Navrhnite metódu na identifikáciu významných fragmentov textov, ktorá využíva implicitnú spätnú väzbu. Príkladom môže byť sledovanie zobrazených úsekov textu (angl. scrolling) v súvislosti s časom, pričom štatistickým vyhodnotením týchto údajov získaných od množstva používateľov možno pomerne presne identifikovať významné fragmenty. Ďalej navrhnite spôsob ako využiť získané informácie v sociálnom kontexte pri odporúčaní obsahu. Keďže zahrnutie časového aspektu je citlivé na reálnu činnosť používateľa, treba zvážiť možnosť, že používateľ sa v danom čase venuje iným aktivitám a tento čas odfiltrovať, napr. analýzou zamerania pohľadu používateľa, ktorá môže byť zároveň použitá aj na spresnenie identifikácie. Navrhnuté riešenie overte v kontexte výučbových textov v doméne výučby programovania, prípadne v otvorenom informačnom priestore (web).

150-200 slov, ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

¹ Veľkosť jednotlivých polí pre vyplňanie nemožno meniť. Návrh zadania vytlačiť obojstranne na jeden list papiera.

² Identifikácia oblastí v rámci odboru štúdia, na ktorú sa projekt primárne viaže

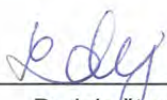
Literatúra

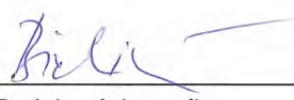
- Claypool, M., Le, P., Wased, M., and Brown, D. (2001). Implicit interest indicators. In Proceedings of the 6th international Conference on intelligent User interfaces (Santa Fe, New Mexico, United States, January 14 - 17, 2001). IUI '01. ACM, New York, NY, 33-40.
- Atterer, R., Wnuk, M., and Schmidt, A. (2006). Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM, New York, NY, 203-212.
- Xu, S., Jiang, H., and Lau, F. C. (2009). User-oriented document summarization through vision-based eye-tracking. In Proceedings of the 13th international Conference on intelligent User interfaces (Sanibel Island, Florida, USA, February 08 - 11, 2009). IUI '09. ACM, New York, NY, 7-16.

2-3 vedecké zdroje, každý na samostatnom riadku a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uvedte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval Bc. Martin Labaj, konzultovala a osvojila si ho prof. Ing. Mária Bieliková, PhD. a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 20.1.2009

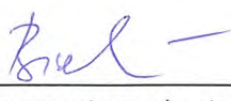

Podpis študenta


Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / nie³

Dňa: 9.2.2010


Podpis garanta predmetov

³ Nehodiace sa prečiarknite

Návrh zadania diplomovej práce

Revízia č.¹: 1

Študent

Meno, priezvisko, tituly:	Martin Labaj, Bc.
Študijný program:	Softvérové inžinierstvo
Kontakt:	martin.labaj@computer.org, +421 902 237 968

Výskumník:

Meno, priezvisko, tituly:	Mária Bieliková, prof. Ing. PhD.
----------------------------------	----------------------------------

Projekt:

Názov:	Odporúčanie a kolaborácia na základe implicitnej spätnej väzby
Miesto vypracovania:	Ústav informatiky a softvérového inžinierstva, FIIT STU, Bratislava
Oblasť problematiky²:	Implicitná spätná väzba, prispôsobovanie obsahu na webe

Text zadania

Identifikácia významných fragmentov textu, čiže častí, ktoré sú pre používateľa zaujímavé alebo náročné, je dôležitý prvok pre činnosti, akými sú napríklad odporúčanie alebo revízie textu. Takéto informácie možno získať prostredníctvom explicitnej spätnej väzby, bežným spôsobom býva hodnotenie používateľom. Explicitné prostriedky však často narúšajú prirodzený priebeh práce používateľa a používateľ na ne nemusí reagovať pravdivo. Zaujímavými a aktuálnymi sú preto implicitné metódy založené na sledovaní konania používateľa.

Analyzujte problematiku získavania informácií o spôsobe práce používateľa s predloženým dokumentom s využitím implicitnej spätnej väzby. Navrhnite metódu na identifikáciu významných fragmentov textov, ktorá využíva implicitnú spätnú väzbu. Príkladom môže byť sledovanie zobrazených úsekov textu (angl. scrolling) v súvislosti s časom, pričom štatistickým vyhodnotením týchto údajov získaných od množstva používateľov možno pomerne presne identifikovať významné fragmenty. Ďalej navrhnite spôsob ako využiť získané informácie v sociálnom kontexte pri odporúčaní obsahu. Keďže zahrnutie časového aspektu je citlivé na reálnu činnosť používateľa, treba zvážiť možnosť, že používateľ sa v danom čase venuje iným aktivitám a tento čas odfiltrovať, napr. analýzou zamerania pohľadu používateľa, ktorá môže byť zároveň použitá aj na spresnenie identifikácie. Navrhnuté riešenie overte v kontexte výučbových textov v doméne výučby programovania, prípadne v otvorenom informačnom priestore (web).

150-200 slov, ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

¹ Uvedie sa poradové číslo revízie

² Identifikácia oblasti v rámci odboru štúdia, na ktorú sa projekt primárne viaže

Literatúra

- Claypool, M., Le, P., Wased, M., and Brown, D. (2001). Implicit interest indicators. In Proceedings of the 6th international Conference on intelligent User interfaces (Santa Fe, New Mexico, United States, January 14 - 17, 2001). IUI '01. ACM, New York, NY, 33-40.
- Atterer, R., Wnuk, M., and Schmidt, A. (2006). Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM, New York, NY, 203-212.
- Xu, S., Jiang, H., and Lau, F. C. (2009). User-oriented document summarization through vision-based eye-tracking. In Proceedings of the 13th international Conference on intelligent User interfaces (Sanibel Island, Florida, USA, February 08 - 11, 2009). IUI '09. ACM, New York, NY, 7-16.

2-3 vedecké zdroje, každý na samostatnom riadku a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uvedte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval Bc. Martin Labaj, konzultovala a osvojila si ho prof. Ing. Mária Bielíková, PhD. a súhlasí, že bude takýto projekt viesť.

V Bratislave dňa 27.1.2011



Podpis študenta



Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / nie³

Dňa: 4.2.2011



Podpis garanta predmetov

³ Nehodiace sa prečiarknite

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: SOFTVÉROVÉ INŽINIERSTVO

Autor: Bc. Martin Labaj

Diplomová práca: Odporúčanie a kolaborácia na základe implicitnej spätnej väzby

Vedúci diplomovej práce: prof. Ing. Mária Bieliková, PhD.

máj, 2011

V oblasti elektronickej výučby aj otvoreného webu je identifikácia náročných a/alebo zaujímavých častí textov užitočnou funkciou nápomocnou pri úlohách, akými sú revízia textu, zobrazenie častí, na ktoré je potrebné sa zmerať alebo poskytnutie pomoci prispôsobenej individuálne študentovi. Metódy, ktoré tieto informácie získavajú z priamej interakcie s používateľom, napríklad prostredníctvom požiadania používateľa o ohodnotenie častí dokumentu, však môžu viesť k odpútavaniu pozornosti počas procesu učenia sa a tiež vyžadujú dobrovoľné zapojenie sa používateľa a pravdivé odpovede. V tejto práci sa venujeme sledovaniu implicitných indikátorov záujmu, najmä posúvaniu dokumentu používateľom a zameraniu pohľadu používateľa. Na ich využitie sme navrhli metódu na identifikáciu zaujímavých fragmentov dokumentu a sledovanie fragmentov webovej aplikácie. Pre túto metódu sme navrhli viaceré scenáre využitia a niektoré z nich overili experimentom so zapojením 34 študentov fakulty. Prínos práce spočíva vo využití možnosti sledovania pohľadu používateľov vo webovej aplikácii nenáročnými prostriedkami, lacnou webovou kamerou, v neriadenom prostredí.

Annotation

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: SOFTWARE ENGINEERING

Author: Bc. Martin Labaj

Master's Theses: Recommendation and Collaboration based on Implicit Feedback

Supervisor: Professor Mária Bieliková

2011, May

In the field of e-learning and open web, the identification of difficult and/or interesting parts of learning text can be useful feature for tasks like rewriting, showing where to focus or providing help to student adaptively. Methods which extract this information by directly interacting with the user for example by asking him to rate document fragments can lead to distraction while learning and require that users answer truthfully. In this work, we focus on implicit interest indicators, most importantly document scrolling and user's gaze. With these, we designed a method for identification of interesting fragments of document and tracking of fragments of web application. We proposed several scenarios for use of this method and evaluated some of them in experiment with 34 students of our faculty. Main contribution of this work is in the utilization of users' gaze in the web environment, in common settings and with only commodity hardware.

Obsah

1	Úvod.....	1
2	Monitorovanie používateľa pri práci s informáciami na webe	3
2.1	Implicitná spätná väzba.....	4
2.2	Posúvanie zobrazeného dokumentu	5
2.3	Zisťovanie prítomnosti používateľa.....	7
2.4	Spôsoby sledovanie zamerania pohľadu používateľa.....	9
2.5	Sledovanie zamerania pohľadu používateľa pomocou webovej kamery.....	10
2.5.1	FTR_BY_WLP	11
2.5.2	ITU Gaze Tracker	11
2.5.3	myEye	12
2.5.4	openEyes	12
2.5.5	OpenGazer	13
2.5.6	Zhrnutie.....	13
2.6	Testovanie existujúcich riešení sledovania pohľadu používateľa pomocou webovej kamery	14
2.6.1	FTR_BY_WLP	14
2.6.2	GazeTracker.....	15
2.6.3	OpenGazer (NetGazer)	16
2.6.4	Zhodnotenie sledovania pohľadu používateľa.....	17
2.7	Spôsoby zberu implicitnej spätnej väzby.....	17
2.7.1	Priama (webová) implementácia.....	18
2.7.2	Proxy server	18
2.7.3	Zásuvný modul do prehliadača	19
2.7.4	Samostatná aplikácia.....	19
2.8	Diskusia.....	20
3	Doména vzdelávania na adaptívnom webe – systém ALEF.....	21
3.1	Predchádzajúca aplikácia implicitnej spätnej väzby v systéme ALEF.....	23
4	Metóda na získavanie implicitnej spätnej väzby.....	25
4.1	Získanie údajov	25
4.2	Interpretácia údajov na fragmenty	26
4.2.1	Rozdelenie dokumentu a používateľského rozhrania na fragmenty	26
4.2.2	Vzťah spätnej väzby k fragmentom	27
4.3	Zostavenie charakteristík fragmentov	27
4.4	Diskusia.....	28
5	Využitie informácií o práci používateľa s dokumentom	29

5.1	Významné fragmenty	29
5.1.1	Priama vizualizácia.....	29
5.1.2	Sumarizácia	30
5.1.3	Adaptívny posuvník	30
5.2	Sprievodca prácou v systéme	30
5.3	Obohatená komunikácia používateľov	31
6	Návrh realizácie a implementácia metódy	33
6.1	Prepojenie sledovania pohľadu s webovým prehliadačom	33
6.2	Aplikácia na sledovanie pohľadu GazerTracker	36
6.3	Rozšírenie prehliadača UTrack	37
6.4	Interakcia rozšírenia prehliadača s webovou aplikáciou	39
6.5	Adaptívna explicitná spätná väzba v systéme ALEF	41
6.5.1	Adaptívna explicitná spätná väzba a rozšírenie UTrack	42
6.6	Vizualizácia významnosti fragmentov	43
7	Experimentálne overenie získavania a využitia implicitnej spätnej väzby	45
7.1	Návrh experimentu	45
7.2	Technické zabezpečenie experimentu	45
7.3	Realizácia experimentu	46
7.4	Vyhodnotenie experimentu	47
7.4.1	Adaptívne otázky.....	47
7.4.2	Ohodnotenie fragmentov dokumentu	49
7.4.3	Hodnotenie od účastníkov	52
8	Záver	53
	Použitá literatúra.....	55

Príloha A – Prihláška projektu do súťaže o štipendium

Príloha B – Príspevok publikovaný v rámci konferencie 6th E-learning Conference

Príloha C – Príspevok zaslaný na konferenciu EC-TEL 2011

Príloha D – Plagát prezentovaný na konferencii IIT.SRC 2011

Príloha E – Technická dokumentácia

Príloha F – Dokumentácia k experimentom

Príloha G – Používateľská príručka systému ALEF

Príloha H – Používateľská príručka rozšírenia UTrack

Príloha I – Obsah elektronickej prílohy

1 ÚVOD

Nielen v oblasti elektronickej výučby, ale i v otvorenom priestore webu a ďalších doménach je vhodné a užitočné identifikovať dôležité, náročné, zaujímavé časti obsahu. Z takto identifikovaných fragmentov obsahu následne môžeme vychádzať v rôznych aplikáciách, ako napríklad pri revidovaní textu, odporúčaní dôležitých častí, sumarizovaní dokumentu. Častým priamočiarym prístupom je využitie explicitnej spätnej väzby od používateľa (čitateľa dokumentu): požiadanie používateľa o hodnotenie kvality alebo náročnosti obsahu, zodpovedanie kontrolných otázok na overenie jeho porozumenia obsahu, atď. Tieto metódy však závisia od dobrovoľnej spolupráce používateľa a od správnosti jeho hodnotenia, ktoré nemusí ani vedieť sám vyhodnotiť, a zároveň ho môžu vyrušovať pri práci s obsahom, čo môže zvlášť v doméne podpory vzdelávania odpútať od cieľov prezentácie obsahu – učenia sa.

Zaujímavými a aktuálnymi sú preto implicitné metódy získavania spätnej väzby, ktoré spočívajú v sledovaní konania používateľa a následnom odhadovaní jeho zvykov, zámerov, charakteristík. Implicitné metódy bývajú menej presné než explicitné, pretože pasívne pozorovanie aktivít používateľa nemusí umožniť jednoznačnú interpretáciu jeho zámerov – ak pozorujeme, že používateľ nepoužíva určitú funkciu, je to preto, lebo mu poskytuje neužitočné údaje, alebo používateľ o tejto funkcii iba nevie? Pri explicitných metódach ale tiež môže dôjsť k tomu, že používateľ niečo odpovie iba preto, lebo je vyzvaný až prinútený odpovedať a získame tak nesprávnu informáciu. Práve vďaka tomu, že používateľ nemusí aplikáciu implicitných metód na rozdiel od explicitných hodnotení ani spozorovať, tieto metódy nenarúšajú prirodzený spôsob jeho práce, čím nedochádza ku skresleniu výstupov. Implicitné metódy sú tiež nezávislé od toho, či sa používateľ rozhodne spätnú väzbu poskytnúť, pretože ju poskytuje neustále.

V prípade implicitných metód však treba zvlášť uvažovať pasívnu prácu používateľa s obsahom, ako je tomu napríklad v systéme elektronickej výučby alebo pri prehliadaní materiálov na webe po vykonaní navigácie na požadovaný dokument a pri následnom samotnom prehliadaní dokumentu. Keď používateľ prehliada už zobrazený dokument, okrem posúvania obsahu nehýbe a nekliká myšou a nestláča klávesy, nie je možné priamo aplikovať ľubovoľnú z metód, ktoré sa sústreďia na priamu interakciu používateľa.

Jednou z možností ako využiť implicitnú spätnú väzbu v tomto prípade, je sledovanie posúvania obsahu používateľom (angl. *scrolling*) a následne teda aj času, po ktorý mal používateľ zobrazenú niektorú časť obsahu. Pretože úseky čítania sú ale založené na čase a v tomto čase sa používateľ môže venovať iným aktivitám, neposúvanie obsahu neznamená automaticky jeho čítanie. Toto môžeme spresniť sledovaním prítomnosti používateľa pri počítači napríklad prostredníctvom webovej kamery, pričom pokročilejšie metódy, napríklad sledovanie zamerania pohľadu vytvárajú ďalší implicitný indikátor aplikovateľný aj v prípade pasívneho prehliadania obsahu používateľom.

Naším cieľom je využitie implicitnej spätnej väzby pri sledovaní práce používateľa s predloženým dokumentom v prostredí webovej aplikácie so zahrnutím pohľadu zisteného v bežnom prostredí prostredníctvom lacnej webovej kamery.

Cieľom tejto práce je:

- vytvoriť metódu na získavanie implicitnej a explicitnej spätnej väzby o práci používateľa s dokumentom prezentovaným na webe (špeciálne v adaptívnych systémoch),
- vytvoriť metódu na identifikáciu významných fragmentov dokumentov a základných aspektov práce používateľa s adaptívnym webovým systémom pomocou získanej spätnej väzby,
- navrhnúť možnosti využitia odvodených informácií o dokumentoch a práci používateľa,
- overiť navrhnuté riešenie v doméne adaptívnej elektronickej výučby, alternatívne v otvorenom informačnom priestore webu.

V tejto práci sa zameriavame na implicitnú spätnú väzbu vo forme posúvania zobrazeného dokumentu a sledovania prítomnosti používateľa/zamerania jeho pohľadu. Práca sa skladá z ôsmich kapitol. V druhej kapitole predstavujeme analýzu súčasného stavu oblasti sledovania činnosti používateľa najmä implicitnými prostriedkami a zisťovania pohľadu používateľa fyzickým sledovaním webovou kamerou v neriadenom prostredí. V tretej kapitole uvádzame doménu adaptívneho vzdelávania na webe a pozornosť venujeme systému ALEF, ktorý je vyvíjaný a prevádzkovaný na Fakulte informatiky a informačných technológií a ktorý sme zvolili ako prostredie pre overenie nášho riešenia. V štvrtej kapitole uvádzame nami vytvorený návrh metódy pre identifikáciu významných fragmentov na základe činnosti používateľa, včítane rozdelenia dokumentov a webových aplikácií na fragmenty. V kapitole 5 navrhujeme viaceré scenáre využitia navrhutej metódy vo webových aplikáciách. V kapitole 6 opisujeme pretvorenie navrhovanej metódy do formy implementovaného riešenia a postupne sa zaoberáme jednotlivými súčasťami riešenia: získaním pohľadu, rozšírením prehliadača, komunikáciou s webovou aplikáciou a využitím údajov vo webovej aplikácii. V siedmej kapitole opisujeme overenie riešenia prostredníctvom experimentov vykonaných s pomocou študentov predmetu Princípy softvérového inžinierstva. Dokument ďalej obsahuje ôsmu kapitolu – záver, v ktorom načrtáme ďalšie možné smerovanie.

Dokument pokračuje zoznamom použitej literatúry a deviatimi prílohami, ktoré obsahujú dokumenty a publikácie vytvorené v súvislosti s touto diplomovou prácou: úspešná prihláška do súťaže o štipendium organizovanej spoločnosťou Softec, s.r.o, článok publikovaný na konferencii 6th E-learning Conference, práve posudzovaný článok zaslaný na konferenciu EC-TEL 2011 a plagát prezentovaný na študentskej vedeckej konferencii IIT.SRC 2011. Ďalej sa medzi prílohami nachádza technická dokumentácia k riešeniu a vykonaným experimentom a používateľské príručky pre systém ALEF a rozšírenie prehliadača UTrack (včítane súčasti GazerTracker). Prílohy uzatvára obsah elektronickej prílohy diplomovej práce.

2 MONITOROVANIE POUŽÍVATEĽA PRI PRÁCI S INFORMÁCIAMI NA WEBE

V oblasti sledovania a vyhodnocovania práce používateľa s obsahom sa často používajú metódy využívajúce explicitnú spätnú väzbu, čiže spätnú väzbu, ktorú vedome poskytuje používateľ napríklad ohodnotením obsahu. Tento spôsob má ale viaceré nevýhody (Claypool et al., 2001), ktoré môžu byť nielen v oblasti elektronickej výučby nežiaduce. Určitý aspekt je samotné selektívne poskytnutie spätnej väzby používateľom – rozhodnutie, či používateľ spätnú väzbu vôbec poskytne, respektíve ju bude poskytovať len k dokumentom, ktoré sa pre neho niečím odlišujú. Ak používateľ nie je dostatočne motivovaný, tak spätnú väzbu jednoducho neposkytne a tiež môže vplyvom samotného úkonu hodnotenia dôjsť k zmene vzorcov správania používateľa pri výbere a prehliadaní dokumentov (používateľ je vyrušený náhle zobrazenou otázkou a následne sa bude správať inak, než keby sa mu otázka nepoložila), čo môže následne ovplyvniť výsledky spätnej väzby.

Problém selektívneho poskytovania explicitnej spätnej väzby je známy ako distribúcia hodnotenia v tvare písmena J – v hodnotení obsahu, dokumentu, produktu, používateľa najčastejšie uvádzajú minimálne a maximálne hodnotenie (maximálne hodnotenie prevažuje), pričom hodnotenia medzi týmito hraničnými hodnotami sa využívajú veľmi málo. Pritom práve hodnotenia medzi hraničnými hodnotami by mali podľa normálneho rozloženia prevažovať. Pravdepodobným z viacerých možných vysvetlení je, že používatelia poskytnú spätnú väzbu v prípade, keď sú veľmi spokojní alebo naopak veľmi nespokojní, pričom používatelia, ktorí by hodnotili priemerne, nemajú dostatočnú motiváciu obetovať čas a úsilie na poskytnutie spätnej väzby (Hu, Zhang & Pavlou, 2009). Objektom štúdie prezentovanej v (Hu, Zhang & Pavlou, 2009) boli používateľské recenzie produktov v internetovom obchode, ktoré okrem jednoduchého hodnotenia jednou až piatimi hviezdikami obsahujú aj text a sú tak náročnejšie na vynaložené úsilie, no rovnaký efekt sa objavil aj v praxi pri hodnotení videozáznamov na internetovom portáli YouTube, kde hodnotenie pozostáva z jednoduchého kliknutia na počet hviezdíček¹. V tomto prípade mala distribúcia hodnotení tiež výrazný tvar písmena J, pričom veľmi výrazne prevažovalo maximálne pozitívne hodnotenie.

Pretože sa z hľadiska aktivity používateľov ukazuje explicitná spätná väzba aj v najjednoduchšom prípade hodnotenia celého objektu naraz pomocou jednoduchých hviezdíček ako nevhodná, nemôžeme v prípade zjemnenia hodnotenia až na jednotlivé fragmenty dokumentu, čo predstavuje zväčšenie počtu úkonov, očakávať dostatočnú aktivitu používateľov. V oblasti témy projektu (práca používateľa s dokumentom) je viac vhodná implicitná spätná väzba. Práve pri vyhodnocovaní práce s dokumentom je možné sledovať viaceré implicitné indikátory záujmu, od jednoduchých, ako napríklad počet kliknutí pomocou myši (Longo & Barrett, 2009), cez akcie ako označovanie textu, vyhľadávanie v rámci dokumentu (Longo & Barrett, 2009), až po posúvanie dokumentu

¹ Official YouTube Blog: Five stars dominate ratings [online]. 2009 [cit. 2011-05-05]. <http://youtube-global.blogspot.com/2009/09/five-stars-dominate-ratings.html>

používateľom (Kelly & Belkin, 2001), respektíve z posúvania odvodený čas čítania jednotlivých častí dokumentu (Hill et al., 1992) a pod.

Okrem čisto explicitnej spätnej väzby a čisto implicitnej spätnej väzby sa používa ich kombinácia, teda oba typy súčasne. Možné spôsoby zahŕňajú využitie explicitnej spätnej väzby na doplnenie implicitnej a naopak doplnenie explicitnej spätnej väzby implicitnou, skúmanie ich vzájomnej súvislosti (Fox et al., 2005) a využitie explicitnej spätnej väzby na overenie výsledkov odvodených z implicitnej spätnej väzby (Longo & Barrett, 2009).

2.1 Implicitná spätná väzba

V prípade implicitnej spätnej väzby a implicitných indikátorov záujmu existuje množstvo indikátorov, ktoré je možné sledovať na rôznych úrovniach. Napríklad v prípade používania myši a klávesnice môžeme toto používanie vyhodnocovať najjednoduchším spôsobom vo forme počtov kliknutí alebo stlačení kláves počas celého zobrazenia dokumentu alebo podrobnejším spôsobom podľa fragmentov dokumentu (kliknutia v rámci jednotlivých fragmentov).

Implicitné indikátory záujmu môžeme rozdeliť do dvoch základných skupín (Fox et al., 2005; Badi et al., 2006):

- indikátory súvisiace s dokumentom ako celkom,
- indikátory súvisiace s fragmentmi dokumentu.

Do prvej skupiny (dokument) môžeme zaradiť napríklad tieto indikátory:

- počet stlačených klávesov, pohybovanie myšou,
- počet kliknutí myšou, celková vzdialenosť, o ktorú bol dokument posúvaný,
- počet posúvaní dokumentu,
- celkový čas posúvania dokumentu,
- počet zmien smeru posúvania dokumentu,
- čas do prvého kliknutia,
- spôsob opustenia dokumentu,
- vytlačenie dokumentu,
- počet a veľkosť textov skopírovaných z dokumentu,
- celkový čas zobrazenia dokumentu,
- počet označení textu,
- otváranie záložiek počas práce s dokumentom,
- počet vyhľadávaní v dokumente.

Do druhej skupiny (fragmenty dokumentov) patria napríklad tieto indikátory:

- označenia myšou,
- presuny kurzora,
- pohľad,
- čas zobrazenia (pri sledovaní posúvania dokumentu),
- kopírovanie textu.

Indikátory súvisiace s posúvaním dokumentu a sledovaním zamerania pohľadu poskytujú zaujímavé možnosti, pretože poskytujú podrobnejšie informácie než napríklad agregované indikátory (počet určitých akcií) a to aj na úrovni podrobných fragmentov dokumentu. Tiež priamo súvisia s tým, ktorou časťou dokumentu sa používateľ zaoberá, kým napríklad pohyb myši môže byť vzhľadom k samotnému dokumentu mimovoľný, keď používateľ prechádza myšou na iný ovládací prvok aplikácie.

2.2 Posúvanie zobrazeného dokumentu

Myšlienka sledovania posúvania dokumentu (angl. *scrolling*) je v (Hill et al., 1992) označená ako *opotrebovanie dokumentu*, respektíve posuvníka (angl. *scrollbar*) čítaním (angl. *read wear*) a upravovaním (angl. *edit wear*). Názov a samotný princíp vychádzajú z fyzického opotrebovávaní viacstranového dokumentu počas čítania – čím dlhšie a opakovane sa u papierovej knihy venujú jej čitatelia rovnakým stranám, tým viac sú tieto strany pokrčené, väzba je na danom mieste opotrebovaná a pod. V princípe prenesenom na elektronické dokumenty uvažujeme samotné zobrazenie jednotlivých častí dokumentu a čím viac je určitá časť sumárne zobrazená, tým je význačnejšia.

Autori toto virtuálne opotrebovanie implementovali tak, že zaznamenávali čas zobrazenia každého riadku dokumentu na obrazovke prostredníctvom značkovania novozobrazených riadkov časom zobrazenia a odpočítaním tohto času v momente, keď sa tieto riadky dostali pri ďalšom posúvaní mimo obrazovku. Sledovanie jednotlivých riadkov sa podľa autorov preukázalo ako dostačujúce. Skúmajú aj myšlienku sledovania opotrebovania na úrovni jednotlivých znakov, ktorú však zamietajú z dôvodu nárastu potrebného úložného priestoru a ako kompromis navrhujú sledovanie na úrovni slov alebo tokenov. Tento výskum však prebiehal takmer pred dvomi dekadami, v súčasnej dobe by zrejme bolo prakticky aplikovateľné aj sledovanie na hlbšej úrovni.

Pre nás je zaujímavý spôsob, akým autori v (Hill et al., 1992) ošetrili možnosť, že sa používateľ prestane venovať dokumentu – zaviedli čas vypršania záujmu 3 minúty od poslednej akcie používateľa, po ktorom sa čítanie ďalej neuvažuje. Zobierané informácie o čítaní a úpravách dokumentu využili na obohatenie posuvníka dokumentu o zobrazenie opotrebovania jeho jednotlivých častí (rovnakými autormi je patentovaný posuvník rozšírený o atribúty (Wroblewski et al., 1990)).

Touto problematikou sa tiež zaoberali v (Cockburn et al., 2009), kde myšlienku opotrebovávaní dokumentu a následného zobrazenia získaných informácií v posuvníku skúmali v kontexte znovunavštívení rôznych častí dokumentu a sledovania nedávno navštívených miest. Na posuvníku vytvárali značky v prípade, že používateľ na nich zotrval viac ako dve sekundy, pričom tieto značky tak vytvárali zoznam naposledy navštívených miest. Zároveň autori poskytli klávesové skratky (ďalšia/predchádzajúca značka, priamo číslo značky) pre rýchle pohybovanie sa medzi týmito navštívenými miestami. V práci autori následne overili, že takéto obohatenie posuvníka zrýchliло prácu používateľov, ale taktiež, že v doméne dokumentov (overenie prebiehalo prostredníctvom editora dokumentov Microsoft Word a prehliadača dokumentov Adobe Acrobat), je znovunavštevovanie rôznych regiónov významné a vyskytuje sa často.

Jednoduchší prístup bol použitý v (Kelly & Belkin, 2001), kde autori ako metriku posúvania dokumentu zvolili počet kliknutí na posuvník. Okrem tohto implicitného indikátora záujmu využili celkový čas čítania dokumentu a ako vyhodnotenie relevantnosti používateľom využili akciu uloženia dokumentu (uloženie predstavovalo pozitívne posúdenie relevantnosti, neuloženie predstavovalo negatívne). Výsledkom bolo, že používatelia neposúvali dokument výrazne odlišne v prípade relevantných oproti nerelevantným dokumentom. Môže to indikovať, že metrika počtu kliknutí na posuvník nemusí byť dostačujúca. Na druhú stranu, autori poskytli používateľom aj ďalšie možnosti práce s dokumentom (tlačidlá zobrazenia nasledujúceho kľúčového slova, nasledujúcej najlepšej časti a pod.), ktoré vyhodnocovali ako interakciu s dokumentom. Podrobnosti neuvádzajú, ale v prípade, že tieto funkcie pracovali ako navigácia v dokumente a posúvali teda dokument, mohlo tým dôjsť k skresleniu vyhodnotenia samotného posúvania dokumentu uvažovaného iba z klikania na posuvník. Autori taktiež zadali používateľom komplexnejšie úlohy s časovým ohraničením, výsledkom bolo, že čas čítania dokumentu nesúvisí s relevanciou dokumentu. Naopak, v prípade jednoduchších úloh v citovanej štúdií bol výsledkom súvis času čítania a relevantnosti dokumentu. Je preto stále možné, že uvedená metrika posúvania dokumentu v jednoduchých prípadoch postačuje.

V práci (Braganza et al., 2009) vykonali autori štúdiu správania sa používateľov pri posúvaní dokumentu pri jednoduchých a dvojstĺpcových rozloženiach dokumentu. Zaoberajú sa tromi štandardnými *modelmi rozloženia dokumentu – súvislé zvislé posúvanie*, pri ktorom je obsah dokumentu zobrazený v jedinom súvislom stĺpci a používateľ ho zvislo posúva; *stránkované rozloženie*, pri ktorom je obsah rozdelený do za sebou nasledujúcich strán, ktoré môžu obsahovať aj viac stĺpcov a *vodorovné posúvanie*, pri ktorom je obsah rozdelený do stĺpcov s pevnou výškou a tie je možné posúvať vodorovne. Ďalej skúmali *mechanizmus posúvania dokumentu* – používanie klávesnice (tlačidlá posúvania po jednotlivých stranách, šípok a posúvania na začiatok a koniec obsahu), kolieska slúžiaceho na posúvanie (na myši a pod.), prostriedkov na obrazovke (posuvník, náhľady dokumentu) a potiahnutia zobrazenia dokumentu. Skúmali pritom tiež *stratégiu posúvania*, teda spôsob, akým používatelia dokument posúvajú a kedy – po riadkoch priebežne, po riadkoch po dosiahnutí určitej časti, po stranách a pod. Autori okrem využitia vlastného prehliadača vytvoreného na zobrazovanie vybraných textov a zaznamenávanie akcií používateľa použili aj sledovanie pohľadu používateľa (prostredníctvom komerčného riešenia faceLAB²).

Z hľadiska práce používateľa s dokumentom na webe je zaujímavý model rozloženia predstavujúci *súvislé zvislé posúvanie*, pretože takýto model sa používa pri webových dokumentoch a všeobecne pri materiáloch vytváraných pre zobrazovanie na počítači. Pri tomto modeli autori zistili, že 13 % používateľov používa stratégiu posúvania po stranách, 46 % používa stratégiu neustáleho posúvania a 31 % čítalo v pevne danej oblasti, čiže viac ako 75 % nečíta medzi jednotlivými posunutiami celú stranu. Medzi výrazne preferovaný a aj používaný mechanizmus posúvania patrilo pri tomto modeli rozloženia posúvacie koliesko (respektíve v štúdií použitá guľa) – pri oboch zvlášť

² Seeing Machines Inc.: faceLAB™ 5 Specifications [online]. 2009 [cit. 2011-05-05]. <http://www.seeingmachines.com/product/facelab/specifications/>

vyhodnocovaných činnostiach čítania a hľadania odpovedí na otázky používal primárne tento mechanizmus približne štvornásobne väčší počet používateľov ako jednotlivé ostatné možnosti. Ďalšími približne rovnako primárne používanými možnosťami boli poťahovanie dokumentu, posuvník na obrazovke a smerové tlačidlá (okrem tlačidiel posúvania po stránkach), pričom z týchto vynikal posuvník tým, že ho celkovo použila významná časť používateľov (aj keď to nebol ich primárny spôsob). Nasledovali tlačidlá posúvania po stranách a náhľad dokumentu, ktorý použil len jeden používateľ a celkovo ho žiadny používateľ nepoužíval ako svoj primárny spôsob posúvania.

S posúvaním dokumentu tiež pracovali v (Ishak & Feiner, 2006), kde predstavili myšlienku posúvania dokumentu, pri ktorom sa uvažuje posúvaný obsah (angl. *content-aware scrolling*, CAS). Používateľovi sprístupnili posuvník, ktorý nepohyboval dokumentom tradičným spôsobom rýchlosťou zodpovedajúcou pohybu myši, ale podľa obsahu posúvaného dokumentu sa menili smer a rýchlosť posúvania a tiež priblíženie zobrazenia. Metóda bola navrhnutá pre mobilné zariadenia s obmedzenou zobrazovacou plochou displeja, preto sa autori zameriavajú najmä na dvojdimenzionálne posúvanie obsahu, pričom jednou z aplikácií je poloautomatické postupné prehliadanie stĺpcov obsahu viacstĺpcového dokumentu pomocou jediného vertikálneho posuvníka (po posunutí na koniec jedného stĺpca sa dokument posunie aj v horizontálnom aj vo vertikálnom smere na začiatok nasledujúceho stĺpca a posúvanie súvislo pokračuje na jedinom pôvodnom vertikálnom posuvníku).

Ďalšou aplikáciou je automatický prechod všetkými tvármi na fotke pomocou len jediného posuvníka. Pri posúvaní v zvislom alebo vodorovnom smere sa teda obsah posúval automaticky aj v druhom smere podľa ciest vypočítaných automaticky z obsahu dokumentu (pre textové stĺpce) alebo ciest vytvorených používateľom (pre obrázky). Pretože v prípade dokumentov a výučbových textov vytváraných pre zobrazovanie na počítači býva obsah prispôbený šírke zobrazenia (jeden stĺpec textu, prispôbená šírka obrázkov), adaptívne menenie smeru posúvania a úrovne priblíženia dokumentu zrejme nemá v kontexte (výučbových) webových dokumentov význam. Premenná rýchlosť posúvania je však aplikovateľná aj v jednodimenzionálnom posúvaní a tvorí tak jedno z možných používateľských rozhraní pre výstup informácií odvodených z implicitnej spätnej väzby (napríklad zobrazenie málo dôležitých fragmentov dokumentu je posúvané rýchlejšie).

2.3 Zisťovanie prítomnosti používateľa

Počas zobrazovania dokumentu používateľ prirodzene nemusí venovať dokumentu neustálu pozornosť, môže sa venovať iným úlohám alebo dokonca opustiť počítač. Pretože týmto môžu byť do značnej miery ovplyvnené metódy zaznamenávajúce čas čítania, či už celých dokumentov alebo jednotlivých fragmentov obsahu dokumentu, je potrebné tento čas určitým spôsobom identifikovať a vylúčiť. Základnou možnosťou je určenie času neaktivity zo štandardne dostupných údajov – pohybovanie myši, používanie klávesnice – čiže: „ak používateľ neinteraguje so vstupnými zariadeniami, nečíta dokument“. Používateľ však zrejme počas čítania dokumentu túto interakciu (okrem momentov posúvania zobrazenia) nemusí vôbec vykonávať, je preto nutné ho

buď k tejto interakcii prinútiť alebo napríklad použiť časovač, kde: „ak používateľ určitú dobu neinteraguje so vstupnými zariadeniami, nečíta dokument“.

Jednoduchý prístup tohto typu bol použitý v práci (Kelly & Belkin, 2001), kde autori využili vlastný časovač vypršania záujmu s konštantnou dobou a napríklad stmavnutie obrazovky, ktoré implikuje že používateľ práve teraz dokument nečíta – ak by ho predsa čítal, musí v prípade stmavnutia vykonať ľubovoľnú akciu, po ktorej sa mu obsah opäť zobrazí. Autori tento postup využili ako integrálnu súčasť metódy, preto ho zvlášť nevyhodnocujú, no je zrejmé, že bude zahŕňať určitú nepresnosť. Okrem prípadu riadeného experimentu môže používateľ čítať dokument v inej záložke, počas toho používať vstupné zariadenia a obrazovka tak nestmavne, aj keď používateľ sledovaný dokument nečíta. Naopak, to, že používateľ nemanipuluje so vstupnými zariadeniami, má najmenej dva časté významy: používateľ číta dokument alebo používateľ sa dokumentu nevenuje (odišiel, atď.), pričom tieto dva významy sú presne opačné stavy, ktoré je pre spresnenie potrebné odlišiť.

Možnosťou je uvažovanie len riadeného experimentu/použitia, pri ktorom sa možnosť napríklad odchodu používateľa vylúči alebo naopak používanie sledovateľných vstupných ovládacích zariadení počas čítania vynútiť. Podľa konkrétnej metódy implementácie potom toto vynútenie poskytuje zároveň aj určitú formu sledovania zamerania pohľadu používateľa (Ullrich & Melis, 2002; Tarasewich & Fillion, 2004), preto sa k týmto metódam ešte vrátíme ďalej v časti venovanej sledovaniu pohľadu.

Sledovanie štandardných používateľských vstupných zariadení preto nepostačuje a je potrebné využiť ďalší zdroj informácií. Tento problém je možné vyriešiť počítačovým videním, čiže využitím kamier na fyzické sledovanie používateľa nezávislé od jeho aktivít. Pri využití kamier existuje viacero prístupov – základným postupom je sledovanie hlavy, respektíve tváre používateľa, na detailnejšej úrovni potom sledovanie pohľadu používateľa, ktorého zahrnutie je cieľom tohto projektu.

Sledovanie hlavy a tváre používateľa môžeme rozdeliť do viacerých úrovní podľa využívaných hardvérových prostriedkov. Z tohto hľadiska existujú komplexné systémy často zároveň sledujúce pohľad používateľa, ide však o komerčné riešenia vyžadujúce špecializovaný hardvér (faceLAB), nákladné profesionálne riešenia (EyeLink³) alebo vybudované riešenia vyžadujúce tiež relatívne nákladný hardvér (DV kamera (D. Hansen & J. Hansen, 2006)) alebo špecifickú konštrukciu (upevnenie dvoch kamier na hlavu používateľa (Babcock & Pelz, 2004)). Naším cieľom je však využitie takéhoto sledovania pre účely identifikácie fragmentov textov u mnohých používateľov a teda z hľadiska využiteľnosti u bežného používateľa bez špecializovaných hardvérových riešení je vhodné sa zamerať na počítačové videnie využívajúce obyčajné, bežne dostupné a rozšírené webové kamery.

Príkladom riešenia využívajúceho lacnú webovú kameru na zistenie prítomnosti používateľa je (Fosso & Porta, 2009). Autori konkrétne rozoznávajú 7 stavov používateľa: používateľ otočený k obrazovke, hlava otočená doľava, hlava otočená

³ SR Research Ltd.: EyeLink 1000 [online]. 20 10 [cit. 2010-05-05]. http://www.sr-research.com/EL_1000.html

doprava, používateľ otočený chrbtom k obrazovke, rozprávajúci telefonujúci používateľ, nerozprávajúci telefonujúci používateľ a používateľ neprítomný. Využili pri tom postup postupného nájdenia používateľa prostredníctvom detekcie pohybu a následnej detekcie farby pokožky prostredníctvom farebnej segmentácie. Následne vykonávajú detekciu tváre, detekciu očí a napokon detekciu rúk a pier (pre identifikáciu telefonovania a rozprávania pri telefonovaní). Autori sa tiež zamerali na kalibráciu, ktorá je pri metódach počítačového videnia nevyhnutná a poskytli automatickú kalibráciu s následnou poloautomatickou dodatočnou kalibráciou tak, že používateľovi zobrazili ovládače parametrov pre úpravu automaticky zistených hodnôt.

Iný spôsob riešenia predstavuje samostatná detekcia tváre v obraze pomocou detektorov vopred naučených objektov (v tomto prípade tvári) obsiahnutých v knižnici *OpenCV* (Bradski & Kaehler, 2008). Tento prístup sme už využili v projekte (Labaj et al., 2009), kde bola detekcia tváre vo forme obdĺžnika opísaného tvári v obraze webovej kamery použitá na určenie polohy používateľa pred displejom a táto poloha bola následne využitá na zmenu zobrazovanej perspektívy pre adaptáciu zobrazenej scény podľa uhľa pohľadu používateľa. Pre účely detekcie prítomnosti používateľa je možné využiť zhodný postup, pričom poloha tváre v obraze sa však zredukuje na dva stavy: používateľ je prítomný a používateľ je neprítomný. Takáto detekcia nemusí zachytiť stav, keď sa používateľ nachádza pred displejom a jeho tvár teda detegujeme v obraze kamery, ale dokumentu sa nevenuje a napríklad pracuje s objektmi mimo počítača.

2.4 Spôsoby sledovanie zamerania pohľadu používateľa

Zvláštnu úlohu predstavuje zameranie pohľadu používateľa. Nielenže ho je možné použiť ako indikátor prítomnosti používateľa na spresnenie času čítania namiesto použitia detekcie tváre, ale zároveň sám o sebe predstavuje implicitný indikátor záujmu, pričom podľa úrovne presnosti je možné pomerne presne identifikovať jednotlivé fragmenty obsahu, na ktoré sa používateľ pozerá.

Sledovanie zamerania pohľadu používateľa však nemusí prebiehať prostredníctvom počítačového videnia. Sledovanie pohľadu používateľa je možné do určitej miery odvodiť z pohybu myši. V štúdiu (Chen et al., 2001) autori skúmali súvislosť medzi pozíciou kurzora myši a pohľadom používateľa počas prehliadania webových stránok, pričom využili profesionálne riešenie EyeLink na sledovanie pohľadu používateľa. Autori rozlíšili štyri druhy náhlych pohybov myši (angl. *mouse saccade*) podľa počiatočného a koncového miesta – zotrvanie v nevýznamnej oblasti, presun do nevýznamnej oblasti, zotrvanie v rovnakej významnej oblasti, presun do novej významnej oblasti. Z výskumu vyplynulo, že 84 % oblastí navštívených kurzorom myši bolo tiež navštívených pohľadom a 70 % prípadov pohybov do novej významnej oblasti bolo nasledovaných pohybom oka. Priemerná vzdialenosť medzi pohľadom a kurzorom myši bola v prípade pohybov končiacich vo významných oblastiach 90 obrazových bodov, pričom v 40 % prípadov z nich bola vzdialenosť do 35 bodov.

Pohľad používateľa teda do istej miery nasleduje kurzor myši, respektíve naopak, používateľ presúva myš na miesta, na ktoré sa pozerá. Nevýhodou tohto prístupu je, že

ho nie je možné použiť na detekciu prítomnosti používateľa – v prípade, že používateľ náhle pohybuje myšou, umožňuje odvodiť, že na koncové alebo začiatkové miesto pohybu myši bude s určitou pravdepodobnosťou podľa typov koncových oblastí mieriť pohľad používateľa, keď však používateľ myšou nehýbe, o pohľade používateľa nič nevyvedá. Niektorí používatelia však počas čítania sledujú kurzorom myši riadok ako pomôcku pre vedenie pohľadu (Atterer et al., 2006). Pohybovanie myšou počas čítania je takto možné vynútiť pomocou zobrazenia iba tých častí obsahu, na ktoré používateľ premiestni kurzor (Tarasewich & Fillion, 2004; Ullrich & Melis, 2002).

V práci (Ullrich & Melis, 2002) autori pôvodne upravujú text na stránke pomocou prekrytia, zmeny jeho farby na farbu blízku farbe pozadia alebo významným zmenšením veľkosti písma tak, aby bol text nečitateľný. Používateľ musí na prečítanie oblasti presunúť na dané miesto kurzor, kedy sa mu daná oblasť zobrazí normálne. Zmeny formátovania sú implementované pomocou jazyka JavaScript pre použitie vo webovom prehliadači. Systém *Enhanced Restricted Focus Viewer* (Tarasewich & Fillion, 2004) využíva rovnaký prístup, ale namiesto manipulácie s formátovaním textu na stránke rozostří celú stránku a používateľ tak môže prehliadať iba aktuálnu oblasť okolo kurzora. Systém pracuje iba s obrázkami vo formáte bitových máp, čo výrazne obmedzuje ako prácu používateľa so stránkou, tak aj prípadné vyhodnocovanie dynamických stránok, ide však o nedostatok implementácie, uvedenú metódu rozostrovania by zrejme bolo možné implementovať aj nad riadnym webovým prehliadačom. Obe tieto metódy rozostrovania a zmeny formátovania však môžu mať vplyv na správanie sa používateľa, pretože používateľ nemôže napríklad rýchlo prejsť stránku pohľadom pre počiatočné oboznámenie sa s ňou a nemôže rýchlo opätovne vyhľadať už prečítanú informáciu/časť, ale musí ju nájsť „v tme“. Z oboch štúdií nie je jasná miera takéhoto dopadu na správanie používateľa.

Aby teda bolo možné sledovať pohľad používateľa spôsobom, ktorý umožní jeho detekciu aj počas doby, keď používateľ nehýbe myšou a zároveň nebolo potrebné si tento pohyb vynucovať, čo by zrejme zasiahlo do vyhodnocovaného správania, vhodnejšie je zamerať sa na počítačové videnie a fyzicky sledovať oči používateľa.

V oblasti sledovania pohľadu fyzického používateľa existujú nákladné komerčné/profesionálne riešenia (faceLAB, EyeLink), ktoré boli s úspechom použité pri výskume, často v súvislosti s vyhodnotením iných implicitných indikátorov, ako napríklad v (Braganza et al., 2009; Chen et al., 2001). Z dôvodov opísaných v predchádzajúcej časti je však lepšie sa zamerať na využitie webovej kamery, prípadne inej lacnej kamery, ktorou ľahko disponuje používateľ v neriadenom prostredí.

2.5 Sledovanie zamerania pohľadu používateľa pomocou webovej kamery

V oblasti využitia jednoduchej webovej alebo podobnej kamery na sledovanie pohľadu používateľa existujú viaceré prístupy, ktoré sa odvíjajú od použitého hardvéru. Častým je použitie infračerveného nasvietenia používateľovej tváre alebo iba oka pri použití kamier s odstráneným infračerveným filtrom alebo zapnutým nočným režimom, kým iné riešenia používajú iba obraz bežnej webovej kamery vo viditeľnom spektre:

Jednotlivé prístupy sa tiež líšia potrebnou veľkosťou oka v obraze – takže obraz kamery je v niektorých prípadoch potrebné priblížiť k tvári používateľa – buď opticky alebo priblížením samotnej kamery.

Pokročilejšiu detekciu zamerania pohľadu od základov extenzívne opísali autori v (D. Hansen & Pece, 2005) – autori namiesto vyhľadávania známych tvarov očí, ktoré využívajú nasledujúce riešenia, využívajú aktívne sledovanie obrysov. Podľa autorov je táto metóda odolná voči zmenám svetelných podmienok, rozostreniam kamery, ale i prechodmi medzi infračerveným a neinfračerveným nasvietením. Opisovaná metóda je však značne rozsiahla a tak v tomto projekte, u ktorého nie je jedinou náplňou samotné sledovanie pohľadu používateľa, je lepšie sa zamerať na jednoduchšie alebo už dostupné riešenia.

Okrem výskumov rôznych metód sú viaceré riešenia k dispozícii vo forme dostupných aplikácií alebo knižníc, z ktorých niektoré boli overené použitím vo výskumných projektoch. Analyzovali sme päť takto dostupných riešení: FTR_BY_WLP⁴, ITU Gaze Tracker⁵, openEyes⁶, myEye⁷ a OpenGazer⁸.

2.5.1 FTR_BY_WLP

Jedným z riešení používajúcich infračervené prsvietenie je systém *FTR_BY_WLP*. Autor zverejnil ukážku systému implementovaného s využitím knižnice OpenCV. Toto riešenie vyžaduje infračervené prsvietenie na detekciu odrazu rohovky, pričom centrá zorníc sú určené pomocou sledovania okrajovej časti rohovky oka nazývanej limbus. Riešenie je však uzavreté a autor k nemu nezverejnil podrobnejšie informácie a detaily použitých metód. Výhodou tohto systému je automatická detekcia tváre a očí, následne je potrebná iba manuálna kalibrácia umiestnenia pohľadu, ktorá spočíva v postupnom zobrazovaní bodov na obrazovke, na ktoré používateľ zameriava pohľad.

2.5.2 ITU Gaze Tracker

Ďalším riešením využívajúcim infračervené svetlo je systém *ITU Gaze Tracker*. Tento systém predstavuje otvorené riešenie so zverejnenými zdrojovými kódmi. Autori proces sledovania pohľadu používateľa rozdelili do dvoch podprocesov:

⁴ PIECHULLA, W.L.: FTR_BY_WLP [online]. 2008 [cit. 2011-05-05].

http://www.walterpiechulla.de/fttr_by_wlp_08/doc/

⁵ HANSEN, D., et. al.: ITU GazeTracker [online]. 2010 [cit. 2010-05-12].

<http://www.gazegroup.org/research>

⁶ LI, D., et. al.: openEyes [online]. 2011 [cit. 2011-05-05].

<http://thirtysixthspan.com/openEyes/>

⁷ LAGINESTRA, M.: myEye Project [online]. 2009 [cit. 2011-05-05].

<http://myeye.jimdo.com/>

⁸ ZIELINSKI, P.: Opengazer: open-source gaze tracker for ordinary webcams (software), Samsung and The Gatsby Charitable Foundation [online]. 2009 [cit. 2011-05-05]. <http://www.inference.phy.cam.ac.uk/opengazer/>

- sledovanie samotných očí – tento proces zahŕňa detekciu a následne sledovanie pohybu očí na základe tvarov oka,
- odhadnutie pohľadu – výpočet zamerania pohľadu z tvarov už nájdeného oka.

Špecifikom tohto systému je, že vyžaduje priblížený obraz oka, je teda potrebné využitie kamery s funkciou priblíženia (zoom) (v (D. Hansen & J. Hansen, 2006) využili digitálnu kameru umiestnenú v blízkosti displeja s obrazom priblíženým na oko používateľa, pričom cieľom výskumu bolo sledovanie pohľadu pacientov s poruchami motoriky) alebo umiestnenie bežnej kamery do blízkosti oka buď pomocou pripevnenia k hlave používateľa, alebo pomocou umiestnenia na stojan. Oba spôsoby fyzického priblíženia kamery k oku overili autori systému v aplikácii systému na písanie pomocou pohybu očí v (San Agustin et al., 2010; San Agustin et al., 2009) – kameru buď umiestnili na stôl pomocou stojanu vyrobeného zo stolnej lampy alebo na malú drevenú dosku, do ktorej používateľ zahryzol a kameru tak použitím zubov držal v blízkosti tváre. V prípade použitia u pacienta s poruchou motoriky a nemožnosťou žmurkať autori zaznamenali vysušanie očí LED diódami slúžiacimi na infračervené prislvetenie oka.

Tento systém opäť poskytuje čiastočne automatickú kalibráciu, keď automaticky nájde odraz infračerveného svetla od zreničiek alebo odraz od oka, môže však byť potrebné upraviť parametre veľkosti zreničiek a pod. Následne je potrebná manuálna kalibrácia umiestnenia pohľadu opäť prostredníctvom sledovania zobrazovaných bodov.

2.5.3 myEye

Projekt *myEye* tiež využíva infračervené svetlo a bol taktiež vyvinutý na pomoc pacientom s motorickými problémami. Kalibrácia je jednoduchšia v zmysle nižšieho počtu bodov, na ktoré sa používateľ musí pozrieť, než u väčšiny ostatných riešení (iba štyri rohy obrazovky), no kalibrácia nie je automatizovaná a pri pohľade do rohu obrazovky je potrebné stlačiť kláves s číslom príslušného rohu. Výstup údajov o získanom pohľade je prispôsobený použitiu so špecializovanými programami pre písanie alebo syntézu reči, ktoré využívajú ako vstup aktuálnu pozíciu kurzora myši. Program tak iba pohybuje kurzorom myši, čo nie je nevhodné pri iných použitíach, pri ktorých je potrebné, aby používateľ pomocou myši nezávisle ovládal systém.

2.5.4 openEyes

Projekt *openEyes* sa skladá z viacerých podprojektov, ktoré umožňujú sledovanie pohľadu z obrazu kamery získavaného vo viditeľnom aj v infračervenom spektre. Projekt je skôr zacielený na použitie v pohybe, mimo displeja počítača, keď sa používateľ pohybuje s kamerou pripevnenou na hlave a softvér podporuje aj vstup obrazu druhej kamery, ktorá naopak sníma scénu, na ktorú sa používateľ pozerá. Autori poskytli podrobné príručky pre zostrojenie potrebného hardvéru v dvoch variantoch: jednoduchší variant spočívajúci v nahraní obrazu z kamery na SD kartu, odkiaľ je následne na počítači analyzovaný a zložitejší variant vyžadujúci hardvérové úpravy, ktorý umožňuje sledovanie pohľadu v reálnom čase.

2.5.5 OpenGazer

System *OpenGazer* predstavuje otvorený systém so zverejnenými zdrojovými kódmi umožňujúci sledovanie pohľadu pomocou bežnej webovej kamery bez použitia infračervených diód. Vzhľadom na absenciu pomocných infračervených odrazov je tak v prípade tohto systému potrebná viac manuálna kalibrácia pozostávajúca z určenia bodov v rohoch očí a určenia ďalších bodov na tvári pomocou myši. Následne sú tieto body sledované v obraze webovej kamery a ďalším krokom je kalibrácia umiestnenia pohľadu rovnaká ako v ostatných opísaných systémoch. Počas kalibrácie pohľadu autori používajú extrahované obrázky očí (na extrakciu používajú používateľom zvolené body umiestnené do rohov očí) na učenie gaussovského procesu, ktorým reprezentujú mapovanie medzi obrázkami očí a pozíciou na obrazovke. Následne sa pomocou naučeného gaussovského procesu z obrázkov očí počíta poloha pohľadu na obrazovke. Autori uvádzajú potrebu prísne statického držania hlavy počas celého procesu kalibrácie.

Tento systém bol s modifikáciami úspešne použitý v (Xu et al., 2008; Xu et al., 2009). Obe uvedené práce vytvoril jeden tím autorov, pričom riešenie na sledovanie pohľadu vytvorili kombináciou lacnej webovej kamery bez funkcií infračerveného podsvietenia a nočného videnia. V práci (Xu et al., 2008) následne sledovanie pohľadu využili pri online odporúčaní dokumentov, obrázkov a videozáznamov, v prípade práce (Xu et al., 2009) bolo sledovanie pohľadu využité pri sumarizácii dokumentov.

Obe práce sú z pohľadu témy tohto projektu zaujímavé. Autori v oboch predstavujú koncept času záujmu, v ktorom priradujú jednotlivým slovám, obrázkom a pod. vzorky pohľadu na základe gaussovského rozloženia, pri ktorom uvažujú obalovací box daných prvkov. Predpovedaný čas záujmu pre celý dokument je potom priemer časov záujmov slov, ktoré sa v dokumente nachádzajú a ich čas záujmu je už známy z minulosti. V prípade predpovedania času záujmu pre slová využívajú už známe časy záujmu pre slová, ktoré sú najviac sémanticky podobné s daným slovom. Čas záujmu pre vety následne predpokladajú ako súčet časov záujmov jednotlivých slov, pričom slová vážujú hodnotou 0, ak ide o tzv. „stop word“ (časté slovo), hodnotou 0,6, ak bol čas záujmu daného slova predpovedaný a hodnotou 1, ak bol jeho čas záujmu získaný pohľadom používateľa. Sumarizáciu dokumentu vykonávali výberom potrebného počtu viet s najvyšším časom záujmu, pričom neznáme časy záujmu, keď používateľ len začína čítať, dopĺňajú konvenčným sumarizérom MEAD.

2.5.6 Zhrnutie

Vlastnosti jednotlivých preskúmaných riešení sú zhrnuté v tab. 1. Okrem už uvedených rozdielov v potrebnom type kamery a jej umiestnení/použití, sú dôležité rozdiely v licencií, pričom riešenia FTR_BY_WLP a myEye nezverejnili zdrojový kód a nie je využiteľný iný spôsob prístupu k pohľadu používateľa.

Tabuľka 1: Prehľad dostupných projektov na sledovanie pohľad používateľa.

Projekt	Platforma	Licencia	Hardvér	Snímanie obrazu	Prístup k údajom
FTR_BY_WLP	neznáme	CC BY-NC 2.0, bez zdrojového kódu	web. kamera s LED	<i>vzdialené:</i> pod displejom	nie
ITU Gaze Tracker	Emgu CV; Windows XP SP2	GPLv3 a komerčná	DV kamera s nočným videním, IČ web. kamera	<i>vzdialené:</i> priblížené na oko <i>blízke:</i> uchytenie na hlave/stojan	UDP/TCP API, knižnica, zdrojový kód
myEye	OpenCV; Windows XP SP2	CC BY-NC-ND 2.5, bez zdrojového kódu	IČ web. kamera	<i>blízke:</i> uchytenie na hlave/stojan	pohyb kurzora myši
openEyes	Matlab/OpenCV; multiplatformné	GPLv2	web. kamera/ IČ web. kamera	<i>blízke:</i> uchytenie na hlave	zdrojový kód
OpenGazer (NetGazer)	OpenCV; Linux, MacOS (Windows)	GPLv2 (GPL)	web. kamera	<i>vzdialené:</i> pod displejom	knižnica, zdrojový kód

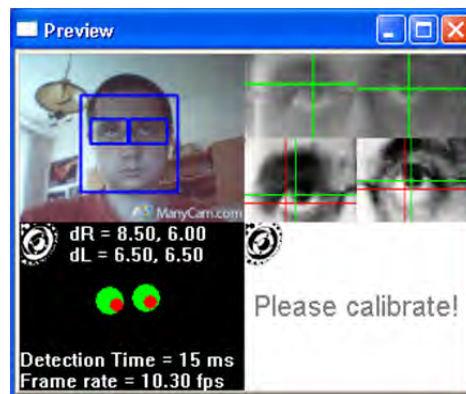
2.6 Testovanie existujúcich riešení sledovania pohľadu používateľa pomocou webovej kamery

Vzhľadom na komplikovanosť samotnej detekcie pohľadu používateľa vytvárajú sa od základov a naproti tomu možnosť použitia dostupnej knižnice OpenCV, ale i hotových riešení, ktoré túto knižnicu využívajú a poskytujú služby sledovania pohľadu, je vhodné v riešení použiť tieto existujúce projekty. Z tohto dôvodu sme vo fáze hľadania vhodnej technológie prevzali a otestovali niektoré nami analyzované riešenia. Testovanie prebehlo pomocou bežnej jednoduchej webovej kamery, akú by mal k dispozícii typický používateľ (rozlíšenie 640x480, 30 snímkov/s, bez infračerveného prísветenia) a kamera bola umiestnená vo vzdialenosti.

2.6.1 FTR_BY_WLP

Tento softvér pracuje s bežnou webovou kamerou, vyžaduje však infračervené podsvietenie. Ukážková aplikácia okamžite po zapnutí obrazu z kamery našla tvár a oči (obr. 1), avšak táto detekcia bola veľmi závislá od natočenia hlavy v osi natočenia nahor/nadol a už v prípade relatívne malých uhlov strácala aplikácia sledovanú tvár. Oproti ostatným pohybom bola detekcia dostatočne odolná. Pretože naša kamera nedisponovala infračerveným podsvietením, ktorého odraz je v tomto riešení

vyžadovaný na detekciu niektorých vlastností oka, kalibrácia zamerania pohľadu neprebehla úspešne. Pohľad detegovaný na obrazovke počas kalibrácie síce vzdialene nasledoval skutočný pohľad, testovací náhľad však ukazoval na veľké odchýlky.



Obrázok 1: Ukážka detekcie tváre a očí softvérom FTR_BY_WLP.

2.6.2 GazeTracker

Tento softvér fungoval v daných ohraničeniach testovania veľmi obmedzene, zrejme to vyplýva z potreby kamery priblíženej k oku, pričom bežne býva webkamera umiestnená v blízkosti displeja. Porovnanie spracovaného výstupu pri použití takejto nevhodnej kamery (navyše bez infračerveného podsvietenia) a pri použití kamery podľa pokynov sa nachádza na obr. 2. Zelené oblasti predstavujú odraz, ktorý by mal vychádzať iba zo zreničiek (odraz infračerveného svetla), modré oblasti predstavujú „odraz“, ktorý by mal umožniť sledovanie pohybu oka podľa zmeny odrazu (na pravom príklade nie je táto funkcia zapnutá), no je jasné, že v prípade bežnej, nepriblíženej kamery bez infračerveného svetla nefunguje správne ani jeden z týchto dvoch režimov detekcie.



Obrázok 2: Porovnanie riešenia GazeTracker pri použití bežnej webovej kamery zo štandardnej pozície (vľavo) a blízko umiestnenej kamery s infračerveným prísiviením (vpravo – prevzaté zo stránok riešenia⁹).

⁹ <http://www.gazegroup.org/home/6-news/26-itu-gaze-tracker-released>

2.6.3 OpenGazer (NetGazer)

Projekt OpenGazer sa nám neporadilo bez potreby úprav úspešne spustiť, nakoľko je určený pre operačný systém Linux a je závislý od viacerých externých knižníc. Tento softvér sa však vyskytuje aj v podobe nezávislej distribúcie portovanej na operačný systém Windows a platformu .NET (NetGazer¹⁰). Táto verzia síce neumožňuje výber kamery ako zdroja signálu a je zjednodušená len na základné predvedenie, poskytovala však ako jediná z ostatných riešení funkčné sledovanie pohľadu. Na obr. 3 je uvedená ukážka, pričom používateľ sa pozerá na väčší (fialový) štvorec a menší (modrý) štvorec predstavuje pohľad vypočítaný sledovaním pohľadu, v závislosti od rôznych faktorov (pohyb hlavou, pozícia na obrazovke a pod.) je však presnosť často výrazne nižšia. Obrázok predstavuje iba výrez z obrazovky celého programu, pričom bližšie ku kraju sa presnosť mierne znižuje. Toto riešenie funguje aj s okuliarmi.

Samotnému sledovaniu pohľadu predchádza dvojité kalibrácia – v prvom kroku používateľ určí rohy oboch očí (na obr. 3 červené body na vonkajších okrajoch očí) a následne aspoň dva body (odporúčaných je päť bodov a viac) pre sledovanie pohybu celej hlavy podľa pohybu tváre. Následne prebehne kalibrácia, pri ktorej sa postupne zobrazuje štvorec na rôznych miestach obrazovky (rohy, stredy strán, okolo obrazu z kamery v strede), na ktorý používateľ zakaždým upriami pohľad a program túto pozíciu spracuje. Nevýhodou tohto systému je citlivosť na pohyby hlavy – ako pri kalibrácii, tak aj pri následnom používaní, kedy aj menšia zmena polohy hlavy spôsobí vznik veľkých odchýlok vo vypočítanom pohľade.



Obrázok 3: Ukážka sledovania pohľadu riešením NetGazer (OpenGazer), menší (modrý) štvorec v pravej časti predstavuje pohľad detegovaný softvérom, väčší (fialový) štvorec predstavuje miesto, na ktoré sa používateľ pozerá. (Obrázok je iba výrezom zo sledovanej plochy.)

¹⁰ NetGazer [online]. 2009 [cit. 2010-05-11]. <http://sourceforge.net/projects/netgazer/>

2.6.4 Zhodnotenie sledovania pohľadu používateľa

Riešenie WTR_BY_WLP nie je podľa našich zistení pre tento projekt použiteľné, aplikácii zjavne chýbalo infračervené podsvietenie. V prípade riešenia GazeTracker bol hlavný nedostatok v absencii priblíženia obrazu kamery. Podľa niektorých tvrdení autorov používa aplikácia infračervené svetlo iba ako podporné, ale na druhú stranu autori ani neuvádzajú, že by bolo infračervené svetlo iba voliteľné a nebolo by nevyhnutné, takže k príčine neuspokojivého fungovania mohlo prispieť aj ohraničenie testu na webovú kameru bez infračerveného svetla. Riešenie NetGazer predstavujúce port riešenia OpenGazer z operačného systému Linux na operačný systém Windows sa osvedčilo aj pri mierne rozostrenom obraze kamery (otestované ručným zaostrovaním na webovej kamere) a pri použití okuliarov (avšak s antireflexnou úpravou). Vypočítaný bod pohľadu sa pohyboval vo vzdialenosti približne 100 obrazových bodov od skutočného zamerania pohľadu, pričom presnosť v blízkosti stredu displeja bola lepšia a ku krajom sa mierne zhoršovala. Toto vypočítané zameranie pohľadu bolo mierne nestabilné a pri statickom pohľade oscillovalo približne do vzdialenosti 10 bodov okolo jedného miesta – podľa predchádzajúcich skúseností s počítačovým videním je to však bežný jav vyskytujúci sa pri používaní obrazu z webových kamier.

Nevýhodou portovaného systému NetGazer je však jeho ranná fáza – vyskytuje sa iba v jedinej verzii 0.1 a táto verzia je dostupná už od januára 2009. Keďže toto riešenie je však založené na knižnici OpenGazer, ktorú iba obaluje, primárne jeho aktuálnosť závisí od knižnice OpenGazer. Ako riešenie na sledovanie pohľadu tak vychádza systém NetGazer najvýhodnejšie.

Počas testovania všetkých riešení sa pri použití staršej webovej kamery vyskytol problém s prístupom k obrazu kamery – programy ponúkali nesprávne hodnoty nastavení rozlíšenia a snímkovacej frekvencie kamery a po vybraní týchto hodnôt prestávali pracovať. Tento problém sme vyriešili použitím softvéru ManyCam, ktorý zaviedol v systéme virtuálnu kameru, na ktorú smeroval obraz zo skutočnej a s touto virtuálnou kamerou pracovali jednotlivé riešenia bez problémov. Tento problém bol však ohraničený na konkrétny hardvér, pri použití webovej kamery poskytujúcej štandard UVC (USB video class) sa tento problém neprejavil. Podpora tohto štandardu je súčasných kamier už bežnou funkciou.

2.7 Spôsobu zberu implicitnej spätnej väzby

Zbieranie implicitnej spätnej väzby je najmä v prostredí webu väčšinou náročnejšie než získavanie explicitnej spätnej väzby, pri ktorej často postačuje používateľovi položiť otázku, predostrieť nástroj na hodnotenie (hviezdičky, označenie „páči/nepáči sa mi to“) prostredníctvom štandardných postupov na webe. Rôzne druhy implicitnej spätnej väzby sú špecifické v tom, že na ich sledovanie je potrebná rôzna úroveň „oprávnení“, respektíve implementácie: od jednoduchých kódov v skriptovacích jazykoch ako napríklad JavaScript, ktoré zachytia používanie myši a klávesnice iba v rámci aktuálnej stránky, cez riešenia pomocou proxy serverov, ktoré umožnia sledovať viaceré stránky naraz pomocou vloženia kódu do každej z nich, až po zásuvné moduly do prehliadača, ktoré v rámci miernejších obmedzení zásuvných modulov už umožnia aj pozorovať

používanie prehliadača (prepínanie záložiek a pod.) a samostatné aplikácie, napríklad vlastné prehliadače, pri ktorých sú možnosti takmer neobmedzené.

Tento problém je možné názorne ilustrovať na príklade v (Atterer et al., 2006). Autori použili na sledovanie používateľa práve kód v jazyku JavaScript, ktorý z dôvodu bezpečnostných a ďalších implikovaných obmedzení umožňuje len obmedzené zachytávanie aktivít používateľa a prirodzene napríklad neumožňuje pozorovať dialógové okná prehliadača ako napríklad okno vyhľadávania v stránke. Ak teda používateľ použil klávesovú skratku Ctrl+F pre zobrazenie tohto okna, napísal na klávesnici hľadaný výraz a nechal prebehnúť vyhľadávanie, autori uvádzajú, že v záznamoch ich riešenia sa objaví stlačenie klávesu F (kláves Ctrl ako modifikátor zrejme nebol zachytený) a následné náhle posunutie stránky na nájdený výraz (zadaný výraz nemohol byť zaznamenaný). Odlíšenie od používateľov, ktorí vyhľadávanie zrušili, bolo možné podľa toho, že títo používatelia vygenerovali viacero akcií posunutia oproti jedinému posunutiu spôsobenému automatickým posunutím po nájdení výrazu.

Je teda zrejmé, že od zvoleného prístupu zachytávania spätnej väzby sa odvíjajú možnosti vytvoreného riešenia. V rôznych prácach je pritom možné pozorovať rôzne prístupy a úrovne tohto zachytávania.

2.7.1 Priama (webová) implementácia

Tento spôsob zahŕňa implementáciu potrebných funkcií pre zber spätnej väzby priamo do webovej aplikácie, napríklad zabezpečením vkladania kódu v jazyku JavaScript alebo Flash, prípadne appletov Java do jednotlivých stránok. Toto jednoduché riešenie nie je potrebné samostatne uvažovať, nakoľko je ho možné jednoducho rozšíriť na použitie proxy servera (pozri ďalšiu časť), kde funguje rovnako, ale poskytne aplikáciu uvažovaných metód do ľubovoľných stránok.

2.7.2 Proxy server

Tento spôsob zberu spätnej väzby je podobný priamej implementácii, ale rozdiel je v tom, že daný kód do stránok vkladá proxy server, cez ktorý klient pristupuje k rôznym stránkam. Záznamy o aktivitách používateľa sú potom uchovávané na tomto proxy serveri. Tento prístup bol využitý na vyhodnocovanie správania sa používateľov na webových stránkach v (Atterer et al., 2006; Velayathan & Yamada, 2006).

Výhodou tejto možnosti je z hľadiska tohto projektu existujúca implementácia adaptívneho proxy servera na Fakulte informatiky a informačných technológií (Barla & Bieliková, 2009)¹¹, ktorá už zabezpečuje zber niektorých implicitných indikátorov a zároveň umožňuje pridávanie ďalších zásuvných modulov.

¹¹ Adaptive Proxy β [online]. 2010 [cit. 2011-05-05]. <http://peweproxy.fiit.stuba.sk/>

2.7.3 Zásuvný modul do prehliadača

V tomto spôsobe zberu spätnej väzby je vytvorený zásuvný modul do webového prehliadača. Oproti predchádzajúcemu spôsobu tak dochádza k rozšíreniu sledovateľných aktivít, napríklad o sledovanie samotného prehliadača (dôležité je prepínanie záložiek, potom používanie tlačidiel pohybu smerom dopredu a dozadu v histórii navštívených stránok a pod.), ktoré nie je možné vykonávať „zvnútra“ stránky, ktorá je ohraničená navigáciou medzi stránkami (prechod na predchádzajúcu stránku zruší vykonávanie kódu a pod.), respektíve k iným stránkam v ostatných záložkách nemá prístup.

Taktiež sa vynárajú bohatšie možnosti získavania aj explicitnej spätnej väzby – používateľovi môžeme vytvoriť komplexnejšie používateľské rozhranie a toto zobrazenie môžeme opäť naviazať na aktivity nezachytiteľné z pohľadu stránky a kódu v nej (napríklad zobrazenie okna pre ohodnotenie stránky pri opúšťaní stránky zatváraním záložky a pod.). Obe tieto výhody boli použité v (Fox et al., 2005), kde autori vytvorili prídavný modul pre prehliadač Internet Explorer a následne zbierali implicitné indikátory súvisiace najmä s dokumentom (celkový čas posúvania, celkový čas čítania, vytlačenie dokumentu, spôsob zatvorenia, čas do prvého kliknutia, atď.) a tiež spomínanú explicitnú spätnú väzbu.

2.7.4 Samostatná aplikácia

Ďalšiu možnú úroveň zberu spätnej väzby predstavuje vytvorenie samostatnej aplikácie vo forme prehliadača materiálov. Túto možnosť aplikovali v (Braganza et al., 2009), kde vytvorili prehliadač podporujúci podmnožinu technológií HTML/CSS. Výhoda tohto prístupu je, že tvorca tak má absolútnu kontrolu nad všetkými možnými druhmi zachytiteľnej implicitnej spätnej väzby, čiže aj vrátane akcií, ktoré by napríklad v prehliadači nemohol odchytiť z dôvodu obmedzení zásuvných modulov. Nevýhodou tohto prístupu je jeho komplikovanosť pre použitie v súvislosti s webom – v citovanej práci využili autori tento vlastný prehliadač na prezentovanie krátkych príbehov pre účely skúmania rôznych stratégií posúvania dokumentu. Na takéto použitie postačovala aj uvedená jednoduchšia implementácia podmnožiny webových technológií, no pre umožnenie prehliadania otvoreného webu 2.0, kde sú extenzívne využívané pokročilé možnosti webových štandardov, by bolo potrebné implementovať plnohodnotný webový prehliadač.

V rámci samostatnej aplikácie existuje ešte jedna možnosť – vytvorenie ekvivalentu „zásuvného modulu“ „do operačného systému“, čiže vytvorenie samotnej aplikácie, ktorá umožní sledovanie ľubovoľných aplikácií, no tento prístup v súvislosti s implicitnou spätnou väzbou zahŕňajúcou posúvanie dokumentov a pod. implikuje mnohé komplikácie vychádzajúce z veľkých odlišností rôznych aplikácií a prístupu „do vnútra“ týchto aplikácií k reprezentácii dokumentov, používateľského rozhrania a pod. Naopak je takýto prístup zrejme výhodný pri sledovaní pohľadu používateľa, nakoľko pohľad používateľa musí v súčasnej dobe bežať natívne – technológii ako Flash, ktoré majú tiež prístup k webovej kamere, nemajú dostatočné zázemie knižníc potrebných algoritmov alebo dostatočný výkon pre výpočet pohľadu.

Jednoduchší variant predstavuje samostatná aplikácia, ktorá je však pripravená na použitie iba s konkrétnou aplikáciou (webovým prehliadačom) – vo svojej podstate ide o zásuvný modul webového prehliadača aplikovaný „zvonku“. Tento prístup použili v (Reeder et al., 2001), kde vytvorili nástroj, ktorý zaznamenával akcie počas experimentov a pritom automaticky spúšťal prehliadač Internet Explorer, sledoval akcie v ňom a mapoval údaje zo systému na sledovanie pohľadu na obrazové body prehliadača.

2.8 Diskusia

Ako ukazuje analýza, v oblastiach implicitnej spätnej väzby a konkrétne sledovania posúvania dokumentu a sledovania pohľadu používateľa existujú viaceré riešenia zaoberajúce sa príslušnou oblasťou. Niektoré riešenia overujú metódy pomocou ostatných prístupov (typicky napríklad overenie hypotézy používania myši pomocou kontroly sledovaním pohľadu používateľa a pod.) alebo porovnávajú tieto metódy navzájom (napríklad skúmanie korelácie medzi používateľovým pohybom myšou a zacielením jeho pohľadu). Okrem toho existujú výskumy, ktoré na implicitnú spätnú väzbu nahliadajú komplexnejšie a zbierajú množstvo implicitných indikátorov záujmu, ale ich cieľom je vyhodnotenie týchto indikátorov, priama štúdia správania sa používateľa, atď.

Ďalšie výskumy naopak využívajú zbierané identifikované implicitné indikátory na analýzu práce používateľa s dokumentom do väčšej hĺbky, tak že následne sa zaoberajú vlastnosťami textu (zaujímavé fragmenty), dokumentov a pod., ale tieto nie vždy komplexne používajú dostupné metódy, typicky ide napríklad pri rozsiahlom sledovaní primárne dostupných indikátorov o zamietnutie sledovania pohľadu z dôvodu nákladov a komplexnosti. Tiež je možné do určitej miery badať odčleňovanie indikátorov záujmu pracujúcich na úrovni dokumentov (čas celkového čítania, spôsob opustenia dokumentu a pod.) od indikátorov pracujúcich na úrovni fragmentov (pohľad na konkrétne riadky, čas zobrazenia riadku počas posúvania dokumentu a pod.).

3 DOMÉNA VZDELÁVANIA NA ADAPTÍVNOM WEBE – SYSTEM ALEF

Doména vzdelávania je jedna z najviac prepracovaných oblastí adaptívnych webových aplikácií a práve v tejto doméne má veľký význam sledovanie používateľa. Vo vzdelávacích aplikáciách sa kladie veľký dôraz na model používateľa, u ktorého nás primárne zaujíma, čo sa učí a ako je v tom úspešný, ale aj na doménový model, ktorý slúži ako podpora pre sledovanie učenia používateľa – napríklad aby sme vedeli povedať, ktoré učivo a tým pádom aj skupinu textov už používateľ študent ovláda, musia byť vzdelávacie texty združené v skupinách podľa ich obsahu. Elegantné riešenie je priradenie konceptov dokumentom, zároveň tak môžeme skúmať ich vzájomné vzťahy ako nadväznosť a pod. Doménový model, respektíve samotný obsah vzdelávacieho systému je dôležitý aj vo vzťahu k samotným vzdelávacím textom – používateľov učiteľov zaujíma, ktoré časti vzdelávacieho obsahu sú prehliadané, ktoré sú naopak dôležité, kde sa vyskytujú chyby a podobne.

Identifikácia významných fragmentov textu má viaceré aplikácie v oblasti elektronickej výučby, napríklad identifikácia textu dôležitého pre učenie, sumarizácia výučbového obsahu na účely opakovania alebo zrýchleného učenia. Vhodnou oblasťou pre aplikáciu tohto projektu a zároveň overenie navrhovaných metód sa tak javí oblasť adaptívnej elektronickej výučby, konkrétne využitie systému ALEF.

Systém ALEF (Adaptive LEarning Framework) (Bieliková, Šimko, Barla, et al., 2010; Šimko, Barla & Bieliková, 2010), navrhnutý, implementovaný a prevádzkovaný na Fakulte informatiky a informačných technológií, je príkladom adaptívneho elektronickeho systému, ktorý umožňuje pokročilé funkcie, ako sú anotovanie obsahu používateľmi, tagovanie, práca s externými zdrojmi a bodovanie aktivity používateľov. Tiež poskytuje odporúčanie nasledujúcich výučbových objektov podľa vedomostí používateľa a ostávajúceho času, ktorý je k dispozícii na učenie. Obdobným spôsobom (na základe používateľových vedomostí) aj adaptívne vyberá do výučbových textov vnorené objekty – otázky a cvičenia. Z tohto hľadiska systém závisí od spätnej väzby, či už explicitnej, pri ktorej používateľ hodnotí svoje porozumenie cvičeniu, tak aj implicitnej, pri ktorej systém zaznamenáva činnosti používateľa, napríklad časový sled navštevovania výučbových objektov používateľom.

Základným dokumentom, s ktorým používatelia v tomto systéme pracujú, je vzdelávací objekt (Learning Object). Tento objekt môže byť troch typov:

- cvičenie (Exercise) – používateľovi sa zobrazí zadanie úlohy, ktorú sa pokúsi samostatne vyriešiť, prípadne si môže vyžiadať zobrazenie návodu (pomôcky) a následne si svoje riešenie porovná so vzorovým,
- otázka (Question) – používateľovi sa zobrazí stručná otázka s krátkymi odpoveďami rôznych typov (výber jednej/viacerých správnych odpovedí, doplnenie vynechaných reťazcov, usporiadanie odpovedí do správneho poradia),
- vzdelávací text (Explanation) – samotný učebný text – obsahuje obrázky, odstavce textu, ukážky kódu a pod.

Na obr. 4 je ukážka webového používateľského rozhrania práve zobrazujúceho vzdelávací objekt typu Explanation. Na ľavej strane rozhrania je dominantným prvkom menu, na pravej strane od výučbového objektu sa nachádzajú zásuvne moduly zobrazujúce používateľovi informácie, napríklad skóre a nahlásené chyby, a tiež umožňujúce interakciu používateľa so systémom, napríklad vkladanie externých zdrojov a pridávanie značiek. Niektoré zásuvné moduly, napríklad odporúčanie, sa môžu v závislosti od aktuálnej konfigurácie systému a kurzu pre daného používateľa zobrazovať aj na ľavej strane v blízkosti menu.

Obrázok 4: Ukážka používateľského rozhrania systému ALEF. Zobrazuje menu (vľavo), vzdelávací objekt (v strede, v tomto prípade ide o objekt typu Explanation) a zásuvné moduly (vpravo).

Z hľadiska indikátorov spätnej väzby ako je *read wear* je zaujímavé, že vzdelávací obsah ma dostatočnú veľkosť pre to, aby ho bolo nutné posúvať a zároveň je posúvaný samostatne bez ohľadu na ostatné prvky aplikácie. Pre sledovanie zamerania pohľadu prostredníctvom bežnej webovej kamery s nižšou presnosťou je tiež dôležitá funkcia zásuvných modulov a tiež ich veľkosť, napríklad skóre nie je zobrazované ako malý jednoriadkový text, ale je mu venovaná dostatočná plocha.

Systém ALEF je vytvorený pre použitie vo webovom prehliadači a ako taký je implementovaný v jazyku Ruby s použitím rámca Ruby on Rails. Autor tohto dokumentu sa v rámci predmetov Projektovanie a Odborné praktikum podieľal na vývoji systému ALEF, najmä na viacerých základných prvkoch v ranných fázach vývoja systému (prihlasovanie používateľov, zobrazovanie vzdelávacích objektov, úprava použitého rámca Apotomo, návrh a zmeny modelu údajov) a tiež na zaznamenávaní činností používateľa. V čase písania tejto práce sa už pripravuje a testuje nová generácia systému: ALEF 3.

3.1 Predchádzajúca aplikácia implicitnej spätnej väzby v systéme ALEF

Aj pred zapojením nášho riešenia bola v systéme ALEF využívaná implicitná spätná väzba od používateľov. Zber tejto implicitnej spätnej väzby bol však implementovaný na strane servera a preto boli najmä zachytávané akcie ako navštívenie vzdelávacieho objektu, správne/nesprávne zodpovedanie otázky/cvičenia, navštevovanie položiek ponúknutých používateľovi v menu a odporúčaných položiek. Takáto úroveň do určitej miery postačuje pre účely zisťovania vedomostí používateľa, pričom napríklad otázka má priradený jeden alebo viacero konceptov a pri správnom alebo nesprávnom zodpovedaní je možné zvýšiť alebo znížiť sledovanú znalosť týchto konceptov u používateľa.

Pre niektoré ďalšie účely však takáto spätná väzba nepostačuje. V prípade odporúčania vzdelávacích textov, otázok a cvičení vieme, či ho používateľ použil (navštívil vzdelávací objekt prostredníctvom menu alebo zásuvného modulu odporúčania). Avšak v prípade, že používateľ odporúčania nepoužíva, nevieme, či tak činí z dôvodu, že si odporúčania ani nevšimol, alebo si odporúčania aj všimol, ale túto funkciu nechce používať, pretože má iný spôsob práce alebo ho odporúčané položky neoslovili. So zapojením pohľadu používateľa vieme odlišiť, či si používatelia konkrétny zásuvný modul všímajú.

V prípade, že si modul používateľa ani nevšímajú, môžeme vykonať dizajnové zmeny vo webovej aplikácii pre zvýraznenie daného modulu alebo adaptívne upozorniť na daný modul práve tých používateľov, ktorí si ho nevšimli. V prípade, že si modul používateľa prezerajú, ale napriek tomu ho nevyužívajú, ich vnútorné dôvody už zistiť nemôžeme. Môžeme sa ich však adaptívne spýtať, najlepšie práve v momente, keď si používateľ modul a odporúčané položky prehliadol, ale následne ich nevyužil.

Ďalšie účely, pre ktoré implicitná spätná väzba získavaná zo záznamov na serveri nepostačuje, spočívajú v práci s textom. V prípade zaznamenávania akcií používateľa z pohľadu servera vieme, že navštívil určitý vzdelávací objekt a o niečo neskôr navštívil objekt ďalší, no z tohto samotného času vieme len veľmi obmedzene vyvodit' čas jeho práce s dokumentom, pretože sa mohol venovať aj iným činnostiam. Získať takýmto spôsobom informácie na úrovni častí (fragmentov) dokumentu je už nemožné a vyžaduje zapojenie sledovania indikátorov záujmu zo strany klienta, čiže prehliadača.

4 METÓDA NA ZÍSKAVANIE IMPLICITNEJ SPÄTNEJ VÄZBY

Ako vhodný koncept identifikácie významných fragmentov sa javí využitie času záujmu, pričom najmenším logickým celkom obsahu – slovám, obrázkom a pod., sa priradí určitý index záujmu. Tento index môžeme získať pomocou identifikovaného pohľadu používateľa na tento fragment obsahu. Zároveň je možné do neho skombinovať ďalšie implicitné indikátory súvisiace s pozíciou v dokumente, napríklad označovanie textu, kopírovanie textu, zobrazenie fragmentu na obrazovke (sledovanie posúvania dokumentu), pohybovanie kurzora v okolí fragmentu a pod. Jednotlivým indikátorom záujmu je potrebné určiť vhodné váhy, nakoľko napríklad pohybovanie kurzorom v okolí fragmentu môže používateľ vykonávať náhodne alebo za iným účelom (presun kurzora na iné miesto), kým napríklad pohľad používateľa na daný fragment významne prispieva k zisteniu záujmu o fragment.

Pre tento postup vykonávame viacero krokov:

1. získanie informácií o práci používateľa,
2. naviazanie týchto informácií na fragmenty, na ktoré je potrebné vopred dokument a používateľské rozhranie rozdeliť,
3. spracovanie rôznych identifikátorov patriacich jednotlivým fragmentom:
 - spracovanie identifikátorov u jedného používateľa,
 - agregovanie informácií od množstva používateľov.

4.1 Získanie údajov

Zdroje spätnej väzby týkajúce sa práce používateľa môžeme vzhľadom na doménu webovej aplikácie získať:

- z webovej kamery – pozícia pohľadu používateľa,
- z ovládacích vstupných zariadení – kliknutia, pozícia kurzora,
- z prehliadača – čas zobrazenia časti dokumentu (read wear), označovanie a kopírovanie textu, doménovo-špecifické akcie (anotácie a pod.), vytlačenie dokumentu, vyhľadávania v dokumente.

Pri získavaní údajov uvažujeme aj prípadné nepresnosti a čiastočné akcie. Pohľad používateľa nemožno uvažovať ako osamotený bod, pretože ľudský zrak vníma obrazové informácie aj periférne a používateľ tak pri čítaní odseku nesmeruje presne od jedného okraja ku druhému. Keďže pohľad je snímaný a odhadovaný externým zariadením a to navyše pomocou jednoduchej webovej kamery v neriadenom prostredí, vzniká ďalší zdroj nepresnosti. Z tohto dôvodu je pohľad modelovaný ako kruh, pričom fragmenty zasahujúce do blízkosti stredu získavajú plný odčítaný čas pohľadu, fragmenty vzdialenejšie od stredu získavajú iba časť. Túto časť modelujeme lineárne v závislosti od vzdialenosti fragmentu k odhadnutému bodu pohľadu.

Obdobne, ak je fragment zobrazený čiastočne, nemožno ho považovať za plne zobrazený, ale zároveň ani za nezobrazený. Fragmenty pretínajúce okraj zobrazovacej plochy, v rámci ktorej sa dokument (fragment) posúva, tak získavajú iba časť z času, ktorý bol vyhodnotený. Táto časť je úmerná veľkosti zobrazenej časti fragmentu k jeho celkovej veľkosti.

4.2 Interpretácia údajov na fragmenty

Pre priradenie významnosti fragmentov je najskôr potrebné dokument, s ktorým používateľ pracuje a používateľské rozhranie systému, v ktorom s ním pracuje, rozdeliť na fragmenty, ktorých významnosť vyhodnotíme. Rozdelenie môžeme vykonať aj automaticky, práve na základe spätnej väzby, ale keďže pracujeme v známej doméne konkrétneho adaptívneho webového systému s náročnejším rozhraním, toto rozdelenie môžeme vykonať pomocou vopred zadaných pravidiel, ktoré určujú, ktorá časť webovej stránky predstavuje dokument, ktorá predstavuje navigačné menu a pod.

4.2.1 Rozdelenie dokumentu a používateľského rozhrania na fragmenty

Úrovne vnorených fragmentov dokumentu, ktoré môžeme identifikovať sú:

- vnútro odseku, tabuľky (a1),
- odsek, ucelená časť dokumentu, obrázok, tabuľka a pod. (a2),
- dokument (a3),
- kurz (a4).

Mimo tejto hierarchie fragmentov v dokumente:

- súčasti systému (b1).

V rámci adaptívneho webového výučbového systému ALEF zdefinujeme jednotlivé fragmenty nasledovne:

- v rámci dokumentov (výučbových objektov (a3)):
 - učebný text (*Explanation*) – učebný text je súvislý dokument (a3), členený tradične na odseky (a2) s vnútorným textom (a1), obsahujúci obrázky (a2), tabuľky (a2), atď.,
 - otázka (*Question*) – zadanie (a1) a odpovede (a1), nakoľko sú otázka a najmä jej odpovede často veľmi krátke, na tejto úrovni nemusí byť možné dosiahnuť požadovanú presnosť; celá otázka (a2),
 - cvičenie (*Exercise*) – zadanie (a2), pomôcka (a2), riešenie (a2); ich vnútorný obsah (a1).
- v rámci systému:
 - navigačné menu – (b1),
 - zobrazenie odporúčaní – (b1),

- ďalšie prvky (navigácia pomocou konceptov, hodnotenie používateľa, externé zdroje, atď.) – každý takýto nástroj predstavuje samostatný vizuálny prvok (widget) (b1).

4.2.2 Vzťah spätnej väzby k fragmentom

Pre nadviazanie indikátorov spätnej väzby na fragmenty dokumentu a používateľského rozhrania systému je potrebné zaoberať sa tým, ako a s ktorými fragmentmi súvisia. Indikátory spätnej väzby sme preto rozdelili do skupín *nezacielených* indikátorov, *aktívne a pasívne zacielených indikátorov* a *dokumentových* indikátorov. Indikátory sú zacielené a nezacielené podľa toho, či používateľ vykonáva danú akciu priamo v súvislosti s fragmentom – napríklad samotný pohyb kurzora nad fragmentom nemusí súvisieť s daným fragmentom, používateľ môže presúvať kurzor na iné miesto na obrazovke, avšak priamo už označovanie textu v danom fragmente je zacielený indikátor spätnej väzby. Aktívnym a pasívnym zacielením označujeme akcie podľa toho, ako používateľ vykonáva danú akciu – fragment je *pasívne* zobrazený (read wear) a prehlíadaný (pohľad), používateľ *aktívne* označuje časti textu a kliká. V tab. 2 je uvedené rozdelenie indikátorov aj s návrhom ich váh a metrík.

Tabuľka 2: Počiatočný návrh váh a metrík indikátorov záujmu.

Nezacielené ($w_G = 0,2$)		
	w	[h]
Pohybovanie kurzorom	0,5	vzdialenosť v bodoch
Kliknutie (bez významu)	1,0	počet
Pasívne zacielené ($w_G = 2,0$)		
	w	[h]
Zobrazenie (read wear)	1,0	čas v minútach
Pohľad	4,0	čas v minútach
Aktívne zacielené ($w_G = 1,0$)		
	w	[h]
Označovanie textu	1,0	počet, dĺžka v znakoch
Skopírovanie textu	2,0	počet, dĺžka v znakoch
Doménovo-špecifické akcie (vloženie anotácie a pod.)	4,0	počet
Kliknutie na odkaz (alebo pod.)	1,0	počet
Výsledok vnútrostránkového hľadania	1,0	počet
Dokumentové ($w_G = 0,2$; pre fragmenty systému nesúvisiace s dokumentom $w_G = 0,0$)		
	w	[h]
Vytlačenie dokumentu	1,0	počet
Počet návštev	1,0	počet

4.3 Zostavenie charakteristík fragmentov

Na nadviazanie indikátorov na významnosť fragmentu použijeme nasledujúci postup: každému sledovanému fragmentu f priradíme index záujmu AI_f (Attention Index), ktorý vypočítame takto: $AI_f = \frac{\sum_i (w_{Gi} \cdot \sum_j w_j \cdot v_{fj})}{|f|}$, kde v_{fj} je hodnota j -teho indikátora

spadajúceho na fragment f ; w_j je váha j -teho indikátora; w_{G_i} je váha i -tej skupiny indikátorov.

Index záujmu pre zložený fragment F skladajúci sa z viacerých fragmentov s nenormalizovaným indexom záujmu AI'_{f_i} je takýto: $AI_F = \frac{\sum_i AI'_{f_i}}{\sum_i |f_i|}$. Normalizácia sa teda vykonáva vždy až na aktuálne skúmanej úrovni fragmentov, pretože sčítaním už normalizovaných indexov záujmu by sme potlačili vzájomné závislosti medzi vnorenými fragmentmi v rámci zloženého fragmentu.

Pri porovnávaní ohodnotenia fragmentov získaných výpočtom s rôznymi váhami je potrebné normalizovať ohodnotenia fragmentov navzájom na aktuálne uvažovanej úrovni hĺbky fragmentov.

4.4 Diskusia

V navrhutej metóde v kroku rozdelenia dokumentov a webovej aplikácie na fragmenty využívame rozdelenie dodané samotnou webovou aplikáciou, vytvorené mimo našej metódy, či už človekom – administrátorom webovej aplikácie, alebo inou automatizovanou metódou. Vzniká tu určitý priestor na automatizované delenie dokumentu a aplikácie na fragmenty práve na základe samotnej nami sledovanej implicitnej spätnej väzby, no ani v rámci aplikácií Web 2.0 ešte stále nie je bežné úplné oddelenie údajov od formy – webového dokumentu od jeho zobrazenia a samotný dokument je zaťažovaný rôznymi atribútmi a elementmi určenými pre formátovanie, čo môže automatizované delenie sťažiť. Taktiež môže dochádzať k nepredvídateľným zmenám v poskytovaných dokumentoch, napríklad v prípade anotovania používateľmi, kde môžu vznikať a zanikať ďalšie elementy takmer v ľubovoľnom mieste dokumentu, čo by daná metóda automaticky rozdeľujúca dokument a aplikáciu musela taktiež uvažovať.

Jedným z hlavných cieľov našej práce je preskúmanie možnosti využitia pohľadu používateľa na webe a preto sa zameriavame na identifikáciu dôležitosti už špecifikovaných fragmentov. V predostretej metóde uvažujeme okrem pohľadu aj zobrazenie dokumentu a ďalšie implicitné indikátory záujmu používateľa, najväčšiu váhu však pritom pripisujeme práve pohľadu a zobrazeniu dokumentu, spolu s detekciou nečinnosti, respektíve inej činnosti používateľa, takže ako zobrazenie dokumentu sa uvažuje len čas, kedy používateľ s dokumentom aktívne pracuje.

5 VYUŽITIE INFORMÁCIÍ O PRÁCI POUŽÍVATEĽA S DOKUMENTOM

Získané informácie o práci používateľa s webovým dokumentom môžeme využiť viacerými spôsobmi – buď v súvislosti so samotným dokumentom alebo v súvislosti so samotným používateľom (používateľmi).

5.1 Významné fragmenty

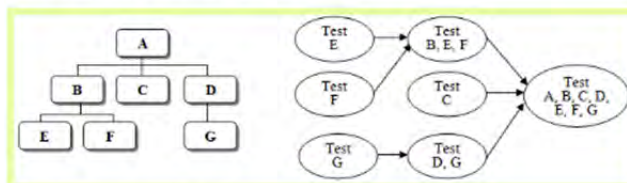
Na základe vypočítaného indexu záujmu o fragmenty dokumentu je možné tieto informácie používateľovi odovzdať v rôznej forme, či už priamou vizualizáciou, vizualizáciou so skrytím málo dôležitých fragmentov (sumarizácia) alebo pomocou zmeny správania sa posuvníka.

5.1.1 Priama vizualizácia

Používateľovi zobrazíme ako pomôcku pri učení (respektíve inej práci s dokumentom), ktoré fragmenty sú ako ohodnotené. Využijeme na to zvýraznenie samotného fragmentu (textu) zmenou farby pozadia. Zvýrazniť môžeme fragmenty s najvyšším indexom ako fragmenty, ktorým je potrebné venovať pozornosť a zároveň fragmenty s najnižším indexom ako fragmenty, ktorým sa postačuje venovať kratšie. Zároveň tým vytvoríme odporúčanie na úrovni dokumentu.

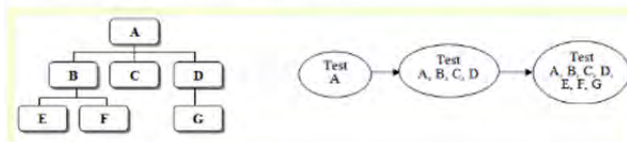
Testovanie zdola-nahor angl. bottom-up testing

- vhodné, ak veľa modulov "nižšej" úrovne predstavuje všeobecné moduly, ktoré iné časti systému často používajú



Testovanie zhora-nadol angl. top-down testing

- problémy so simuláciou modulov na nižších úrovniach
- obmedzená znovupoužiteľnosť modulov



Jednofázové testovanie angl. big-bang testing

- moduly sa otestujú samostatne a potom sa naraz integrujú
- hodí sa iba pre malé systémy

Obrázok 5: Grafická vizualizácia dôležitosti fragmentov.

5.1.2 Sumarizácia

Sumarizácia učebných textov sa rovnako odvíja od indexu záujmu o fragmenty (vyberieme fragmenty s najvyšším indexom záujmu), no na rozdiel od odporúčania na úrovni dokumentu formou vizualizácie v inak nezmenenom dokumente vynecháva sumarizácia celé úseky dokumentu. Môže tak pracovať aj na úrovni medzi dokumentmi, keď pospája najvýznamnejšie fragmenty z rôznych dokumentov a nevýznamné nezobrazí. Takáto funkcia je však v prípade vzdelávacieho systému až nebezpečná, pretože používatelia síce môžu časť dokumentu často prehliadať, pretože sa im zdá jednoduchá, no to neznamená, že ju môžeme úplne skryť a nasledujúcim používateľom vôbec nezobraziť.

5.1.3 Adaptívny posuvník

Zaujímavým konceptom, pôvodne určeným skôr pre malé displeje prenosných zariadení, je posuvník uvažujúci posúvaný obsah, čiže automaticky prispôsobujúci priblíženie, rýchlosť a smer v druhej osi. Tento posuvník je však možné nasadiť aj na bežných displejoch a systémoch a využiť ho práve na „označenie“ významných fragmentov – posuvník tak môže nad zobrazením málo významných fragmentov posúvať obsah v závislosti od pohybu myši rýchlejšie alebo v závislosti od stláčania kláves viac, pričom naopak nad zaujímavými fragmentmi môže posúvať pomalšie/po menších úsekoch.

Manipulácia priblíženia nie je na webových systémoch používaných na stolných počítačoch taká častá ako na prenosných zariadeniach a nie vždy je zmene priblíženia obsah prispôsobený. Taktiež, keď je dokument členený do fragmentov takmer výhradne zvislo, posúvanie v druhej osi nemá veľký význam. Preto by v uvažovanom kontexte webových aplikácií mal takýto posuvník význam iba pri práci s rýchlosťou posúvania vo zvislom smere, bez úprav pozície na druhej osi a celkového priblíženia.

5.2 Sprievodca prácou v systéme

Pri sledovaní fragmentov systému máme k dispozícii informácie o práci používateľa s prvkami systému ako sú odporúčanie, navigačné menu a pod. Na základe týchto informácií vytvárame „sprievodcu“ prácou v systéme, ktorý používateľovi na základe informácií o jeho práci napomáha – napríklad v prípade, že používateľ nepoužil odporúčanie, napovie mu, kde odporúčanie nájde a aký má pre neho prínos. S využitím informácií o pohľade používateľa vieme rozlíšiť aj prípad, kedy používateľ možnosti odporúčania nepoužíva, lebo si odporúčané odkazy nevšima, alebo ich pohľadom prezrel, ale na odkazy neklikol. V prvom prípade postačuje používateľovi poukázať na umiestnenie odporúčaných odkazov, v druhom prípade mu môžeme vysvetliť, prečo je odporúčanie pre neho dobré a priamo mu môžeme dať možnosť vyjadriť sa, prečo odporúčanie nevyužil a poskytnúť tak cennú explicitnú spätnú väzbu.

Iný prípad zobrazenia výzvy a možnosti na poskytnutie explicitnej spätnej väzby nastane napríklad v momente, keď používateľ viackrát opakovaním kliká na rovnakú

položku menu – môže tak konať napríklad pri dlhších odozvách systému alebo pri chybe počas zobrazovania kliknutej položky.

Rozdiel medzi explicitnou spätnou väzbou, pri ktorej používateľa obťažujeme požiadavkami na vyjadrenie sa neustále alebo pri náhodných udalostiach a medzi explicitnou spätnou väzbou získavanou týmto spôsobom je, že sprievodca prácou v systéme požaduje explicitnú spätnú väzbu práve v momente, kedy by ju používateľ rád poskytol (je nespokojný s odporúčanými objektmi, opakovane kliká do menu, pretože kvôli dlhšej odozve si nie je istý, či klikol správne a podobne). Mali by sme tak získať kvalitnejšiu explicitnú spätnú väzbu.

5.3 Obohatená komunikácia používateľov

Pri sledovaní používateľa vieme informácie o práci používateľa ešte nepremenené na index záujmu o fragment použiť na zvýšenie kolaborácie používateľov prostredníctvom obohatenia komunikácie. Obohatenia spočíva v rozšírení jednoduchého nástroja na zasielanie okamžitých správ medzi používateľmi o indikáciu pozícií ostatných používateľov – primárne prostredníctvom sledovania posúvania (opotrebovania) zobrazeného dokumentu, zobrazovať je však možné aj výstupy sledovania pohľadu, aj keď pohľad je oveľa menej stály ako posúvanie dokumentu a vo webovej aplikácii s väčším intervalom získavania informácií zo servera je asi nevhodný.

Obohatenie komunikačného nástroja o pozície spolužiakov v aktuálnom dokumente by malo umožniť študentovi kontaktovať práve tých, ktorí sa v tomto momente nachádzajú na rovnakej časti ako on, čím by ich jednak menej vyrušoval a taktiež je predpoklad, že od osloveného spolužiaka, ktorý práve s danou témou v dokumente pracuje na rovnakom mieste, získa študent lepšiu pomoc, než od spolužiaka, ktorý by musel dané miesto hľadať a znovu sa nad ním zamýšľať.

Nevýhodou je, že používatelia už majú v súčasnosti svoje vlastné spôsoby komunikácie (napríklad nástroje okamžitého zasielania správ (angl. *instant messaging*) mimo webových systémov a prehliadačov alebo nástroje dostupné v rámci rôznych sociálnych sietí).

6 NÁVRH REALIZÁCIE A IMPLEMENTÁCIA METÓDY

6.1 Prepojenie sledovania pohľadu s webovým prehliadačom

Špecifikom tohto projektu je nasadenie sledovania pohľadu používateľa. Z tohto dôvodu je potrebné, aby časť sa riešenia (preberanie obrazu z webovej kamery a hľadanie pohľadu) vykonávala lokálne natívne u používateľa. Implementácia celého zberu spätnej väzby priamo vo forme zásuvného modulu prináša možnosť sledovať prehliadač, ale zároveň umožňuje zahrnúť aj sledovanie pohľadu. Variant implementácie vlastného prehliadača nie je prakticky realizovateľný z dôvodu plánovaného využitia spolu s výučbovým systémom ALEF, ktorý extenzívne využíva možnosti technológií JavaScript (a AJAX), CSS a ďalších.

Celkovo sa teda v súčasnosti ako najvýhodnejší variant ukazuje zber spätnej väzby prostredníctvom zásuvného modulu do prehliadača. Aktuálne najviac používanými webovými prehliadačmi sú napríklad podľa štatistík spoločnosti Wikimedia za mesiac september¹² prehliadače: Microsoft Internet Explorer (46,45 %), Mozilla Firefox (29,33 %), Google Chrome (9,00 %), Safari (5,49 %) a Opera (3,19 %). Presné podiely používania prehliadačov mesačne kolíšu a do veľkej miery závisia aj od konkrétnej webovej stránky, webové sídlo zamerané na všeobecné informácie ako sú portály spoločnosti Wikimedia, napríklad encyklopédia, slovník, má iné zloženie návštevníkov a tým pádom aj rozloženie nimi používaných prehliadačov, než napríklad webové sídlo zamerané na programátorov a webdizjanérov. Významný rozdiel v rádoch percent medzi prehliadačmi Internet Explorer a Firefox a ostatnými však naznačuje, že tieto dva prehliadače sú výrazne používanéjšie. Vzhľadom na zásadné odlišnosti rozširovania týchto prehliadačov je vhodné zvoliť si práve jeden z nich, a preto na základe domény overenia riešenia – webového adaptívneho výučbového rámca ALEF, ktorý je používaný v prehliadači Firefox, sme zvolili pre riešenie práve tento prehliadač.

Prehliadač Firefox je možné rozširovať dvomi spôsobmi, vytváraním zásuvných modulov (angl. *plugin*) a rozšírení (angl. *add-on*). Zásuvné moduly sú vytvárané prostredníctvom technológie NPAPI a sú vykonávané ako natívny kód. Určením zásuvných modulov je, že sa vkladajú do webovej stránky, typický príklad zásuvného modulu je prehrávač Adobe Flash. Sú ohraničené na samotnú stránku s ukončením pri opustení stránky (navigácii na inú stránku) a opätovným načítaním na novonavštvivenej stránke, čo predstavuje nevýhodu pre zber spätnej väzby.

Naproti tomu rozšírenia sa vykonávajú priamo v prehliadači, nezávisle od otváraných a zatváraných webových stránok. Rozšírenia však nie sú plne natívny kód, ale sú vytvárané v jazyku JavaScript, avšak s vyššími právami než kód v jazyku JavaScript vložený do webovej stránky a má prístup k funkciám samotného prehliadača. Grafické

¹² Wikimedia Traffic Analysis Report - Browsers e.a. [online]. 2010 [cit. 2010-12-05]. http://stats.wikimedia.org/archive/squid_reports/2010-09/SquidReportClients.htm

rozhranie je vytvárané v jazyku XUL. Rozšírenie môže obsahovať zásuvný modul, opačne však zásuvný modul nemôže obsahovať rozšírenie.

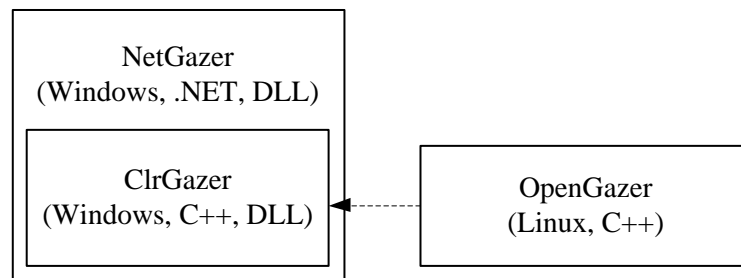
Na naše účely sme sa preto rozhodli vytvoriť rozšírenie (*extension*) prehliadača Firefox. Dôležitou súčasťou riešenia je tak prepojenie rozšírenia bežiaceho v skriptovacom jazyku JavaScript s riešením sledovania pohľadu používateľa NetGazer, ktoré je natívne (respektíve beží v rámci .NET Framework). Preskúmali sme preto viaceré možnosti takéhoto premostenia:

- *Sledovanie pohľadu ako súčasť rozšírenia.* Okrem možnosti obalenia aplikácie na sledovanie pohľadu do NPAPI zásuvného modulu a zahrnutia tohto modulu do rozšírenia má prehliadač Firefox aj prirodzenejšie mechanizmy práce s natívnym kódom – sledovanie pohľadu sa obalí do objektu modelu XPCOM alebo dynamickej knižnice DLL a stane sa tak súčasťou rozšírenia s možnosťou priameho použitia z kódu v jazyku Javascript. Použiteľné prístupy:
 - *XPCOM (Cross Platform Component Object Model), XPConnect (Cross Platform Connect)* – XPCOM predstavuje komponentový model od spoločnosti Mozilla, ktorý umožňuje v rámci aplikačného prostredia prístup k funkciám správy pamäte, komunikácie objektových správ, sieťovej komunikácie, objektového modelu dokumentu (DOM), atď. Pomocou tohto komponentového modelu je možné vytvárať natívne komponenty a následne pomocou technológie XPConnect je možné prepojiť kód v jazyku JavaScript (rozšírenie) s vlastnými objektmi XPCOM. Nevýhodou tohto prístupu je strata výkonu pri prevádzaní objektov medzi rôznymi jazykmi, ako i meniaci sa podpora medzi rôznymi verziami prehliadača Firefox 3.6 a 4.
 - *ctypes.jsm* (pôvodný názov js-ctypes) – tento kódový modul umožňuje volanie knižničných funkcií kompatibilných s jazykom C (knižnice DLL na operačnom systéme Windows, dynamické zdieľané objekty DSO na operačných systémoch rodiny Unix) z kódu v jazyku JavaScript v rozšírení prehliadača. Výhodou tohto riešenia je väčšia jednoduchosť, než v prípade vytvárania XPCOM objektov, avšak tento mechanizmus bol prvýkrát zahrnutý v prehliadači Firefox 4, ktorý bol v čase implementácie metódy v testovacích fázach a nebol rozšírený medzi potenciálnymi používateľmi.
- *Sledovanie pohľadu ako samostatná aplikácia.* Aplikácia na sledovanie pohľadu môže byť samostatnou natívnou aplikáciou a s rozšírením komunikovať prostredníctvom lokálnej sieťovej komunikácie (sockety). Použiteľné prístupy:
 - *JNEXT (Javascript Native EXTensions)* – tento otvorený rámec umožňuje využívať natívne funkcie (TCP sockety, súbory, vlákna, atď.) z jazyka JavaScript práve pomocou zahrnutia rámcom poskytovaného NPAPI alebo ActiveX zásuvného modulu (podľa použitého prehliadača, podporované sú viaceré), ktorý sprostredkúva vykonávanie natívneho kódu. Výhodou tohto riešenia je jeho jednoduchosť, nevýhodou nutnosť samostatnej inštalácie. Taktiež bola najnovšia verzia vydaná v decembri 2008, čo v súvislosti s využitým mechanizmom zásuvných modulov spôsobuje, že tento rámec nefunguje na súčasných aktuálnych prehliadačoch úplne bezchybne.

- *jsLib* – táto knižnica fungujúca ako rozšírenie prehliadača poskytuje prístup k funkciám siete (sockety, HTTP požiadavky), súborov, atď., pričom jej cieľom je odstránenie zložitosti vytvárania XPCOM komponentov pri vytváraní rozšírenia. Výhodou je, že knižnica je aktualizovaná a udržiavaná, avšak práve súčasť knižnice sprostredkujúca komunikáciu cez sockety sa v súčasnosti nachádza v neaktuálnom stave a je odporúčané ju nepoužívať.
- *služby prehliadača dostupné prostredníctvom technológie XPConnect* – prehliadač Firefox sám poskytuje prostredníctvom technológie XPConnect prístup ku komponentom umožňujúcim sieťovú komunikáciu. Ide o princíp využitý v knižnici *jsLib*. Dostupná je implementácia používajúca tento princíp, ktorá je vyvinutá špeciálne práve pre jednoriadkovú textovú príkazovú komunikáciu rozšírenia s externým programom (zariadením)¹³.

Na obr. 6 je znázornená architektúra riešenia NetGazer – toto riešenie sa skladá z dvoch častí:

- ClrGazer – upravená knižnica OpenGazer pôvodne vytvorená pre operačný systém Linux, táto časť je vytvorená v CLR C++ a je kompilovaná do knižnice DLL,
- NetGazer – obalovač DLL knižnice ClrGazer do jazyka C#, táto časť je kompilovaná do knižnice .NET DLL.



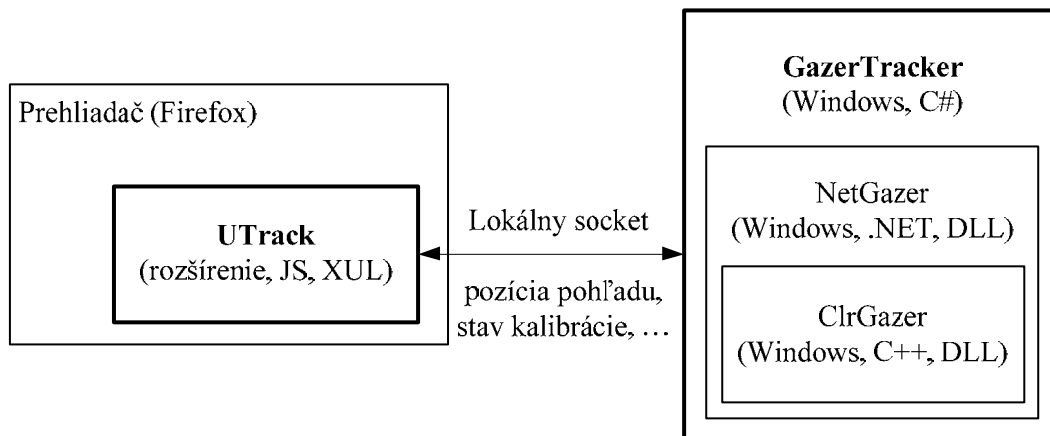
Obrázok 6: Základné časti riešenia NetGazer.

Pretože technológie XPCOM a ctypes.jsm nepodporujú platformu .NET, pri využití jednej z týchto dvoch technológií by bolo potrebné buď vyňať časť riešenia ClrGazer, čím by sme však stratili niektoré funkcie obalovača v časti NetGazer, ktoré by sme museli doplniť v XPCOM/C obalovači, alebo by sme museli opäť obaliť celú knižnicu NetGazer do XPCOM komponentu alebo C knižnice. Toto viacnásobné obalovanie je nevýhodné z hľadiska zložitosti aj výkonu.

Naproti tomu, riešenie so samostatnou aplikáciou je jednoduchšie a tiež umožňuje využitie sledovania pohľadu v ľubovoľných ďalších aplikáciách podporujúcich lokálnu sieťovú komunikáciu, bez ohľadu na použité technológie a schopnosť práce s objektmi XPCOM a knižnicami DLL. Zvolené riešenie je znázornené na obr. 7: vytvorili sme

¹³ MORENCY, B.: Jslib mailing list [online]. 2007 [cit. 2011-05-05]. <http://www.mozdev.org/pipermail/jslib/2007-August/000897.html>

samostatnú aplikáciu *GazerTracker*, ktorá spravuje pripojenie kamery, kalibráciu a sledovanie pohľadu a poskytuje informácie o pohľade ostatným aplikáciám.

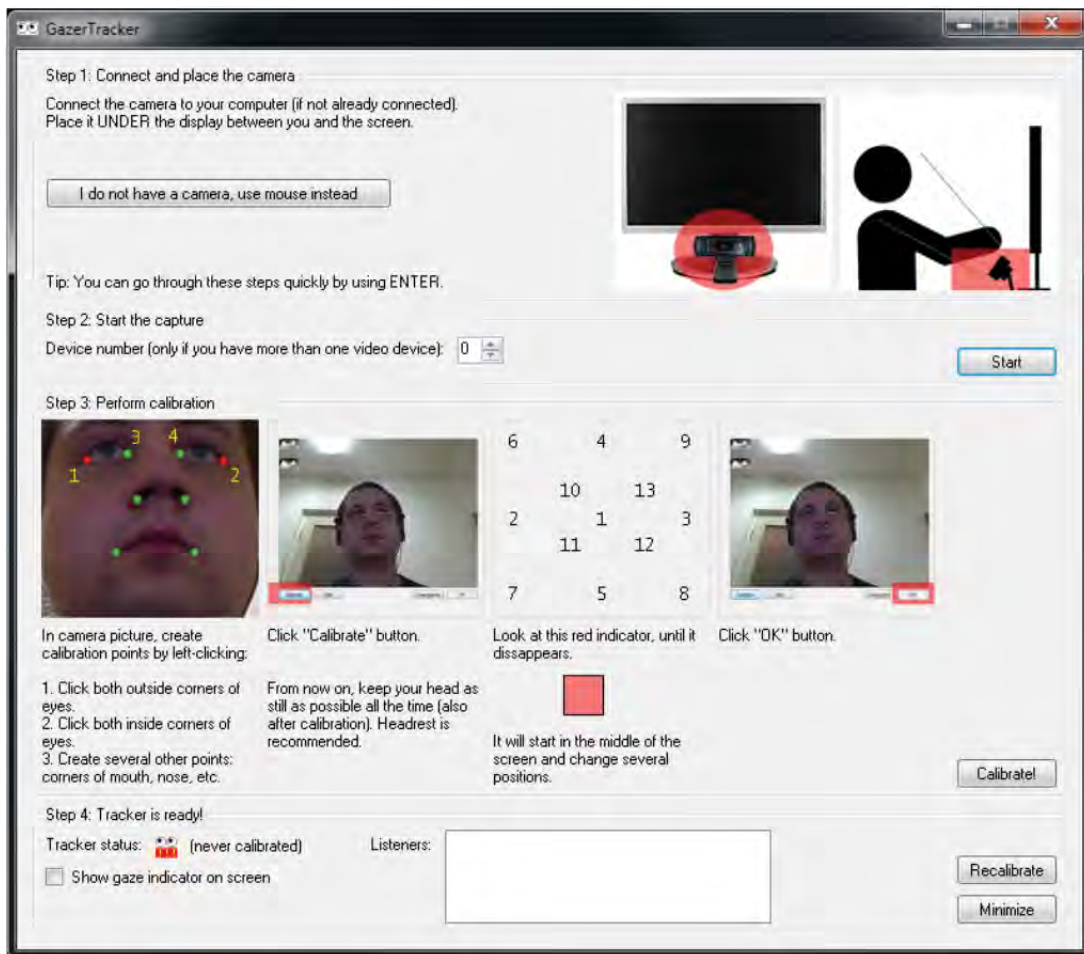


Obrázok 7: Aplikácia *GazeTracker* a prepojenie s rozšírením *UTrack* – oddelenie sledovania pohľadu používateľa od rozšírenia prehliadača.

6.2 Aplikácia na sledovanie pohľadu GazerTracker

Aplikácia GazerTracker obsahuje knižnicu NetGazer a umožňuje používateľovi zvoliť číslo kamery pripojenej k počítaču, vykonať kalibráciu a prípadne nechať si zobrazit' aktuálne zistenú pozíciu pohľadu. Okrem toho umožňuje používateľom, ktorí nedisponujú kamerou využiť na simulované zisťovanie pohľadu pozíciu kurzora myši. Používateľské rozhranie (obr. 8) obsahuje návod, ktorý používateľa postupne inštruuje ako má umiestniť kameru a vykonať kalibráciu. Po vykonaní kalibrácie používateľ aplikáciu minimalizuje do systémovej lišty, v ktorej sa zobrazuje aktuálny stav kalibrácie, a pokračuje v práci.

Aplikácia načúva ako server na lokálnej adrese *127.0.0.1* (tzv. loopback) na porte *56 453*. Po pripojení nového klienta mu odošle svoju identifikáciu „*GazerTracker vX.XX\n*“ a uvítaciu správu „*HELLO {číslo klienta}\n*“. Následne môže klient odosielať textové riadkové príkazy, ktorými môže oznámiť svoje meno, spýtať sa na verziu aplikácie, vyžiadať si aktuálny stav pohľadu a kalibrácie („*STATUS\n*“), nechať aplikáciu zobrazit' konfiguračné okno (má zmysel, ak je minimalizovaná v systémovej lište) a aplikáciu ukončiť. Okrem dotazov na aktuálnu zistenú pozíciu pohľadu sa môže klient prihlásiť na odber stavu („*SUBSCRIBE\n*“) a v tom prípade bude aplikácia GazerTracker posielat' klientovi stav pohľadu po každom cykle výpočtu z obrazu kamery a tiež udalosti získania/straty kalibrácie. GazerTracker podporuje pripojenie viacerých klientov súčasne. Podrobnejšie je formát komunikácie aplikácie opísaný v technickej dokumentácii, ktorá je súčasťou tejto práce.



Obrázok 8: Aplikácia na sledovanie pohľadu GazerTracker. Používateľské rozhranie je vyhotovené vo forme postupného návodu. (Počas používania sú všetky kroky okrem aktuálneho vizuálne potlačené, takže v každej chvíli používateľ vidí výrazne iba tú časť, ktorú má aktuálne vykonať.)

Aplikácia je vytvorená v aplikačnom rámci .NET Framework 3.5 ako aplikácia Windows Forms a využíva mierne modifikovanú knižnicu NetGazer, v ktorej sme vykonali úpravy vo vizualizácii procesu kalibrácie (skrytie interpolovaného indikátora), odstránili sme chyby prejavujúce sa ukončením aplikácie vznikajúce náhodne pri strate kalibrácie a umožnili sme pripojenie aj inej kamery, než s prvým poradovým číslom v systéme.

6.3 Rozšírenie prehliadača UTrack

Získavanie informácií o pohľade ale i iných identifikátorov spätnej väzby obstaráva nami implementované rozšírenie prehliadača UTrack. Rozšírenie bolo vytvorené s cieľom minimálneho dopadu na spôsob práce používateľa a tak jeho základným grafickým prvkom je plocha v stavovej lište prehliadača (obr. 9).



Obrázok 9: Rozšírenie na sledovanie používateľa UTrack, zobrazenie v stavovej lište prehliadača.

Rozšírenie prehliadača bolo zároveň implementované tak, aby používateľom umožňoval vykonávať bežnú prácu vo viacerých záložkách a oknách, iba otvorenie nových okien prehliadača bolo v širšej verzii rozšírenia zablokované z dôvodu výkonu – každé nové okno prehliadača predstavuje nový samostatný prehliadač založený na dokumente *browser.xul* a tak dochádza k viacnásobnej inicializácii nášho rozšírenia, pre každé okno existuje zvlášť. Synchronizácia sledovania okien medzi sebou (vzor Singleton prostredníctvom JS modulu), propagácia aktualizácie stavu grafického rozhrania z práve prijímajúceho okna do ostatných (prihlasovanie a odhlasovanie okien) a ďalšie komplikácie sú však vyriešené a v prípade potreby je možné jednoducho povoliť prácu s viacerými oknami, no treba počítať so zníženým výkonom.

Indikátory spätnej väzby sú sledované rôznymi spôsobmi podľa ich pôvodu. Pre pohľad rozšírenie otvorí automaticky po spustení aplikáciu GazerTracker a prihlási sa k nej na príjem správ o stave pohľadu a kalibrácie, ktoré následne asynchrónne prijíma na vytvorenom lokálnom sieťovom spojení. Indikátor zobrazenia na obrazovke (posúvania – read wear) sa aktualizuje zachytávaním práve viditeľných prvkov v pravidelných intervaloch, pričom práve zobrazeným prvkom a prvkom, v ktorých blízkosti sa nachádza pohľad, sa pripočíta uplynulý čas od posledného spustenia časovača aktualizácie, pomerne znížený v prípade vzdialenosti od odhadnutého stredu pohľadu alebo čiastočného prekrytia prvku okrajovou pozíciou okna, v rámci ktorého sa obsah posúva. Identifikátory ako strata kontextu prehliadača (čo znamená, že používateľ pracuje s inou aplikáciou) sú implementované zachytením priamo správ prehliadača generovaných pri takejto udalosti.

Polomer kruhu, v rámci ktorého sa fragmentu ešte priradí o vzdialenosť lineárne znížený čas pohľadu, sme na základe testov presnosti sledovania pohľadu (pozri nasledujúcu kapitolu) určili na 100 bodov (obr. 10).

Aby mohli byť nezávisle sledované rôzne záložky, bolo potrebné pre každú záložku vytvoriť skupinu údajov, ktoré sa priradujú záložke v dobe načítania novej adresy:

- Informácie zistené z načítanej stránky (pozri ďalej) – ktoré fragmenty sa majú sledovať, ako nájsť dokument v dokumentovom objektovom modeli stránky, ako identifikovať z navštívenej stránky dokument a používateľa webovej aplikácie a kam odosielať spätnú väzbu.
- Informácie potrebné pre spracovanie spätnej väzby – čas posledného prechodu aktualizáčného časovača nad danou záložkou, počiatočný čas návštevy záložky, celkový aktívny čas používateľa v záložke, priradená spätná väzba, atď.

The screenshot shows the ALEF web application interface. At the top, there is a navigation bar with tabs for 'Administrácia', 'Debug', 'SI', and 'C'. The user is logged in as 'Martin Labaj (administrátor)'. The main content area displays the course '10.4 Stratégie testovania'. On the left, there is a sidebar with a list of topics, including '1 Úvod do softvérového inžinierstva' and '10.1 Cieľ verifikácie a validácie'. The main content area shows a list of topics under 'prístup k testovaniu celého systému', including 'testovanie zdola-nahor', 'testovanie zhora-nadol', 'jednofázové testovanie', 'sendvičové testovanie', and 'testovanie porovnávaním'. Below this list is a diagram showing the relationship between 'Vodiaci modul', 'Testovaný modul', and 'testovacie údaje'. The diagram indicates that 'Vodiaci modul' sends 'testovacie údaje' to 'Testovaný modul', which returns 'výsledky' to 'Vodiaci modul'. The right sidebar contains user statistics, including 'Tvoje skóre: 13.9', and a list of external resources like 'Software Testing Strategy' and 'Back-To-Back Testing'.

Obrázok 10: Uvažovanie nepresnosti odhadnutého pohľadu – do vzdialenosti 100 obrazových bodov od odhadovaného stredu pohľadu sa fragmentu priradí lineárne znížená časť času zodpovedajúceho práve vyhodnocovanému časovému okamihu.

6.4 Interakcia rozšírenia prehliadača s webovou aplikáciou

Aby bolo rozšírenie použiteľné pre ľubovoľnú webovú aplikáciu, nielen pre vybraný systém – ALEF, údaje závislé od konkrétnej webovej aplikácie načítavame zo súboru *utrack_def.txt*, ktorý sa pokúsime nájsť u každej navštívenej webovej stránke a ak daná aplikácia tento súbor poskytne, načítame z neho údaje opisujúce rozhranie zloženie používateľského rozhrania aplikácie pre prácu s aplikačnými a dokumentovými fragmentmi a aplikačné rozhranie pre odosielanie získanej spätnej väzby webovej aplikácii. Princíp využitia špecifikovaného súboru je podobný súboru *robots.txt*, ktorý je používaný webovými stránkami na ovládanie prehľadávačov webu napríklad pri indexácii stránok do vyhľadávačov.

Pretože niektoré webové aplikácie odpovedajú aj na adresy, ktoré neobsahujú a snažia sa používateľa presmerovať na iný obsah, pri pokuse o prevzatie definičného súboru môže byť vrátený ľubovoľný dokument z webového servera a nie vždy je zaslaný zodpovedajúci kód protokolu HTTP znamenajúci presmerovanie. Preto je povinnou súčasťou definičného súboru hlavička, ktorá sa musí uviesť ako prvý riadok definičného súboru.

Niektoré definície v rámci riadkov obsahujú výrazy používané ako selektor vo formáte knižnice jQuery¹⁴ pre jazyk JavaScript, ktorá je štandardne využívaná vo webových aplikáciách a je aj použitá v rozšírení UTrack. Tieto výrazy umožňujú veľmi presne vybrať ľubovoľný prvok dokumentu DOM. Pomocou rôznych konštrukcií používajúcich metaznaky je možné vyberať objekty podľa triedy, identifikátora,

¹⁴ <http://jquery.com/>

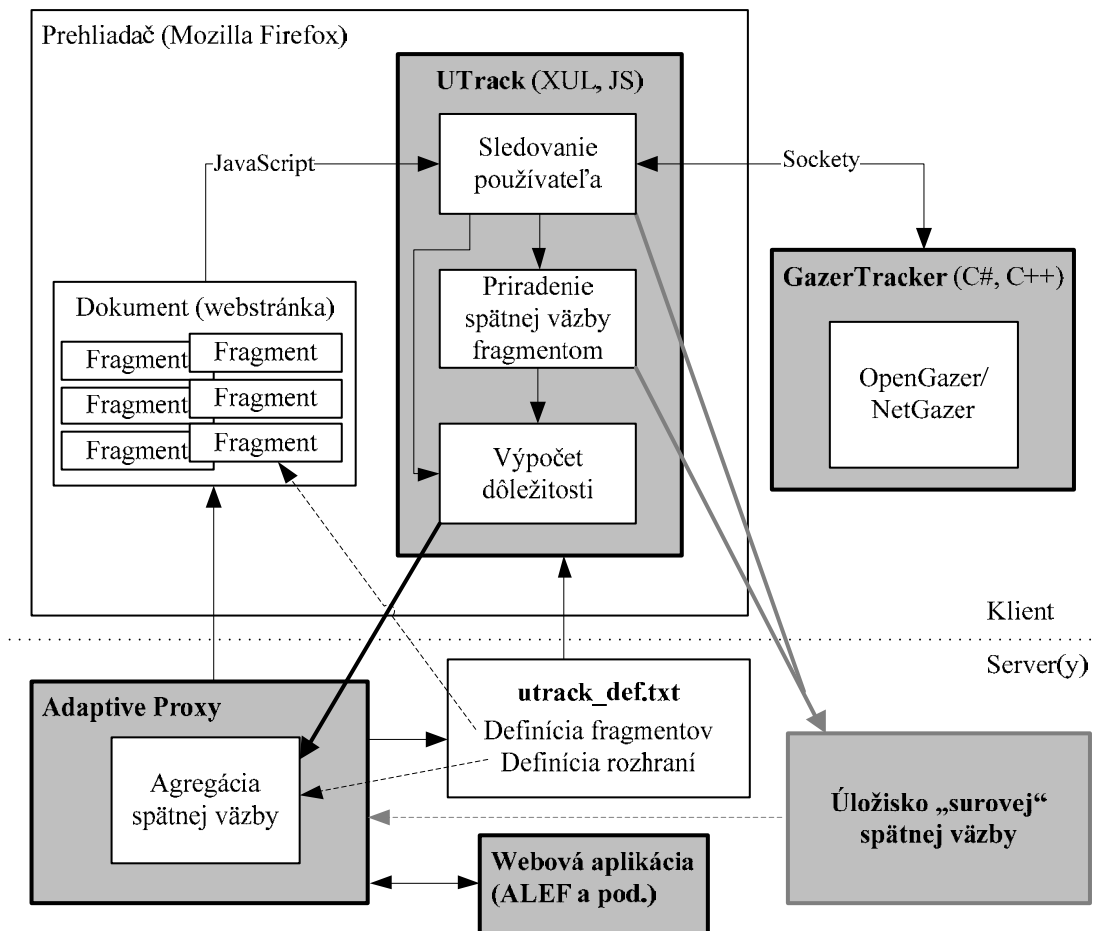
vyberať rodičov, súrodencov a potomkov objektov a pod. Definícia pomocou týchto selektorov umožňuje flexibilitu spoluprácu rozšírenia s webovou aplikáciou bez ohľadu na konkrétnu implementáciu, štýlovanie a rozloženie používateľského rozhrania aplikácie.

Definičný súbor tiež špecifikuje, na ktorých adresách URL očakáva webová aplikácia spätnú väzbu z rozšírenia. Formát spätnej väzby je predpísaný rozšírením a využíva formát JSON, no vďaka štandardnému využitiu GET, respektíve POST požiadaviek s URL parametrami a nezávislosti štandardného (RFC 4627) formátu JSON od konkrétneho jazyka, je formát spätnej väzby spracovateľný v širokej škále bežne používaných implementačných jazykov webových aplikácií. Stručný opis definičného súboru je uvedený v tab. 3.

Tabuľka 3: Definičný súbor `utrack_def.txt`.

Riadok	Význam
REALUTRACKDEF!	Hlavička (povinne prvý riadok súboru).
<code>userdef^{jQuery sel.}</code>	jQuery výraz pre prvok obsahujúci identifikáciu používateľa.
<code>documentdef^{regexp}</code>	Regulárny výraz pre identifikáciu dokumentu z adresy URL.
<code>fragmenturl^{url}</code>	URL adresa pre odosielanie celkovej spätnej väzby k fragmentom.
<code>overgazeurl^{url}</code>	URL adresa pre odosielanie okamžitej spätnej väzby pri prekročení definovaného záujmu o jeden fragment.
<code>fragment^{name}^{jQuery sel.}</code>	Názov a jQuery výraz pre aplikačný fragment (môže sa vyskytovať viackrát).
<code>fragment^document^{jQuery sel.}</code>	jQuery výraz pre prvok dokumentu v obsahu webstránky.
<code>fragment^documentparent^{jQuery sel.}</code>	jQuery výraz pre rodiča prvku dokumentu, v ktorom sa dokument posúva (nepovinné, vhodné pre stránky v ktorých sa posúva nielen celá stránka, ale aj sledovaný dokument zvlášť v ďalšom prvku).
<code>fragment^documentstop^{class}</code>	Trieda prvkov, na ktorých sa má zastaviť rekurzia fragmentov v dokumente (nepovinné).

Táto otvorenosť rôznym webovým aplikáciám vedie k možnému využitiu adaptívneho proxy servera. Tak ako webovej aplikácii postačuje vytvoriť súbor definujúci jej používateľské rozhranie, aplikačné rozhranie a už len prijímať na vlastných určených adresách spätnú väzbu, tak isto môže toto vykonať proxy server pre ľubovoľné webové aplikácie bez zapojenia ich správcov/autorov. Architektúra celkového riešenia so zapojením proxy servera je načrtnutá na obr. 11. Pri overovaní riešenia sme pre jednoduchosť využili priamo systém ALEF na mieste proxy servera, čo nemá vplyv na použité postupy v metóde.



Obrázok 11: Náhľad na celkový návrh realizácie metódy, v overovanej implementácii webová aplikácia nahrádza proxy server.

6.5 Adaptívna explicitná spätná väzba v systéme ALEF

Na poskytovanie adaptívnych otázok, tipov a pod. (scenár adaptívneho sprievodcu výučbovým kurzom) na základe implicitnej spätnej väzby o práci používateľa alebo na základe iných udalostí bolo potrebné do systému ALEF doplniť funkcionality, ktorú pôvodne neobsahoval.

V spolupráci s ďalším autorom systému ALEF sme implementovali systémové vyhodnocovacie otázky (*EvaluationQuestions*), ktoré je možné pokladať náhodne podľa vopred zadaných pravidiel (*EvaluationQuestionTriggers*) alebo priamym vložением do radu čakajúcich otázok pre konkrétneho používateľa (*EvaluationQuestionQueues*) pri nastaní rôznych udalostí. Navštívená webová stránka systému ALEF si v pravidelných intervaloch (30 sekúnd) žiada zo servera otázky a v prípade, že používateľ má čakajúcu otázku, táto sa mu zobrazí nad prehliadanou stránkou. Používateľ môže zodpovedanie otázky odložiť alebo odpovedanie na otázku odmietnuť.

Použitie vyhodnocovacích otázok sa zaznamenáva – ako samotné ponúknutie otázky používateľovi a spôsob jej vzniku, tak aj odpovede používateľov.

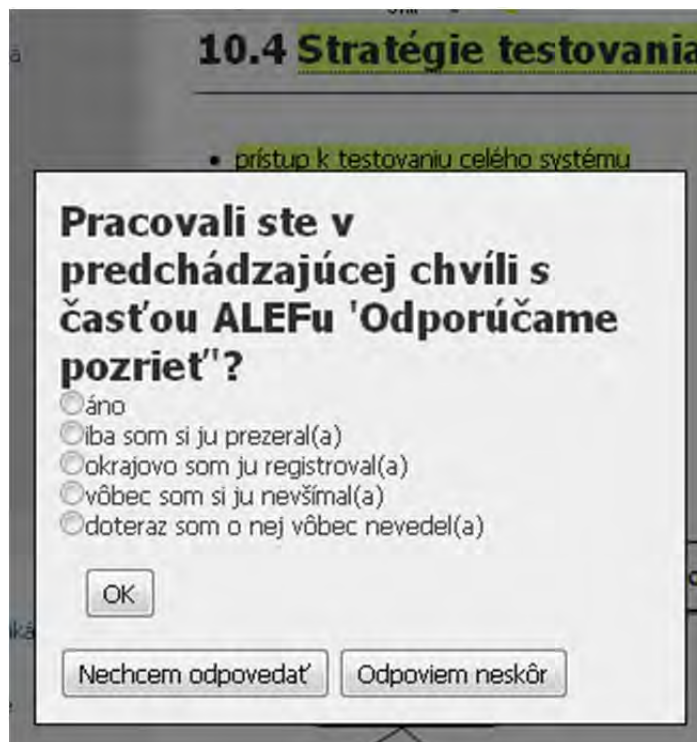
6.5.1 Adaptívna explicitná spätná väzba a rozšírenie UTrack

V prípade, že používateľ zotrvá pohľadom na jednom aplikačnom fragmente príliš dlho, rozšírenie UTrack, v prípade, že webová aplikácia zadefinovala komunikačné rozhranie cez definičný súbor, automaticky odošle navštívenej webovej aplikácii notifikáciu. Táto doba pohľadu je definovaná ako 10 sekúnd, pričom používateľ nemusí pozerat' na fragment presne desať neprerušených sekúnd, ale v prípade pohľadu v blízkosti fragmentu sú z dôvodu presnosti zisťovaného pohľadu pripočítavané čiastkové časy záujmu podľa vzdialenosti pohľadu od fragmentu. V prípade opustenia fragmentu sa doba záujmu nenastaví na nula, ale sa postupne rýchlosťou reálneho času znižuje.

Hranica 10 sekúnd záujmu sa teda prekoná aj napríklad v prípade nasledujúcej činnosti používateľa:

- 8 sekúnd pohľad v rámci fragmentu (aktuálny čas záujmu 8 sekúnd),
- 3 sekundy pohľad mimo fragmentu (aktuálny čas záujmu 5 sekundy),
- 4 sekundy pohľad mimo fragmentu vo vzdialenosti zodpovedajúcej polovici presnosti (aktuálny čas záujmu 5 sekúnd),
- 5 sekúnd pohľad v rámci fragmentu (prekročená časová hranica 10 sekúnd záujmu).

V prípade systému ALEF sa na základe tejto udalosti vyberie a naplánuje otázka súvisiaca s daným fragmentom (obr. 12). Samozrejmosťou je zadefinovanie ochranného časového intervalu (60 sekúnd) od zodpovedania predchádzajúcej otázky, v ktorom sa nová otázka nenaplánuje, aby nebol používateľ neustále vyrušovaný.



Obrázok 12: Príklad adaptívnej otázky pre explicitnú spätnú väzbu. Otázka bola vyvolaná dlhým pohľadom na zásuvný modul odporúčania.

6.6 Vizualizácia významnosti fragmentov

Pre kombináciu informácií o práci viacerých používateľov do jediného vyhodnotenia významnosti fragmentov v dokumente kombinujeme informácie o spätnej väzbe tak, že sčítame všetky ukazovatele (pohľad, zobrazenie dokumentu a pod.) od jednotlivých používateľov na danom fragmente a následne z týchto kombinovaných indikátorov ohodnotíme jednotlivé fragmenty rovnako ako v prípade informácií iba od jediného používateľa.

Používateľ má v dokumentoch, ku ktorým existujú informácie o práci používateľov z rozšírenia, k dispozícii možnosť zvýraznenia dôležitosti fragmentov (obr. 13). Nástroj umožňuje využiť kombinované informácie o pohľade, zobrazení dokumentu a ďalších indikátoroch (prvá ikona, navrhnutá metóda). Pre porovnanie je k dispozícii aj zvýraznenie s využitím pohľadu od tých používateľov, ktorí nepoužívali skutočné sledovanie pohľadu webovou kamerou ale iba jeho simuláciu prostredníctvom myši (druhá ikona) a s využitím iba informácií o zobrazení a posúvaní dokumentov (tretia ikona).



Obrázok 13: Nástroj pre zvýraznenie významnosti fragmentov odvodených z práce používateľov (vľavo hore). Prvá ikona predstavuje našu navrhnutú metódu, ostatné slúžia na porovnanie.

Systém ALEF tiež umožňuje používateľom zvýrazňovať dôležité časti dokumentov, čo predstavuje analógiu zvýrazňovača v tlačенých textoch. Pre porovnanie navrhutej metódy s názormi používateľov na dôležité fragmenty umožňuje nástroj vykonať zvýraznenie aj podľa ich vyznačení. Dôležitosť fragmentu v tomto prípade zodpovedá počtu zvýraznení od všetkých používateľov, ktoré cez neho prechádzajú. Pretože zvýraznenie môže začínať a končiť kdekoľvek na úrovni znakov, aj vo vnútri slov, označenie začínajúce alebo končiacie vo fragmente je považované za čiastočné a k ohodnoteniu fragmentu, napríklad jednej odrážky, prispieva pomerom dĺžky označenej časti k celej dĺžke fragmentu.

7 EXPERIMENTÁLNE OVERENIE ZÍSKAVANIA A VYUŽITIA IMPLICITNEJ SPÄTNEJ VÄZBY

7.1 Návrh experimentu

Pre overenie oboch základných častí riešenia, identifikácie významných fragmentov dokumentu aj sledovanie fragmentov webovej aplikácie, sme navrhli experiment so zapojením študentov predmetu Princípy softvérového inžinierstva vyučovaného na Fakulte informatiky a informačných technológií a s využitím vzdelávacieho systému ALEF používanom v rámci prípravy na cvičenia tohto predmetu. Pretože našim cieľom je umožniť využitie zisťovania pohľadu s bežnými prostriedkami v bežnom prostredí, študentov sme nechali používať systém ALEF rovnakým spôsobom ako zvyčajne, u nich doma. Študenti mali za úlohu sa doma pripravovať sa na vstupné testy konané na jednotlivých cvičeniach a pritom používať rozšírenie prehliadača so sledovaním pohľadu.

Na overenie, či sa používatelia pozerajú tam, kam predpokladáme a zároveň na overenie identifikácie aplikačných fragmentov sme využili adaptívnu explicitnú spätnú väzbu v systéme ALEF a v prípade, že používateľ prekročil hranicu záujmu o aplikačný fragment, v tomto prípade zásuvný modul systému ALEF, ako sú menu, skóre a pod., položili sme mu otázku, či práve v tejto chvíli pracoval s uvedeným fragmentom. Medzi otázky sme zahrnuli aj kontrolné otázky na iné než identifikované moduly.

Na overenie identifikácie fragmentov dokumentu sme počas práce používateľov zbierali spätnú väzbu k fragmentom a vyhodnotili dôležitosť fragmentov získanú z práce používateľov oproti zvýrazneným dôležitým častiam, ktoré študenti označovali anotačným nástrojom v systéme ALEF pre vlastné potreby pri učení, aj na základe výzvy na zvýraznenie dôležitých častí.

7.2 Technické zabezpečenie experimentu

Pre vykonanie experimentu sme vybrali a zakúpili desať webových kamier, dva kusy z piatich modelov rôznej ceny a vlastností, pre pokrytie nehomogénneho hardvéru očakávaného u používateľov v domácich podmienkach. Vybrané kamery sú spolu s vlastnosťami odlišujúcimi ich od ostatných uvedené v tab. 4.

Tabuľka 4: Kamery použité pri overovaní.

Model	Vlastnosti	Cena s DPH (€)
LOGITECH QuickCam Pro 9000	2 mpix, autofocus; nástupca modelu použitého autormi riešenia OpenGazer	78,40 x2
Microsoft LifeCam HD-5000	720p, autofocus, 16:9	29,58 x2
LOGITECH MP Webcam C500	RightLight technology	32,45 x 2
GENIUS Eye 320 SE	manuálne ostrenie	12,96 x2
Microsoft LifeCam VX-700	najlacnejšia	8,51 x2

Ešte pred zapožičaním kamier študentom sme overili presnosť pri použití spolu s knižnicou OpenGazer riadeným experimentom (tab. 5). Na vzorke piatich ľudí sme v rôznych svetelných podmienkach overili presnosť zamerania pohľadu tak, že používatelia mali sledovať pohybujúci sa objekt po obrazovke a zaznamenávali sme priemernú vzdialenosť zisteného pohľadu od polohy objektu. Pretože pre použité algoritmy je vyžadované umiestnenie kamery pod displej, na čo nie sú kamery vždy prispôsobené (predpokladá sa umiestnenie nad displej), vyhodnotili sme aj možnosti správneho umiestnenia kamery bez podkladania ďalšími objektmi a pod.

Tabuľka 5: Presnosť zistenia pohľadu a možnosti umiestnenia kamier pri použití s knižnicou OpenGazer.

Model	Presnosť v osi X (body)	Presnosť v osi Y (body)	Umiestniteľnosť (subj. skóre)
LOGITECH QuickCam Pro 9000	97	66	3
Microsoft LifeCam HD-5000	102	70	2
LOGITECH MP Webcam C500	103	76	2
GENIUS Eye 320 SE	99	70	1
Microsoft LifeCam VX-700	96	56	0

Rozdiely v presnosti boli v závislosti od kvality použitých kamier veľmi malé, pozorovali sme však krátkodobé výpadky pri niektorých modeloch, keď dochádzalo vplyvom snímaného prostredia ku krátkodobým zmenám zaostrenia. Kvalita kamery má tiež vplyv na snímkovaciu frekvenciu v bežných a najmä znížených svetelných podmienkach. Toto je dôležité pri procese kalibrácie, pretože je potrebný určitý počet snímok pohľadu na dane miesto, s klesajúcou snímkovacou frekvenciou pri nižších svetelných podmienkach alebo menej kvalitných kamerách dochádza k predlžovaniu času potrebného na kalibráciu. Tento čas sme namerali v rozsahu od 12 sekúnd až do 35 sekúnd. Overili sme tiež použitie okuliarov – okuliare s antireflexnou úpravou s nízkymi dioptriami nepredstavovali pre sledovanie pohľadu žiadny problém, v prípade vznikajúcich odrazov prekrývajúcich obraz oka však môže dochádzať k problémom s kalibráciou a teda k nemožnosti sledovania pohľadu.

7.3 Realizácia experimentu

Zo študentov predmetu Princípy softvérového inžinierstva sme vybrali 34 študentov, z nich desiatim sme zapožičali uvedené kamery, jeden študent použil vlastnú kameru a okrem toho dvanásť študenti použili vlastnú kameru zabudovanú v notebooku (takáto kamera však nie je umiestnená pod displejom tak, ako algoritmy sledovania očakávajú).

Študentom sme sprístupnili rozšírenie na webovej stránke <http://utrack.labaj.sk> a stručne sme im predviedli prevzatie prenosnej verzie s obsiahnutým rozšírením a proces kalibrácie pohľadu. Počas používania systému v domácom prostredí bez dohľadu nad priebehom experimentu sa nám ozvali len dvaja študenti s problémami ohľadom spustenia rozšírenia a procesu sledovania pohľadu. Jeden študent oznámil nemožnosť spustenia sledovania pohľadu, tento problém bol odstránený nainštalovaním

požadovaných knižníc a druhý študent oznámil sporadicky sa vyskytujúci pád sledovania pohľadu, ktorý sme následne úpravou knižnice OpenGazer odstránili.

Študenti sa následne v čase 16.4.2011 až 2.5.2011 pripravovali na jedno až dve cvičenia (podľa časového harmonogramu rôznych cvičení), používali pritom rozšírenie a odpovedali na zobrazované otázky. Po ukončení tohto obdobia sme ich požiadali o vyplnenie dotazníka ohľadom práce so zapnutým rozšírením, čo vykonalo 28 študentov.

7.4 Vyhodnotenie experimentu

7.4.1 Adaptívne otázky

Celkovo sme počas trvania experimentu vygenerovali 591 otázok na používateľov, či práve pracovali s určitými fragmentmi, z toho 410 otázok vzniklo na základe pohľadu simulovaného z pozície kurzora myši, 115 otázok vzniklo na základe pohľadu odhadnutého z obrazu webovej kamery a 66 otázok bolo kontrolných, keď sme zistili, že používateľ sa pozeral na určitý fragment, ale zámerne sme sa ho spýtali na iný, nesúvisiaci fragment.

Výsledky pre pohľad skutočne zisťovaný našou metódou pomocou webovej kamery sú uvedené v tab. 6. Viac ako polovica zistení bola správna (55,14 % odpovedí „pracoval“ alebo „prezeral“), približne v pätine prípadov (19,63 %) používateľ odpovedal, že identifikovaný modul registroval pohľadom a v ďalšej pätine prípadov (21,50 %) odpovedal, že si modul nevšimol.

Tabuľka 6: Odpovede na otázky ohľadom zamerania pohľadu na aplikačné fragmenty (pohľad zisťovaný webovou kamerou). Položili sme 115 otázok, získali sme 107 odpovedí.

Webová kamera, identifikovaný fragment	Áno	Prezeral som si ju	Registroval som ju	Nevšimol som si ju	Nevedel som o nej	Neodpovedal
Pracovali ste v predchádzajúcej chvíli s časťou ALEFu „XYZ“?	55 (51,40 %)	4 (3,73 %)	21 (19,63 %)	23 (21,50 %)	4 (3,74 %)	8

Ako jedna porovnávacía skupina slúžili používatelia so zisťovaním pohľadu len simulovaným pomocou pozície kurzora myši (tab. 7). V tomto prípade boli zistenia správne („pracoval“ alebo „prezeral“) v 77,16 % prípadov, v 8,7 % prípadov používateľ odpovedal, že daný zásuvný modul registroval a v 14,21 % používateľ nami identifikovaný zásuvný modul nepoužíval.

Ako ďalšia porovnávacía skupina slúžili používatelia, ktorým sme v momente, keď sme zistili, že sa zaoberajú určitým zásuvným modulom, vygenerovali otázku na nesúvisiaci zásuvný modul (tab. 8). V tejto skupine bolo 36,36 % otázok s pozitívnou odpoveďou („pracoval“ alebo „prezeral“), v 15,9 % prípadov používateľ daný zásuvný modul len

registroval a v 36,36 % prípadov bola odpoveď záporná, používateľ si daný modul nevšimol.

Tabuľka 7: Odpovede na otázky ohľadom zamerania pohľadu na aplikačné fragmenty (pohľad simulovaný podľa pozície kurzora myši). Položili sme 410 otázok, získali sme 359 odpovedí.

Myš, identifikovaný fragment	Áno	Prezeral som si ju	Registroval som ju	Nevšimol som si ju	Nevedel som o nej	Neodpovedal
Pracovali ste v predchádzajúcej chvíli s časťou ALEFu „XYZ“?	234 (65,18 %)	43 (11,98 %)	31 (8,7 %)	51 (14,21 %)	-	51

Tabuľka 8: Odpovede na kontrolné otázky (náhodné fragmenty) ohľadom zamerania pohľadu na aplikačné fragmenty. Položili sme 66 otázok, získali sme 44 odpovedí.

Kontrolný fragment	Áno	Prezeral som si ju	Registroval som ju	Nevšimol som si ju	Nevedel som o nej	Neodpovedal
Pracovali ste v predchádzajúcej chvíli s časťou ALEFu „XYZ“?	12 (27,27 %)	4 (9,09 %)	7 (15,91 %)	16 (36,36 %)	5 (11,36 %)	22

V prípade sledovania kurzora myši sme tak dosiahli lepšie výsledky, než pri sledovaní pohľadu. Pravdepodobná príčina spočíva zrejme v princípe práce používateľa so zásuvnými modulmi – viacero používateľov nám v dotazníku potvrdilo naše skôr uvedené predpoklady, že počas pasívneho čítania myšou nepohybujú. Tiež na základe pozorovania prebieha obsluha napríklad menu je bežne vykonávaná tak, že sa používateľ pozrie na dostupné položky, presunie myš na menu, klikne a už sa opäť venuje dokumentu, pričom ponechá na poslednom mieste.

V tom prípade zisťovaním pohľadu pozornosť venovanú danému aplikačnému fragmentu nezistíme, zisťovaním pohľadu simulovaného kurzorom myši však zistíme umiestnenie myši v danom fragmente a keď sa používateľa spýtame, či v predchádzajúcej chvíli s daným fragmentom pracoval odpovie nám kladne, aj už s daným fragmentom nepracuje a venuje sa napríklad dokumentu. Toto by bolo možné odstrániť okamžitým pýtaním sa, teda otázkou, či používateľ pracoval s fragmentom práve v tom momente, keď ho zastihneme otázkou, avšak tento postup nie je v prostredí webu jednoduchý z dôvodu komunikácie princípom PULL (načítaná stránka v prehliadači sa v časovom intervale pýta servera, či má čakajúcu otázku).

Celkovo sme tak zistili porovnaním sledovania pohľadu so jeho simuláciou a s kontrolnou skupinou zistili, že:

- Vykonávaním akcií na základe zisťovania pohľadu na aplikačné fragmenty jednoduchou webovou kamerou vieme v 55 % prípadov vystihnúť moment, keď používateľ priamo pracoval s fragmentom alebo si ho prehliadal a v ďalších 19,63 % prípadov vystihneme aspoň moment, keď používateľ daný aplikačný fragment viac či menej registroval.

- Ak namiesto pohľadu použijeme len pozíciu kurzora, vystihneme v 77,16 % prípadov moment, keď používateľ pred krátkou chvíľou pracoval s daným aplikačným fragmentom.

V oboch prípadoch pod pojmom „moment“ uvažujeme čas, kedy zistíme aspoň desaťsekundový záujem používateľa o aplikačný fragment a akcia, ktorú vykonáme (v prípade overenia zobrazenie otázky, inak napríklad zobrazenie tipu, pomôcky), sa prejaví u používateľa priemerne do 15 sekúnd z dôvodu dopytovania sa načítanej stránky na otázku v intervale 30 sekúnd. Prenosové oneskorenia v rámci stoviek milisekúnd môžeme zanedbať v prípade bežných internetových pripojení potrebných na normálne využívanie webových aplikácií Web 2.0.

7.4.2 Ohodnotenie fragmentov dokumentu

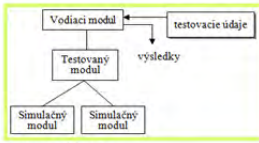
Naším pôvodným zámerom bolo numericky porovnať ohodnotenie dôležitosti fragmentov našou metódou s hodnoteniami vypočítanými prostredníctvom zvýrazňovaní dôležitých častí výučbových objektov študentmi pri učení sa a na základe pokynu. Ukážka vypočítanej a vizualizovanej dôležitosti jednotlivých fragmentov dlhšieho vzdelávacieho objektu je uvedená na obr. 14. Zľava doprava sú zvýraznené dôležité fragmenty: a) na základe našej metódy s využitím pohľadu odhadnutého z obrazu webovej kamery, b) na základe našej metódy s využitím pohľadu od tých používateľov, ktorí nechali pohľad simulovať na základe polohy kurzora myši a c) na základe manuálnych zvýraznení dôležitých častí používateľmi. Červená čiara predstavuje hranicu, po ktorú sa zobrazí obsah dokumentu po načítaní bez posúvania (tzv. *page fold*) na displeji s častým rozlíšením 1280 x 800 v rámci zvyšných častí rozhrania systému ALEFu.

Na obr. 15 je uvedené podobné porovnanie (pohľad zistený na základe webovej kamery oproti pohľadu simulovanému myšou) pre kratší vzdelávací objekt. Obr. 16 obsahuje väčší náhľad toho istého objektu, tentoraz s ručnými zvýrazneniami od používateľov. Intenzita podfarbenia fragmentu predstavuje jeho vypočítanú dôležitosť – čím zelenší je fragment, tým je jeho dôležitosť vyššia.

Môžeme pozorovať, že ručné zvýraznenia používateľmi príliš nesúvisia s pozorovanou prácou používateľov s dokumentmi. Na obr. 14 sú ručnými označeniami zvýraznené ako dôležité fragmenty len prvé odrážky dokumentu sumarizujúce jednotlivé ďalšie časti dokumentu (približne 20 označení na fragment) a zvyšné časti dokumentu sú na základe tejto metriky málo dôležité (približne 3-4 označenia na fragment). Jedným z možných vysvetlení je, že používatelia označovali primárne v časti, ktorá sa im zobrazila pri prvom načítaní dokumentu, pretože veľká väčšina zvýraznení sa nachádza nad hranicou zobrazenia dokumentu bez posúvania. Na obr. 16 sa nachádzajú ručné označenia aj pod touto hranicou, opäť však označujú iba nadpisy, takže ďalšie možné vysvetlenie je, že používatelia pri pokyne na označovanie dôležitých častí vyznačia práve časti určitým spôsobom sumarizujúce dokument alebo určujúce jeho štruktúru a nezvýrazňujú už v rámci jednotlivých podnadpisov na hlbšej úrovni.

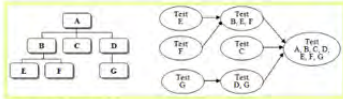
10.4 Stratégie testovania

- prístup k testovaniu celého systému
 - testovanie zdola-nahor
 - testovanie zhora-nadol
 - jednofázové testovanie
 - sendvičové testovanie
 - testovanie porovnávaním



Testovanie zdola-nahor
angl. bottom-up testing

- vhodné, ak veľa modulov "nižšej" úrovne predstavuje všeobecné moduly, ktoré nie často systémovo často používajú



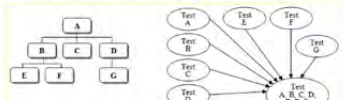
Testovanie zhora-nadol
angl. top-down testing

- problémy so simuláciou modulov na nižších úrovniach
- obmedzená znovupouiteľnosť modulov



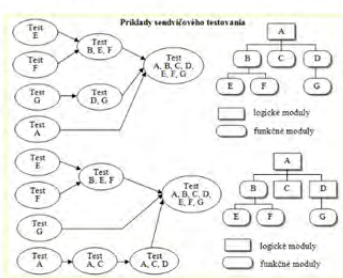
Jednofázové testovanie
angl. big-bang testing

- moduly sa otestujú samostatne a potom sa naraz integrujú
- hodí sa iba pre malé systémy
- náročná identifikácia miesta chyby pri integrácii
- náročné odčinenie chýb v rozhraniach modulov od ostatných chýb



Sendvičové testovanie
angl. sandwich testing

- kombinácia stratégie zhora-nadol a zdola-nahor
- moduly sa rozdeľujú do dvoch skupín:
 - logické: zabezpečujú najmä radenie a rozhodovanie, testujú sa zhora-nadol
 - funkčné: zabezpečujú najmä vykonávanie požadovaných funkcií (operácií), testujú sa zdola-nahor



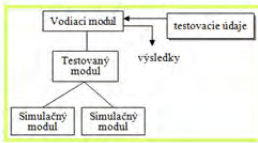
Testovanie porovnávaním
angl. comparison testing, back-to-back testing

- existuje viac verzií systému na testovanie
 - prototyp
 - vývoj vysoko spoľahlivých systémov použitím techniky programovania N verzí
 - vývoj viacerých verzií produktu pre rôzne platformy
- rovnaké výsledky značia, že verzie pravdepodobne pracujú správne
- problémy:
 - rovnaké chyby vo verzách
 - nevychovávajúca špecifikácia



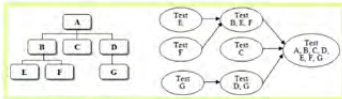
10.4 Stratégie testovania

- prístup k testovaniu celého systému
 - testovanie zdola-nahor
 - testovanie zhora-nadol
 - jednofázové testovanie
 - sendvičové testovanie
 - testovanie porovnávaním



Testovanie zdola-nahor
angl. bottom-up testing

- vhodné, ak veľa modulov "nižšej" úrovne predstavuje všeobecné moduly, ktoré nie často systémovo často používajú



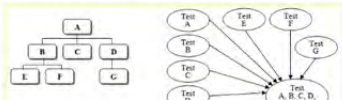
Testovanie zhora-nadol
angl. top-down testing

- problémy so simuláciou modulov na nižších úrovniach
- obmedzená znovupouiteľnosť modulov



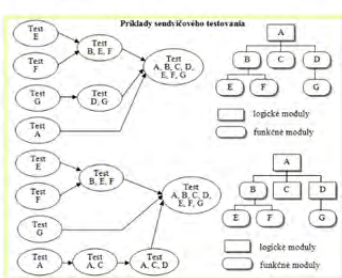
Jednofázové testovanie
angl. big-bang testing

- moduly sa otestujú samostatne a potom sa naraz integrujú
- hodí sa iba pre malé systémy
- náročná identifikácia miesta chyby pri integrácii
- náročné odčinenie chýb v rozhraniach modulov od ostatných chýb



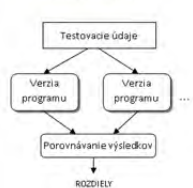
Sendvičové testovanie
angl. sandwich testing

- kombinácia stratégie zhora-nadol a zdola-nahor
- moduly sa rozdeľujú do dvoch skupín:
 - logické: zabezpečujú najmä radenie a rozhodovanie, testujú sa zhora-nadol
 - funkčné: zabezpečujú najmä vykonávanie požadovaných funkcií (operácií), testujú sa zdola-nahor



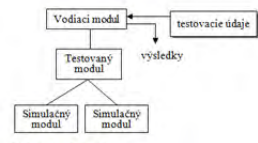
Testovanie porovnávaním
angl. comparison testing, back-to-back testing

- existuje viac verzií systému na testovanie
 - prototyp
 - vývoj vysoko spoľahlivých systémov použitím techniky programovania N verzí
 - vývoj viacerých verzií produktu pre rôzne platformy
- rovnaké výsledky značia, že verzie pravdepodobne pracujú správne
- problémy:
 - rovnaké chyby vo verzách
 - nevychovávajúca špecifikácia



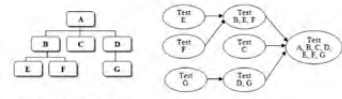
10.4 Stratégie testovania

- prístup k testovaniu celého systému
 - testovanie zdola-nahor
 - testovanie zhora-nadol
 - jednofázové testovanie
 - sendvičové testovanie
 - testovanie porovnávaním



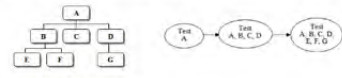
Testovanie zdola-nahor
angl. bottom-up testing

- vhodné, ak veľa modulov "nižšej" úrovne predstavuje všeobecné moduly, ktoré nie často systémovo často používajú



Testovanie zhora-nadol
angl. top-down testing

- problémy so simuláciou modulov na nižších úrovniach
- obmedzená znovupouiteľnosť modulov



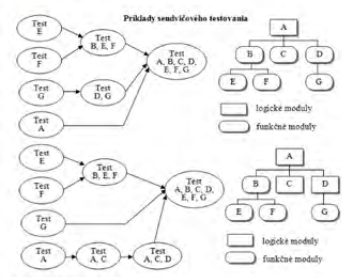
Jednofázové testovanie
angl. big-bang testing

- moduly sa otestujú samostatne a potom sa naraz integrujú
- hodí sa iba pre malé systémy
- náročná identifikácia miesta chyby pri integrácii
- náročné odčinenie chýb v rozhraniach modulov od ostatných chýb



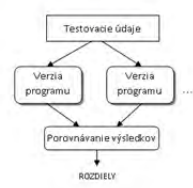
Sendvičové testovanie
angl. sandwich testing

- kombinácia stratégie zhora-nadol a zdola-nahor
- moduly sa rozdeľujú do dvoch skupín:
 - logické: zabezpečujú najmä radenie a rozhodovanie, testujú sa zhora-nadol
 - funkčné: zabezpečujú najmä vykonávanie požadovaných funkcií (operácií), testujú sa zdola-nahor



Testovanie porovnávaním
angl. comparison testing, back-to-back testing

- existuje viac verzií systému na testovanie
 - prototyp
 - vývoj vysoko spoľahlivých systémov použitím techniky programovania N verzí
 - vývoj viacerých verzií produktu pre rôzne platformy
- rovnaké výsledky značia, že verzie pravdepodobne pracujú správne
- problémy:
 - rovnaké chyby vo verzách
 - nevychovávajúca špecifikácia



Obrázok 14: Porovnanie vypočítanej dôležitosti fragmentov (zľava doprava): a) našou metódou s využitím pohľadu sledovaného webovou kamerou, b) využitím pohľadu simulovaného na základe myši, c) podľa zvýraznení používateľmi. Červená čiara približne zobrazuje tzv. page fold.

11.1 Typy zmien počas údržby a používania softvéru

1. Oprava (angl. corrective maintenance)

- odstraňovanie všetkých chýb, ktoré pretrvávajú v systéme



2. Prispôsobenie (angl. adaptive maintenance)

- zmeny softvéru v závislosti od zmien okolitého prostredia (napr. hardvér)
- zmeny časti softvéru v dôsledku modifikácií inej časti

3. Zlepšenie (angl. perfective maintenance)

- vylepšenie produktu (jeho vlastností), ktoré neboli zahrnuté do špecifikácie pôvodného systému
- zmeny v závislosti od zmien problémovej oblasti a požiadaviek používateľa

4. Prevencia (angl. preventive maintenance)

- modifikácie produktu s cieľom zlepšenia ďalšej údržby (čitateľnosť, modifikovateľnosť, spoľahlivosť)

11.1 Typy zmien počas údržby a používania softvéru

1. Oprava (angl. corrective maintenance)

- odstraňovanie všetkých chýb, ktoré pretrvávajú v systéme



2. Prispôsobenie (angl. adaptive maintenance)

- zmeny softvéru v závislosti od zmien okolitého prostredia (napr. hardvér)
- zmeny časti softvéru v dôsledku modifikácií inej časti

3. Zlepšenie (angl. perfective maintenance)

- vylepšenie produktu (jeho vlastností), ktoré neboli zahrnuté do špecifikácie pôvodného systému
- zmeny v závislosti od zmien problémovej oblasti a požiadaviek používateľa

4. Prevencia (angl. preventive maintenance)

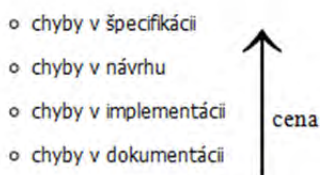
- modifikácie produktu s cieľom zlepšenia ďalšej údržby (čitateľnosť, modifikovateľnosť, spoľahlivosť)

Obrázok 15: Kratší vzdelávací objekt s vyznačenými dôležitými fragmentmi vypočítanými: a) našou metódou s pohľadom sledovaným z obrazu webovej kamery, b) zo simulovaného pohľadu.

11.1 Typy zmien počas údržby a používania softvéru

1. Oprava (angl. corrective maintenance)

- odstraňovanie všetkých chýb, ktoré pretrvávajú v systéme



2. Prispôsobenie (angl. adaptive maintenance)

- zmeny softvéru v závislosti od zmien okolitého prostredia (napr. hardvér)
- zmeny časti softvéru v dôsledku modifikácií inej časti

3. Zlepšenie (angl. perfective maintenance)

- vylepšenie produktu (jeho vlastností), ktoré neboli zahrnuté do špecifikácie pôvodného systému
- zmeny v závislosti od zmien problémovej oblasti a požiadaviek používateľa

4. Prevencia (angl. preventive maintenance)

- modifikácie produktu s cieľom zlepšenia ďalšej údržby (čitateľnosť, modifikovateľnosť, spoľahlivosť)

Obrázok 16: Väčší náhľad vzdelávacieho objektu z obr. 15 a zvýraznenia dôležitých častí od používateľov.

Pri využití pohľadu používateľa a našej metódy môžeme na obr. 14 pozorovať väčšiu významnosť na miestach so zložitejším obsahom a naopak menšiu významnosť na miestach s jednoduchým obsahom. Pretože skúmaný vzdelávací objekt sa zaoberá stratégiami testovania, požiadali sme dobrovoľníka bez konkrétnej znalosti danej témy, avšak so všeobecnými znalosťami v oblasti vývoja softvéru, aby sa v riadenom prostredí pod našim dohľadom naučil obsah tohto vzdelávacieho objektu. Jeho práca s dokumentom zodpovedala pozorovanému ohodnoteniu fragmentov na obrázku – jednoduchšie opísanými a ľahko zapamätateľnými stratégiami sa zaoberal málo, zložitejším venoval viac času.

V prípade využitia pohľadu získavaného z kurzora myši pozorujeme na obr. 14 oveľa rovnomernejšie rozloženie dôležitosti fragmentov. O trochu väčšia dôležitosť je zistená v oblasti sumarizujúcich odrážok a v blízkosti hranice prvotného zobrazenia dokumentu, no ďalej už v dokumente nie je možné pozorovať výraznejšie rozdiely medzi fragmentmi s výnimkou postupného ubúdania dôležitosti smerom ku koncu dokumentu.

U kratšieho vzdelávacieho objektu (obr. 15) nie sú rozdiely medzi metódami a ani medzi fragmentmi v rámci metód také významné. U oboch metód je len veľmi mierne vyššia vypočítaná dôležitosť na nadpisoch a u dominantnej ilustrácie.

7.4.3 Hodnotenie od účastníkov

Odpovede od účastníkov v dotazníku sme vyhodnocovali zvlášť pre používateľov, ktorí využili zapožičanú alebo vlastnú samostatnú kameru, a zvlášť pre používateľov, ktorí využili kameru zabudovanú v notebooku, pretože pri použití kamery mimo očakávané umiestnenie je možné očakávať nižší komfort.

Používateľov sme sa spýtali, či by používali rozšírenie bežne pri používaní webových aplikácií. V prípade používateľov s kamerou zabudovanou v notebooku odpovedalo až 64 %, že by im pravidelné používanie rozšírenia (so sledovaním pohľadu) prekážalo, neprekážalo by len 18 % používateľov. Naopak u používateľov, ktorí použili samostatnú kameru umiestnenú pod displejom uviedlo len 20 %, že by im pravidelné používanie prekážalo a 50 % používateľov by neprekážalo. Medzi dôvodmi odmietavého postoja k pravidelnému používaniu bolo v oboch skupinách uvádzané nepohodlie potrebnej statickej polohy pred kamerou a v skupine používateľov s kamerou z notebooku aj strata kalibrácie. Len jeden používateľ uviedol obavu zo straty súkromia.

Pravidelné používanie rozšírenia so sledovaním pohľadu podľa myši odmietlo 32 % všetkých používateľov, 25 % používateľov by pravidelné používanie neprekážalo. Ako dôvod odmietnutia bola raz uvedená strata súkromia, raz spomalenie prehliadača.

Celkovo účastníci experimentu vo väčšinovom počte uviedli ako odpoveď na otázku na prípadné problémy počas používania stratu kalibrácie a nepohodlie statickej polohy.

Podrobnejší prehľad odpovedí používateľov sa nachádza v prílohe v dokumentácii k experimentu.

8 ZÁVER

V tejto práci sme navrhli metódu pre využitie pohľadu používateľa v kontexte webových aplikácií. V metóde kombinujeme najmä sledovanie zobrazenej časti dokumentu a sledovanie pohľadu, ktorý zisťujeme prostredníctvom bežnej neupravenej webovej kamery v neriadenom prostredí, ale i ďalšie implicitné indikátory záujmu používateľa. Na základe práce používateľa opísanej týmito údajmi následne hodnotíme dôležitosť fragmentov dokumentu, s ktorým používateľ pracuje a tiež vyhodnocujeme spôsob práce používateľa vo webovej aplikácii, v rámci ktorej s daným dokumentom pracuje. Pre tieto odvodené informácie sme navrhli viaceré scenáre využitia.

Navrhnuté metódy sme implementovali vo forme samostatnej aplikácie, ktorá prostredníctvom prevzatej, nami modifikovanej knižnice, zabezpečuje sledovanie pohľadu používateľa a je využiteľná aj v kombinácii s inými aplikáciami. Ďalej sme implementovali rozšírenie prehliadača Firefox, ktoré zabezpečuje sledovanie práce používateľa na webe s využitím zisteného pohľadu a ďalších indikátorov. Scenár zvýrazňovania dôležitých fragmentov ako pomôcka pri učení a adaptívnej explicitnej spätnej väzby/príručky k systému sme implementovali vytvorením prototypu v rámci systému ALEF.

Vytvorené riešenie sme overili neriadeným experimentom so zapojením 34 študentov predmetu Princípy softvérového inžinierstva vyučovanom v druhom ročníku bakalárskeho stupňa štúdia na Fakulte informatiky a informačných technológií STU. V experimente sme preukázali, že sledovaním pohľadu dokážeme úspešne identifikovať, či používateľ venoval v predchádzajúcom momente niektorej súčasť webovej aplikácie vyššiu pozornosť. Ešte vyššiu úspešnosť sme dosiahli pri kontrolnom využití pozície kurzora myši namiesto pohľadu získaného webovou kamerou, čo pravdepodobne vyplýva zo spôsobu použitia konkrétnych súčastí webovej aplikácie, keď používateľ umiestnil na danú časť kurzor kvôli vykonaniu určitej akcie a následne ho tam ponechal pri ďalšej práci.

Pomocou sledovania pohľadu sme dokázali identifikovať náročnejšie a menej náročné časti vzdelávacieho textu, čo sme aj potvrdili riadeným experimentom s používateľom nepoznajúcim oblasť daných vzdelávacích objektov a doménovým expertom. V prípade identifikácie významnosti fragmentov dokumentov už nahradenie sledovania pohľadu pozíciou kurzora myši nepostačovalo a nevedlo k uspokojivým výsledkom.

Počas overenia riešenia sme tiež zistili, že v prípade použitia samostatnej webovej kamery umiestnenej pod displejom by až 50 % používateľov súhlasilo s ďalším používaním rozšírenia prehliadača pri bežnej práci s webovými aplikáciami a ďalších 30 % používateľov zaujalo neutrálny postoj. Naproti tomu pri použití webovej kamery zabudovanej v notebooku, kde obmedzená možnosť manipulácie s kamerou vytvára vyššie nároky na vzdanie sa pohodlia u používateľa, až 64 % používateľov odmietlo bežné používanie rozšírenia. Používatelia často uvádzali nižšie pohodlie a potrebu opakovane vykonať kalibráciu v intervaloch približne desiatok minút, čo je očakávaný efekt daného hardvérového ohraničenia.

Niekoľko používateľov taktiež vyjadrilo obavu zo straty súkromia, čo v prípade sledovania pohľadu vyplýva z toho, že webová kamera je primárne určená na snímanie a nahrávanie alebo odosielanie obrazu, takže používatelia zapnutú kameru nasmerovanú na nich nevnímajú len ako sledovač pohľadu, ale ako nahrávanie ich súkromnej osoby a prostredia. V prípade, že by sa sledovanie pohľadu stalo bežnou súčasťou displejov alebo prenosných osobných počítačov, tento pocit straty súkromia evokovaný zapnutou webovou kamerou by sa určite zmiernil.

Návrh metódy nie je zameraný len na doménu elektronickej výučby a pri implementovaní navrhnutého riešenia bol venovaný dostatok pozornosti tomu, aby bolo výsledné riešenie jednoducho znovupoužiteľné aj pre iné domény a webové aplikácie jednoduchým poskytnutím definície rozhrania samotnou webovou aplikáciou.

Celkovo sa nám podarila preukázať reálnosť a využiteľnosť nasadenia sledovania pohľadu aj u domáceho používateľa, len s minimálnym predchádzajúcim zaškolením.

V ďalšej práci je možné projekt rozvíjať viacerými smermi – vytvoriť ďalšie využitia pre navrhnutú metódu a samotný pohľad, ale dôležité je aj zlepšenie komfortu používateľa pri sledovaní jeho pohľadu.

POUŽITÁ LITERATÚRA

1. ATTERER, R., WNUK, M. & SCHMIDT, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: *Proceedings of the 15th international conference on World Wide Web*, ACM, 2006, p. 203–212.
2. BABCOCK, J.S. & PELZ, J.B.: Building a lightweight eyetracking headgear. In: *Proceedings of the 2004 symposium on Eye tracking research & applications*, ACM, 2004, p. 109–113.
3. BADI, R. et al.: Recognizing user interest and document value from reading and organizing activities in document triage. In: *Proceedings of the 11th international conference on Intelligent User Interfaces - IUI '06*, ACM, 2006, p. 218–225.
4. BARLA, M. & BIELIKOVÁ, M.: Personalizácia "divokého" webu: adaptívny proxy server. In: *Proc. of the 4th Workshop on Intelligent and Knowledge oriented Technologies (WIKT 2009)*, 2009.
5. BIELIKOVÁ, M., ŠIMKO, M., BARLA, M., CHUDÁ, D., MICHLÍK, P., LABAJ, M., MIHÁL, V., UNČÍK, M.: ALEF: Web 2.0 Principles in Learning and Collaboration. In: *e-Learning '10, Proceedings of the International Conference on e-Learning and the Knowledge Society*, 2010, p. 54–59.
6. BRADSKI, G. & KAEHLER, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, Inc., 2008.
7. BRAGANZA, C., MARRIOTT, K., MOULDER, P., WYBROW, M. & DWYER, T.: Scrolling behaviour with single-and multi-column layout. In: *Proceedings of the 18th international conference on World Wide Web*, ACM, 2009, p. 831–840.
8. CHEN, M.C., ANDERSON, J.R. & SOHN, M.H.: What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In: *Conference on Human Factors in Computing Systems*, 2001, p. 281–282.
9. CLAYPOOL, M., LE, P., WASED, M. & BROWN, D.: Implicit interest indicators. In: *Proceedings of the 6th international conference on Intelligent User Interfaces*, ACM, 2001, p. 33–40.
10. COCKBURN, A., FITCHETT, S., GUTWIN, C., GREENBERG, S. & ALEXANDER, J.: Revisiting Read Wear: Analysis, Design, and Evaluation of a Footprints Scrollbar. In: *Word Journal of The International Linguistic Association*, University of Canterbury. Computer Science and Software Engineering, 2009, p. 1665–1674.
11. FOSSO, F. & PORTA, M.: A vision-based attentive user interface with (semi)automatic parameter calibration. In: *Proceedings of the International*

Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing - CompSysTech '09, ACM, 2009, p. IIIA.7-1–IIIA.7-6.

12. FOX, S., KARNAWAT, K., MYDLAND, M., DUMAIS, S. & WHITE, T.: Evaluating implicit measures to improve web search. In: *ACM Transactions on Information Systems (TOIS)*, ACM, 2005, p. 147–168.
13. HANSEN, D. & HANSEN, J.: Eye typing with common cameras. In: *Proceedings of the 2006 symposium on Eye tracking research & applications*, ACM, 2006, p. 55.
14. HANSEN, D. & PECE, A.: Eye tracking in the wild. In: *Computer Vision and Image Understanding*, vol. 98, 2005, no. 2, p. 155–181.
15. HILL, W.C., HOLLAN, J.D., WROBLEWSKI, D. & MCCANDLESS, T.: Edit wear and read wear. In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, ACM, 1992, p. 3–9.
16. HU, N., ZHANG, J. & PAVLOU, P.A.: Overcoming the J-shaped distribution of product reviews. In: *Communications of the ACM*, vol. 52, 2009, no. 10, p. 144–147.
17. ISHAK, E.W. & FEINER, S.K.: Content-aware scrolling. In: *Proceedings of the 19th annual ACM symposium on User Interface Software and Technology - UIST '06*, ACM Press, 2006, p. 155–158.
18. KELLY, D. & BELKIN, N.J.: Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in Information Retrieval*, ACM, 2001, p. 408–409.
19. LABAJ, M., LÍŠKA, P., LOHNICKÝ, M. & ŠVOŇAVA, D.: InFUNmation – Novel Approach to Information Presentation Employing a Game. In: *IIT.SRC 2009: Student Research Conference*, 2009.
20. LONGO, L. & BARRETT, S.: A context-aware approach based on self-organizing maps to study web-users' tendencies from their behaviour. In: *Proceedings of the 1st International Workshop on Context-Aware Middleware and Services affiliated with the 4th International Conference on Communication System Software and Middleware (COMSWARE 2009) - CAMS '09*, ACM, 2009, p. 12–17.
21. REEDER, R.W., PIROLI, P. & CARD, S.K.: Webeyemapper and weblogger: Tools for analyzing eye tracking data collected in web-use studies. In: *CHI'01 extended abstracts on Human factors in computing systems*, ACM, 2001, p. 19–20.
22. SAN AGUSTIN, J., SKOVSGAARD, H., HANSEN, J.P. & HANSEN, D.W.: Low-cost gaze interaction: ready to deliver the promises. In: *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, ACM, 2009, p. 4453–4458.

23. SAN AGUSTIN, J. ET AL.: Evaluation of a low-cost open-source gaze tracker. In: *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ACM, 2010, p. 77–80.
24. ŠIMKO, M., BARLA, M. & BIELIKOVÁ, M.: ALEF: A Framework for Adaptive Web-based Learning 2.0. In: *Key Competencies in the Knowledge Society, WCC 2010*, Springer, 2010, p. 367-378.
25. TARASEWICH, P. & FILLION, S.: Discount Eye Tracking: The Enhanced Restricted Focus Viewer. In: *Proceedings of the Americas Conference on Information Systems AMCIS 2004*, 2004.
26. ULLRICH, C. & MELIS, E.: The Poor Man's Eyetracker Tool of ActiveMath. In: *Proceedings of the World Conference on E-Learning in Corporate Government Healthcare and Higher Education eLearn*, 2002, p. 2313–2316.
27. VELAYATHAN, G. & YAMADA, S.: Behavior-based web page evaluation. In: *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, IEEE Computer Society, 2006, p. 409–412.
28. WROBLEWSKI, D., HILL, W.C. & MCCANDLESS, T.: Attribute-mapped Scroll Bars, U.S. Patent Applications: Serial No. 07/523,117 filed May 14, 1990, Serial No. 07/626,130 filed Dec 11, 1990.
29. XU, S., JIANG, H. & LAU, F.C.M.: Personalized online document, image and video recommendation via commodity eye-tracking. In: *Proceedings of the 2008 ACM conference on Recommender systems*, ACM, 2008, p. 83–90.
30. XU, S., JIANG, HAO & LAU, F.: User-oriented document summarization through vision-based eye-tracking. In: *Proceedings of the 13th international conference on Intelligent User Interfaces*, ACM, 2009, p. 7–16.

Prílohy

- Príloha A – Prihláška projektu do súťaže o štipendium
- Príloha B – Príspevok publikovaný v rámci konferencie 6th E-learning Conference
- Príloha C – Príspevok zaslaný na konferenciu EC-TEL 2011
- Príloha D – Plagát prezentovaný na konferencii IIT.SRC 2011
- Príloha E – Technická dokumentácia
- Príloha F – Dokumentácia k experimentom
- Príloha G – Používateľská príručka systému ALEF
- Príloha H – Používateľská príručka rozšírenia UTrack
- Príloha I – Obsah elektronickej prílohy

PRÍLOHA A – PRIHLÁŠKA PROJEKTU DO SÚŤAŽE O ŠTIPENDIUM

Projekt bol v roku 2009 prihlásený do súťaže o štipendium organizovanej spoločnosťou Softec. Táto príloha obsahuje prihlášku, na základe ktorej sme grant získali.

Prihláška do súťaže o štipendium

Názov projektu:	Odporúčanie a kolaborácia v sociálnom kontexte aplikovaním implicitnej spätnej väzby
Autor projektu:	Bc. Martin Labaj
Fakulta:	Fakulta informatiky a informačných technológií STU
Študijný odbor:	softvérové inžinierstvo
Ročník:	prvý, inžinierske štúdium
e-mail:	martin.labaj@computer.org
Telefón:	+421 902 237 968

Máte posledný ročník uzatvorený?	Áno <input checked="" type="checkbox"/> Nie <input type="checkbox"/>
Študijný priemer za predchádzajúci uzatvorený ročník:	1,19, ukončil Bc. štúdium ako druhý najlepší v kohorte (205 študentov)
Zúčastnili ste sa predchádzajúceho ročníka ŠVOČ (SRC):	Áno <input checked="" type="checkbox"/> Nie <input type="checkbox"/>
Získali ste už nejaké ocenenie na ŠVOČ (SRC)? Aké?	Áno <input checked="" type="checkbox"/> Nie <input type="checkbox"/> Best Paper Award (kategória bc. štúdium)

<p>Popis projektu (max. 1,5 strany A4):</p> <p>Prispôsobovanie prezentovaných informácií v súčasnosti predstavuje významné zlepšenie prístupu k informáciám, či už ide o vzdelávacie materiály, kde sa prispôsobí navigácia v učebných textoch alebo prezentácie samotných materiálov napr. vyznačovaním zaujímavých časti pre študenta a tým sa zlepši efektívnosť výučby, alebo prácu s rozsiahlymi informačnými zdrojmi, kedy prispôsobenie aktuálnym cieľom a potrebám umožní lepší prístup k informáciám. Medzi zaujímavé a dôležité funkcie patrí schopnosť identifikácie významných častí prezentovaných materiálov. Takto možno rozpoznať časti textu, ktoré sú zaujímavé alebo náročné a to využiť pri odporúčaní alebo odporučiť autorovi vykonanie revízie.</p> <p>Bežným spôsobom získavania takýchto informácií je explicitná spätná väzba – od používateľa sa očakáva, že napr. vyplní dotazník, rôznym spôsobom ohodnotí, jednotlivé časti textu a pod. Tieto postupy však so sebou nesú viaceré problémy:</p> <ul style="list-style-type: none"> • narúšajú prirodzený a zaužívaný postup práce používateľa. Používateľ je zvyknutý pracovať (napr. čítať učebné texty) určitým spôsobom a pridaním dodatočnej a navyše z jeho pohľadu nie príliš užitočnej činnosti (vyplňanie prostriedkov pre spätnú odozvu) je vyrušovaný; • pokiaľ používateľ nie je dostatočne motivovaný, nebude takúto odozvu poskytovať vôbec alebo ju bude poskytovať len sporadicky; • nie každý používateľ poskytne pravdivú alebo dostatočne kvalitnú odozvu. <p>Aktuálnymi a z viacerých hľadísk zaujímavejšími sú implicitné prístupy, ktoré spočívajú v sledovaní konania používateľa a získavani informácií na základe jeho aktivít pri práci s informáciami. Napr. pri vyhľadávaní môžeme namiesto explicitného hodnotenia relevancie jednotlivých výsledkov používateľom sledovať, na ktoré výsledky používateľ klikol alebo ešte s nimi naďalej pracoval (tlač, uloženie a pod.). Výsledky vyhľadávania, s ktorými používateľ pracoval, sú často tie, ktoré najviac zodpovedali zámeru používateľa (dopytu pri vyhľadávaní). To, že takto spracúvame jeho akcie používateľa neobťažuje, implicitný prístup netrpí vyššie uvedenými problémami.</p>
--

Projekt sa zaoberá skúmaním možností využitia implicitnej spätnej väzby pri odhaľovaní zaujímavých častí textu a záujmov používateľa formou sledovania, ktorú časť textu používateľ práve prehliada (angl. scrolling) a ako dlho a vloženie tejto informácie do sociálneho kontextu (porovnanie konania viacerých používateľov). Sledovaním, ako dlho je zobrazený každý úsek (obrazovka, okno) textu a vytvorením štatistických prekrytí a prienikov týchto údajov získaných u množstva používateľov by malo byť možné získať informácie o typickej dĺžke čítania až na úrovni jednotlivých riadkov, dokonca pri uvážení rôzneho zalomenia textu u rôznych používateľov až na úrovni jednotlivých slovných spojení. Na základe získaných informácií by malo byť možné odvodiť, či je úsek náročný, zaujímavý alebo iným spôsobom významný.

Tento koncept je zaujímavý práve pre doménu prehliadania materiálov, pretože na úrovni jednotlivých dokumentov alebo webových stránok nie je možné v dostatočnej miere uplatniť iné implicitné metódy vyhodnocujúce úkony používateľa – používateľ počas čítania ucelených textov nekliká na odkazy a často vôbec nepoužíva vstupné zariadenia okrem účelu posúvania textu. Ideálnym prostriedkom implicitnej odozvy by bolo sledovanie zamerania pohľadu používateľa, tieto riešenia však vyžadujú špecializovaný hardvér, ktorý predstavuje ďalšie náklady a požiadavky na vybavenie u používateľa. Do určitej miery je však sledovanie pohľadu alebo tváre používateľa možné vykonať aj bežnou webovou kamerou, ktorá síce neposkytne presnosť na samostatné použitie, ale použitie v projekte spolu so sledovaním zobrazovanej časti textu umožní zvýšenie presnosti, najmä v súvislosti s identifikovaním doby nečinnosti, či inej činnosti používateľa (aj zariadenie menej náročné na kvalitu obrazu nám umožní identifikovať, či je používateľ vôbec prítomný alebo či pracuje s predkladaným textom, na čo postačuje práve aj bežná webová kamera).

Získané informácie o dĺžke čítania jednotlivých fragmentov a následne identifikované náročné a zaujímavé časti priamo vizuálne znázorníme používateľovi ako pomôcku. Plánujeme využiť ich na pokročilejšie odporúčanie – napr. pri učení odporučiť, ktoré časti si má používateľ prejsť ešte raz alebo im venovať zvýšenú pozornosť. Takto môžeme identifikovať, ktoré časti sú nezrozumiteľné a odporučiť autorovi ich revíziu. Pre používateľa môže tiež byť zaujímavá sumarizácia dokumentu, pričom sa zameria na najdôležitejšie časti a používateľ priamo uvidí, ktoré časti ostatní používatelia čítajú najrýchlejšie a môže ich pri čítaní len zbežnejšie prehládnuť.

Informácie o aktuálne zobrazovaných častiach ostatných používateľov môžu byť používateľovi tiež zobrazované priamo v dokumente (texte) v reálnom čase. Pri sprístupnení takto obohatenej sociálnej siete systém podnieti a podporí kolaboráciu. Používateľ tak má možnosť diskutovať aktuálnu časť, na ktorej sa nachádza, s používateľmi, ktorí sa práve zaoberajú rovnakou časťou. Zároveň tým, že títo používatelia aktuálne pracujú na rovnakej časti, sú vyrušovaní menej než v prípade zamýšľania sa nad vzdialenejšou časťou.

Ciele výskumu projektu, ktorého výstupy budú ponúknuté na IIT.SRC 2010:

- Návrh a experimentálne overenie metódy pre automatickú identifikáciu a analýzu významných častí textových, respektíve vizuálnych materiálov pomocou implicitnej spätnej odozvy od množiny používateľov (dĺžka zobrazení rôznych úsekov textu so spresnením pomocou použitia webovej kamery na sledovanie fyzického používateľa alebo podobného princípu).
- Návrh a experimentálne overenie metódy pre odporúčanie náročných alebo zaujímavých častí materiálov a podporu kolaborácie pomocou sociálnej siete na základe identifikovaných významných častí prezentovaných materiálov.

Overenie navrhnutých metód plánujeme v doméne elektronickej výučby – implementácia a experiment v rámci projektu výučby programovania prebiehajúcom na FIIT STU v rámci širšieho systému, na ktorého implementácii sa sčasti podieľa aj autor. Alternatívne v otvorenom informačnom priestore (web) prostredníctvom adaptívneho proxy servera alebo zásuvných modulov do prehliadača.

PRÍLOHA B – PRÍSPEVOK PUBLIKOVANÝ V RÁMCI KONFERENCIE 6TH E-LEARNING CONFERENCE

Táto príloha obsahuje odborný príspevok, ktorý opisuje systém na elektronickú výučbu ALEF a bol publikovaný v rámci konferencie *6th E-learning Conference*.

ALEF: Web 2.0 principles in Learning and Collaboration

Mária Bieliková, Marián Šimko, Michal Barla, Daniela Chudá,
Pavel Michlík, Martin Labaj, Vladimír Mihál, Maroš Unčák

Abstract: *In this paper we present ALEF, a framework for adaptive web-based learning 2.0 that integrates adaptive learning with key concepts of emerging Web 2.0. We outline framework architecture and describe its basic features. We introduce basic modules built on the framework as a part of a prototype implementation for learning lisp programming.*

Key words: *Adaptive Web-based Learning, Framework, Collaboration.*

INTRODUCTION AND RELATED WORK

Web-based educational systems of today continue in addressing the key concepts of Web 2.0. The “2.0” learning environments allow users to create and edit the content, to annotate it and post comments on the top of it and to support social learning (e.g., Moodle [3]). However, current (*non-adaptive*) learning environments do not take into account differences among students and provide “one-size-fits-all” instead of personalized learning experience. This is due to the fact that underlying models of a virtual learning environment often miss explicit and rich metadata descriptions used for inferring by adaptation engines. The complexity of adaptive course authoring is one of the major drawbacks and obstacles when integrating adaptive features into any learning environment [8].

On the other hand, we are not aware that *adaptive* educational systems leverage Web 2.0 advantages to a large extent; especially when integrating and/or combining different approaches to adaptation or when considering social aspects and collaboration. The authors of KnowledgeTree identified the main drawbacks of adaptive web-based educational systems (lack of integration of multiple aspects of learning into one system, poor reuse support) and proposed a distributed architecture which allows a student to reach various interactive learning activities from one single point [1]. However, a drawback of the whole approach is that the activities within a system are still kept separated – the system provides a student with all relevant links pointing to distinct services needs to make a decision whether he or she wants to study the explanatory materials, take some questions and quizzes or practice the acquired knowledge on several exercises.

Related issues were addressed also in [7]. Similarly to our work, authors identified that adaptive educational web-based systems should follow Web 2.0 principles in order to be successful. The authors proposed a framework consisting of five web services which integrate into and extend Moodle LMS by personalization and adaptation features and take advantage of its Web 2.0 presentation layer which provide collaborative parts of the course (activities, blog etc.). The weak point of the solution is that student modeling process as well as whole personalization is separated from collaborative processes – the collaboration within Moodle environment cannot directly influence neither any of the five web services nor their underlying models and thus drive the behavior of the system.

In order to overcome the aforementioned drawbacks, we integrate all available learning and supporting activities into one single framework, which makes them easily accessible from any point of the learning flow. While following the emergence of Web 2.0 features, we develop ALEF, Adaptive LEarning Framework for creating adaptive and interactive web-based learning systems. The conception of ALEF shifts adaptive learning towards Web 2.0 by building on three key principles:

- *Domain modeling* with respect to possibility to automate certain domain model parts creation, and collaborative social aspect and the need to modify or alter domain model by students themselves.
- *Extensible personalization and course adaptation* based on the comprehensive user model, which allows for simultaneous employment of different adaptive techniques to enhance the student's learning experience.
- *Student active participation in learning process* with the ability to collaborate, interact and create content by means of *read-write web* vision. In particular, we exploit different types of annotations as a suitable way to allow for rich interactions on the top of the presented content.

ALEF supports including exercises into the educational course together with their personalized recommendation. Moreover, we developed methods for analyzing submitted solutions of exercises for similarity and style (both texts in formal programming language and texts in natural language, particularly Slovak texts) [2]. It can serve both for improving learning and for plagiarism detection.

In this paper we present an ALEF implementation for learning lisp programming. We focus on description of particular modules – building blocks of adaptive web-based learning 2.0. In particular, we introduce *content recommender*, *sidebar navigator*, *annotation creator* and *collaborative question creator*.

We describe these building blocks by dividing them according to the activities present during an interaction with the learning environment. We recognize two activity flow types: learning and collaborating/creating. Within the learning flow (see Figure 1, solid line), a *personalizer* module plays the key role. Based on a *user model*, it accesses the *domain model* in order to select and personalize educational material to be presented in *presenter*. A *collaborative adaptive content creator* is a basic element of collaborating/creating flow (see Figure 1, dashed line). It enriches presentation from the presenter in order to allow users to assign various types of annotations such as tags or comments to learning objects. The advantage (and the contribution to the state-of-the-art) of ALEF is the possibility to incorporate several personalizers or collaborative adaptive content creators at once, causing educational system to be truly integrated interactive learning environment.

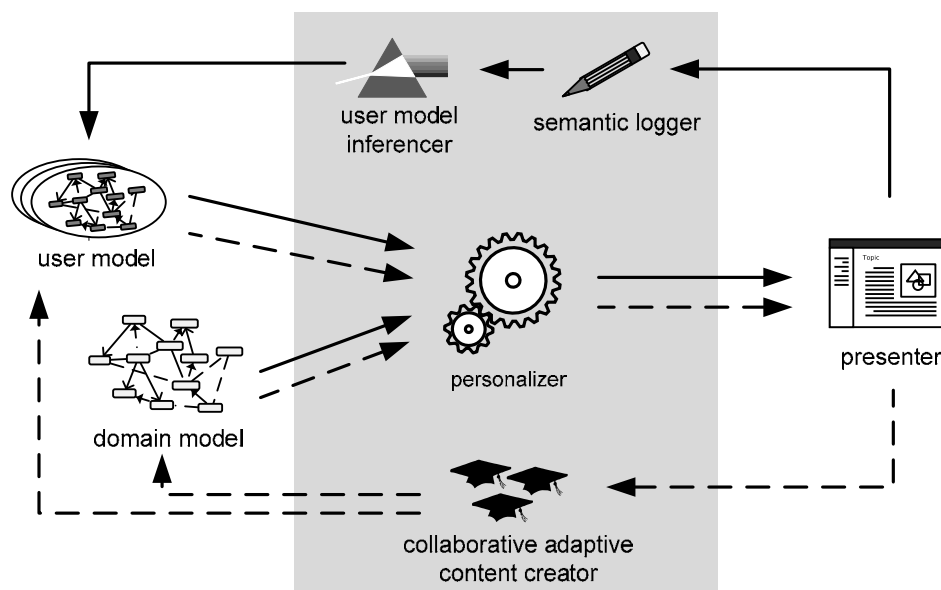


Figure 1. Activity flows within the Adaptive Web-based learning 2.0 environment.

PERSONALIZERS IN E-LEARNING

Content recommender

The objective of adaptive navigation is to help the student to choose the best topics or learning objects to focus on, in order to maximize the learning efficiency. As a key factor we consider the limited time for learning, which might be not sufficient for mastering all required *concepts* (i.e., domain knowledge elements) perfectly. Our strategy is to cover all concepts to some extent rather than to master only a few concepts perfectly.

To achieve this, we estimate potential gain of knowledge at the end of a learning session by considering current learning speed (determined from gain of knowledge from the start of the learning session) and remaining time. Estimated gain of knowledge is divided between the concepts so that the final estimated knowledge levels of concepts respect the importance of a particular concept given by a course author (i.e., a teacher).

To make a recommendation for a student, each learning object is evaluated and assigned a real number as its appropriateness for a student.

Three criteria are used for each learning object evaluation.

- *Concept appropriateness for a student.* A decision whether a student should learn the concepts covered by a particular learning object. It depends on a teacher-given concept importance, a student's knowledge level of the concept and concept.
- *Learning object difficulty appropriateness.* A learning object difficulty should match a student's knowledge level to prevent the student from being uninterested or discouraged. This also maximizes the information value of the student's feedback [6], thus improving student knowledge modeling.
- *Time period since the student's last access the learning object.* Recommending recently viewed learning objects is suppressed.

These criteria are orthogonal and for an ideal recommendation all of them are supposed to be met. Therefore, the final appropriateness of a learning object for the student is defined as the lowest value from the three partial results. After evaluating all learning objects in particular educational course, a predefined number of learning objects with largest appropriateness values are recommended to the student.

Sidebar navigator

Sidebar navigator focuses on tracing student's navigation within the learning object presentation page while learning. Main means of gathering and interpreting the user navigation are based on observation of user actions carried out by keyboard and mouse. Based on collected usage data from different users and taking into the account the social network of a particular student, it personalizes the visualization of certain learning object parts (e.g., by emphasizing mostly visited text). Basic concept of the *sidebar navigator* is similar to the *Read Wear* based on an idea of virtual "wearing-off" of scrollbar [4] – the more time the users spend on a particular portion of a long text (or other visual content), the more will the content become worn-off on this section.

The time spent on learning object portions indicates interesting or problematic textual fragments. Accordingly, the user model is updated by increasing user interest in concepts assigned to those fragments. Portions interpreted as problematic for a student can be planned into learning flow again later, or can be passed to a teacher for a review. By examining the read wear, frequently skipped fragments of the learning object can be visually dimmed or even left out. Fragments within the learning object can be reorganized considering frequent sequences of movement (scrolling). Dimming and reorganizing can significantly optimize students' workflows through learning object.

Data collected by the *sidebar navigator*, especially read wear data, also allow augmentation of social collaboration. Apart from the collaboration through a contribution to the content and respective collaborative adaptive content creators, direct collaboration

based on user communication (i.e., chat) can take advantage of read wear, navigation history and other interest indicators. In this way students have clues which parts took their friends or classmates some time, possibly interesting them or being problematic for them and therefore they are encouraged to communicate and collaborate.

COLLABORATING AND CREATING

Annotations creator

Annotations creator allows for inserting annotations to any resource within a course. Annotations are aimed to obtain fast and targeted feedback from students as well as their contribution to the learning content. We currently focus on comments created by students, which are inserted to user defined places in the learning content.

A student inserts a comment by selecting a portion of text in a learning object (a word or phrase) and writes a comment. The comment is associated with a particular position and text and is displayed within the learning object. Additionally, students can reply to the comment, creating *linked comments*. Many responses to the comment will result in an interactive discussion right 'above' the learning object. Students are also allowed to mark comments as incorrect or inappropriate. This kind of feedback helps the *annotation creator* to filter out confusing or misleading information.

Annotations placed within a learning content in particular positions provide students with another valuable benefit – information with added-value related to the content is collected and available in a convenient place. An example of such information is a link to a website with interesting learning material for further reading.

Moreover, annotations inserted by students can be also considered as a source of implicit feedback. Since every student's comment is associated with a location in the learning object and a small portion of the text, it is possible to associate every comment with a related concept. We can detect interactions between students and concepts that can result into the user model refinements. We have currently identified four kinds of interaction with comments: creation of a comment, marking a comment as incorrect, replying to a comment and deleting one's own comment. We explore the possibilities of user modeling using interactions of students with comments. Our aim is to discover relations between interactions with annotations and student's results.

Collaborative question creator

Collaborative question creator builds on the top of *annotation creator's* mechanisms and allows for creation of specialized annotations holding questions. During the learning, students are encouraged to create testing questions, which are displayed within a page. This way, students themselves become the creators of the educational content.

Collaboratively created questions are usually related to those content parts that are the most important, unclear, or controversial. Adding questions to those parts provides students with further learning possibility and thus supports the overall learning process.

Although the questions are added by students, our goal is to have questions whose quality is near to those provided by domain experts (teachers). The evaluation of question quality is based on the explicit feedback of students in conjunction with actions performed by the students (implicit feedback). It is also dependent on teacher's evaluation. The user explicit feedback is evaluated by taking into account the student model.

The entire method is divided into four processes (not necessary sequenced in the following order): (i) adding a question, (ii) answering a question, (iii) rating a student, (iv) rating a question. The first two processes are short-term (from user's point of view). The third and fourth processes are long-term processes as it takes some time to gather required amount of ratings.

According to these processes a student model holds facts such as a student creates, answer or rates explicitly a question. The rating of a question quality is similar to a rating of student's knowledge. It derives from the explicit rating of questions coming from students and implicit rating, based on the student's actions in the system.

For students' motivation we created simple game. Students gain points, which determine their ranking among their peers. Students should discover a strategy of balance between adding questions and rate questions similarly to others' ratings. We evaluated our approach as a part of the Functional and Logic Programming course last academic year where we considered a reward in form of bonus points included in the overall assessment.

CONCLUSIONS AND FUTURE WORK

We presented basic concepts of Adaptive Web-based Learning 2.0 reflected into ALEF, the framework for adaptive learning. The framework builds upon domain modeling designed to simplify automated course metadata creation [8] with respect to collaborative social aspect of creation. It follows up previous works in this domain [5, 9], while allowing for an extensible personalization and course adaptation based on comprehensive user model and supporting students active participation in the learning process with the ability to collaborate, interact and create content by means of read-write web vision.

Figure 2. Screenshot of ALEF user interface for learning lisp programming (in Slovak). Recommendations coming from *content recommender* are presented either in a separate box (1) above the menu or can be embedded within the main content in the form of interactive examples (3). *Sidebar navigator* visually emphasizes more often read text (2). *Collaboratively created questions* related to current learning object are visualized on-demand in a pop-up widget (4). Displayed content is enriched by adding different types of *annotations*, accessed by hovering the mouse over the underlined sections of text (5).

We used the proposed framework to build an environment for learning lisp programming that combines different learning activities (such as learning from explanatory texts, questions or exercises) along with interactive environment of the Web 2.0 (see Figure 2). Currently we are working on an evaluation of developed modules.

Acknowledgments. This work was supported by the Cultural and Educational Grant Agency of SR, grants No. 028-025STU-4/2010 and 345-032STU-4/2010.

.....

REFERENCES

- [1] Brusilovsky, P. KnowledgeTree: a distributed architecture for adaptive e-learning. WWW 2004, Posters, New York, NY, USA, ACM, New York, NY, pp. 104–113 (2004).
 - [2] Chudá, D., Návrát, P. Support for checking plagiarism in e-learning. In *Procedia - Social and Behavioral Sciences*, Vol. 2, No. 2, Elsevier, pp. 3140–3144 (2010).
 - [3] Dougiamas, M. Taylor, P. Moodle: Using Learning Communities to Create an Open Source Course Management System. In *Proc. of World Conf. on Educational Multimedia, Hypermedia and Telecommunications 2003, AACE*, pp. 171–178 (2003).
 - [4] Hill, W. C., Hollan, J. D., Wroblewski, D., McCandless, T. Edit wear and read wear. In Bauersfeld, P., Bennett, J., Lynch, G. (Eds.) *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems CHI '92*. ACM, NY, pp. 3–9 (1992).
 - [5] Kostelník, R. Bieliková, M. Web-Based Environment using Adapted Sequences of Programming Exercises. In: M. Beneš (Ed.) *ISIM'03, MARQ Ostrava*, pp. 33–40 (2003)
 - [6] Linacre, J., M. Computer-Adaptive Testing: A Methodology Whose Time Has Come. In *MESA Memorandum no. 69* (2000).
 - [7] Meccawy, M., Blanchfield, P., Ashman, H., Brailsford, T., Moore, A.: WHURLE 2.0: Adaptive Learning Meets Web 2.0. In *LNCS 5192, Proc. of the 3rd Int. Conf. on European Conf. on Tech. Enhanced Learning, ECTEL '08*. Springer, pp. 274–279 (2008).
 - [8] Šimko, M., Bieliková, M. Automated Educational Course Metadata Generation Based on Semantics Discovery. In *LNCS 5794, Proc. of 4th European Conf. on Technology Enhanced Learning, EC TEL'09*. Springer, pp. 99–105 (2009).
 - [9] Vozár, O., Bieliková, M. Adaptive Test Question Selection for Web-based Educational System. In: *Proc. of SMAP'08*. CS IEEE Press, pp. 164–169 (2008).
-

ABOUT THE AUTHORS

Prof. Mária Bieliková, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 473, E-mail: bielik@fiit.stuba.sk

Marián Šimko, doctoral student, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 231, E-mail: simko@fiit.stuba.sk

Michal Barla, doctoral student, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 231, E-mail: barla@fiit.stuba.sk

Dr. Daniela Chudá, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 496, E-mail: chuda@fiit.stuba.sk

Pavel Michlík, master degree student, Software Engineering, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail: xmichlik@stuba.sk

Martin Labaj, master degree student, Software Engineering, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail: xlabajm@stuba.sk

Vladimír Mihál, master degree student, Software Engineering, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail: xmihalv@stuba.sk

Maroš Unčík, bachelor degree student, Informatics, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail: xuncik @stuba.sk

PRÍLOHA C – PRÍSPEVOK ZASLANÝ NA KONFERENCIU EC-TEL 2011

Táto príloha obsahuje odborný príspevok, ktorý opisuje aplikáciu sledovania pohľadu používateľa v doméne webových aplikácií a to v bežných podmienkach domáceho používateľa. Príspevok bol zaslaný na konferenciu *EC-TEL 2011* (Springer LNCS), ktorá sa bude konať v septembri 2011 v Palerme (Taliansko). V čase písania diplomovej práce bol príspevok posudzovaný programovým výborom konferencie.

Employing Gaze Tracking as an Implicit Interest Indicator in Web-Based Learning for Common Settings

Martin Labaj and Mária Bieliková

Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3, 842 47 Bratislava, Slovakia
martin.labaj@computer.org, bielik@acm.org

Abstract. In the educational web-based systems implicit feedback measures are often overlooked, possibly because the setting of user browsing the application in the limited environment of web browser. In our research, we aim to bring these technologies into common use with educational applications and as such we use commodity computer equipment. In this paper we present a method for fragment identification based on implicit feedback with emphasis on read wear and other common implicit interest indicators. Since these indicators are time-sensitive, we combine them with gaze tracking in order to detect user inactivity and to increase positional precision. We present realization of such method with common webcams and in the environment of web-based application. Several scenarios for use of this method in the field of web-based learning and open web in general are provided and discussed.

Keywords: gaze tracking, implicit feedback, web-based learning

1 Introduction and Related Work

Identification of important or interesting parts of content is useful in educational systems. It has use in many scenarios common: recommendation, summarization, revision support, etc. Apart from methods based on content itself, user feedback is commonly used. This however presents a challenge, both in explicit and implicit user feedback. With explicit feedback (e.g. document rating or pop-up survey), a user has to cooperate and he may be distracted from his work, which is undesirable when we expect him to be learning effectively.

Also, it has been shown that users tend to provide explicit feedback, e.g. rate an item, only when they are very satisfied or dissatisfied. Users with opinion in between may not see enough motivation to respond [1]. Refining explicit feedback to its fragments may be problematic, so implicit feedback means are often preferred.

In the case of implicit feedback (e.g. click count, time spent on document), the user is left working undistracted and only his actions are observed passively. This removes the need of cooperation in discovering user characteristics and while we may not always exactly interpret user intentions, we are not dependent on getting the user to answer explicit feedback, which may be difficult. Many implicit interest identifiers were described and combined [2], both scoped to entire documents (click count,

printing, aggregations like total count of certain operations, etc.) and scoped to document's fragments (mouse pointer movement, gaze, copy operations, etc.).

There are recognized problems with tracking of users viewing content – a user can pursue other activities while content is displayed, he can even leave completely. Content taking longer to read (learn) is example of such situations – user navigates to a document and then reads it for some time, while he makes no additional implicitly traceable actions. This time periods with no input can ambiguously mean that the user is reading or he left the computer for a break. However, presence and gaze of the user correlates with mouse movement to certain levels [3], but mouse movement is often ignored in educational system.

We address such issues by using implicit interest indicators with emphasis on read wear [4] and mouse and to correctly detect inactive time periods and further improve accuracy, we employ gaze tracking using common webcams. For the purpose of physically tracking the user gaze, there are several gaze tracking projects that rely on low-cost hardware. Projects we analyzed and experimented with are listed in Table 1. As some examples in research, they were successfully used as an input method by disabled people and also for document summarization [5] and online content recommendation.

Table 1. Several available gaze tracking projects with their properties.

Project	Platform	License	Hardware	Image capture	Data access
ITU GazeTracker	Emgu CV; Windows XP SP2	GPLv3 & commercial	nightvision DV cam, IR webcam	<i>remote:</i> zoomed on eye <i>near:</i> head/stand mount	UDP/TCP API, library, source code
OpenGazer, (NetGazer)	OpenCV; Linux, Mac, (Windows)	GPLv2 (GPL)	webcam	<i>remote:</i> under display	library, source code
openEyes	Matlab/Open CV; multi- platform	GPLv2	webcam/IR webcam	<i>near:</i> head mount	source code
myEye	OpenCV; Windows XP SP2	CC BY-NC- ND 2.5, no source	IR webcam	<i>near:</i> head/stand mount	not known
FTR_BY_ WLP	undisclosed	CC BY-NC 2.0, no source	webcam with LED	<i>remote:</i> under display	no

Many of these solutions work on infrared image, which requires both infrared camera (less common webcams or common webcams with removed infrared filter or DV camera with night vision) and infrared lighting (original feature of camera or separate IR LED device). Another difference is the required position of camera – some algorithms require clear image of eye in the entire picture which means that camera have to be close to face. For this purpose, cameras are commonly mounted to caps, glasses or table arm stands, even piece of wood that user bites into was used. As such, it is either easy to get out of camera picture or camera moves with head and with no tracking of position in front of display, user must keep his head still.

One exception is OpenGazer, which sacrifices some precision, but works both without infrared light and with picture of user from absolutely common webcam from standard head-to-monitor distance. As we aim at users working in the web-based educational system, possibly at home, we do not want them to construct still low-cost, yet custom, hardware setups with mounts and stands or to be limited to users with uncommon infrared webcams, and therefore we built upon OpenGazer, respectively its direct Windows port NetGazer.

Our aim is to enhance existing adaptive web-based systems by implicit feedback on everyday settings, which can markedly enhance and also change our possibilities for adaptation of the content and navigation within educational course and improve learning efficiency. In this paper we present a method for fragment identification using implicit interest indicators (most importantly read wear and gaze tracking) and its design which reflect implementation with common webcams and widely used web browser. As a platform for application of proposed method and for its evaluation we use instances of ALEF, Adaptive Learning Framework [6], which we actively use in bachelor Informatics study in several courses of programming (C, Lisp, Prolog) and software engineering principles.

2 Collecting the Feedback in the Web Environment

In order to track users working with web-based document, several approaches exist. Own implementation of document browser supporting subset of HTML/CSS [7] provides absolute overview of users' actions; however it is very difficult to implement such browser in the wake of Web 2.0, which extensively uses JavaScript, etc. Proxy server may be used to track user behavior [8] by modifying responses and inserting information. Conventional browsers are then used for user interface, page rendering, etc. Browser plugin or extension runs with local privileges both in respect to browser and system and may be used to track actions outside web document itself [2].

As we track user's gaze, we have to be able to run code accessing webcam and computing gaze location. We therefore chose to base our realization on two approaches combined – a Firefox browser extension, which traces user's actions within browser and also contains a gazer tracking application which communicates gaze location to it; and a plugin for the Adaptive Proxy server [8], which we use on several projects.

Adaptive Proxy is an enhanced http proxy server which we already actively use for tracking users' movement across web – it works by monitoring and altering both requests and responses passing through it and as such, it models users, detects virtual communities, augment existing web-based application with new features and more. In combination with gaze tracking browser extension, we can incorporate even more information about user's actions within web applications. With the help of proxy, not only we can use our proposed methods in any web application in the open web, but we can also provide information about collected fragments and user actions to several methods currently used in the proxy server.

Overview of feedback collection architecture is shown in the Figure 1.

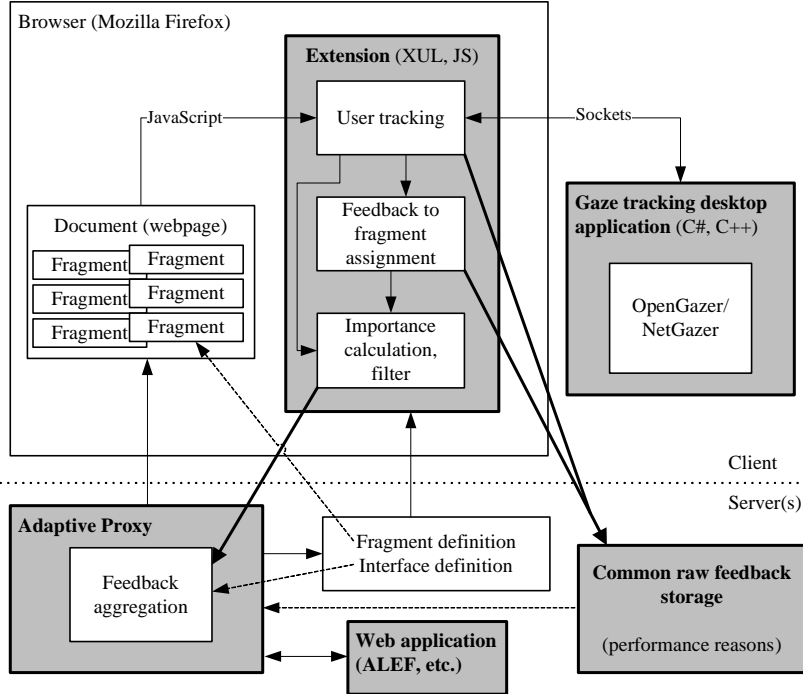


Fig. 1. Web-based feedback collection architecture and technologies used.

3 Employing Collected Feedback for Fragments Identification

We consider interest indicators not only in the relation to the document, but also within the application itself, therefore we divided fragments into *document fragments* and *application fragments*. Document fragments are more extensive and have several nested levels: documents, sections, etc. Application fragments are elements of the web application itself (e.g. in ALEF: menu, recommender widget, navigation widget).

We divided interest indicators into several groups shown in Table 2, which also shows weights we set for groups of indicators and weights and measures for indicators in those groups. The weights were estimated based on our observations and preliminary experiments.

We combine interest indicators collected for fragment as follows

$$AI_f = \frac{\sum_i (w_{G_i} \sum_j w_j v_{f_j})}{|f|}$$

where AI is Attention Index – attention that a user paid to the fragment f .

Identified fragments are then to be used in several scenarios ranging from simple visualization of most interesting fragments to summarization of the document by displaying only most important fragments – which allows users to quickly scan through learning material on the first visit or when repeating the already learned.

Using information about application fragments is to be used to construct adaptive guide to educational application, where different tips and explicit questions are provided to users according to whether they noticed some parts of the system with their gaze and did not work with them afterwards or they did not look at them at all.

Table 2. Examples of interest indicator weights and measures.

Untargeted ($w_G = 0.2$)		
	w	v_f [measure]
Mouse cursor movement	0.5	distance in pixels
Mouse clicks	1.0	count
Passively targeted ($w_G = 2.0$)		
	w	v_f [measure]
Content being displayed (read wear)	1.0	time in seconds
Gaze	4.0	time in seconds
Actively targeted ($w_G = 1.0$)		
	w	v_f [measure]
Text selection	1.0	count & total length
Text copying	2.0	count & total length
Specific actions (annotations, etc.)	4.0	count
Link clicks	1.0	count
Search results within a page	1.0	count
...		
Document-related ($w_G = 0.2$)		
	w	v_f [measure]
Document prints	1.0	count
Number of visits	1.0	count
...		

4 Evaluation and Conclusions

We first separately evaluated the possibility of even performing the gaze tracking with cheap webcam alone. With several common webcams ranging from least to most expensive ones with high definition resolution and with different users, we assessed properties of OpenGazer/NetGazer. It is resistant against lighting conditions (halogen, wolfram, daylight and light from display alone) – it is enough to have only eyes lit sufficiently. However, calibration time (related to frame rate) increased from average 12 seconds to 35 seconds, as frame rate drops in poor light conditions. We found this time acceptable, however longer times in low light may be problematic when recalibration requiring looking on several calibration targets is needed. Accuracy is of course lower than with expensive professional kits, but still achieves acceptable 100 and 70 pixels width-to-height (ratio of accuracy is mimicking aspect ratio of display), which is lower than dimensions of high-level elements of Web 2.0 applications (menus, widgets, document parts, etc.). Interestingly, there were no significant differences in average accuracy between expensive and less expensive webcams; however calibration times (as frame rates) differ under different light conditions.

To evaluate possibility to employ proposed methods in common settings, we gathered 34 bachelor-grade students in the course Software engineering principles, provided them with several diverse webcams and showed them simple instructions as where to download the extension from, how to place the camera and how to go through calibration process. Afterwards, they were instructed to use ALEF educational system at home as usual when preparing for tests. With all students

successfully accessing ALEF with our extension and 15 students providing their gaze tracked, with only one student reporting problems that resolved as prerequisites issue, we proved that gaze tracking is applicable for home settings with standard hardware and brief instructions.

To evaluate whether users are actually looking to areas our system assumes they are looking, we set up explicit questions in ALEF, which are invoked, when a user stares on some application fragment for some time (10 seconds threshold implemented as counter decreased by timer (gaze fading) and increased by gaze within the bounding area). Users were then asked whether they were in fact working with parts of system we identified. Their responses are recorded in Table 3.

Table 3. Answers whether users worked with the fragment we identified based on their gaze.

	Yes, working	Only looking	Partly looking	Not even looked	Did not respond	Total
Have you been just working with part X of ALEF system?	40	4	19	23	6	92

Our proposed method for fragment identification is based on gaze tracking. We aimed at its implementation with common webcams in widely used web browser in order to bring new possibilities to educational systems in common settings. Main prospective contribution is more accurately knowing users' work in web-based applications, particularly in ALEF, where evaluation is being carried out.

As the final effect, we aid students in learning, repeating the learned knowledge and encourage user communication. With our proposal these concepts are also easily brought to open web via the Adaptive Proxy and enhance adaptive web-based applications with the use of cheap stock hardware available in everyday settings.

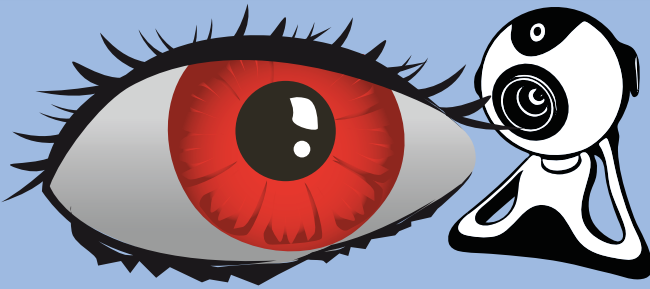
References

1. Hu, N., Zhang, J., Pavlou, P.: Overcoming the J-shaped distribution of product reviews. In: Comm. of the ACM, pp. 144-147. ACM (2009)
2. Fox, S., Karnawat, K., Mydland, M., Dumais, S., White, T.: Evaluating implicit measures to improve web search. In: ACM Trans. on Information Systems (TOIS), pp. 147-168. (2005)
3. Chen, M.C., Anderson, J.R., Sohn, M.H.: What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In: Proc. of Conf. on Human Factors in Computing Systems, pp. 281-282. (2001)
4. Hill, W.C., Hollan, J.D., Wroblewski, D., Mccandless, T.: Edit wear and read wear. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, pp. 3-9. ACM (1992)
5. Xu, S., Jiang, H., Lau, F. C. M.: User-oriented document summarization through vision-based eye-tracking. In: Proc. of the Int. Conf. on Intelligent UIs, pp. 7-16. ACM (2009)
6. Šimko, M., Barla, M., Bieliková, M.: ALEF: A Framework for Adaptive Web-Based Learning 2.0. In: Proc. of IFIP Advances in Inf. and Comm. Tech., pp. 367-378. Springer (2010)
7. Braganza, C., Marriott, K., Moulder, P., Wybrow, M., Dwyer, T.: Scrolling behaviour with single-and multi-column layout. In: Proc. of the 18th Int. Conf. on WWW, pp. 831-840. ACM (2009)
8. Barla, M., Bieliková, M.: Ordinary Web Pages as a Source for Metadata Acquisition for Open Corpus User Modeling. In: Proc. of WWW/Internet 2010, pp. 227-233. IADIS (2010)

**PRÍLOHA D – PLAGÁT PREZENTOVANÝ NA KONFERENCII
IIT.SRC 2011**

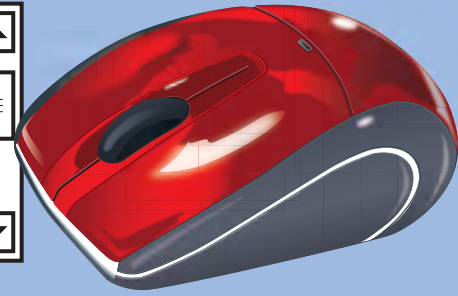
RECOMMENDATION AND COLLABORATION BASED ON IMPLICIT FEEDBACK

MARTIN.LABAJ@COMPUTER.ORG
PROF. MÁRIA BIELIKOVÁ



Aims

- identify important fragments in web app
- employ gaze in common settings (simple webcam)



Untargeted ($w_G = 0.2$)	w	v_r [measure]
Mouse cursor movement	0.5	distance in pixels
Mouse clicks	1.0	count
Passively targeted ($w_G = 2.0$)	w	v_r [measure]
Content being displayed (read wear)	1.0	time in seconds
Gaze	4.0	time in seconds
Actively targeted ($w_G = 1.0$)	w	v_r [measure]
Text selection	1.0	count & total length
Text copying	2.0	count & total length
Specific actions (annotations, etc.)	4.0	count
Link clicks	1.0	count
Search results within a page	1.0	count
Document-related ($w_G = 0.2$)	w	v_r [measure]
Document prints	1.0	count
Number of visits	1.0	count

Application fragments (A)

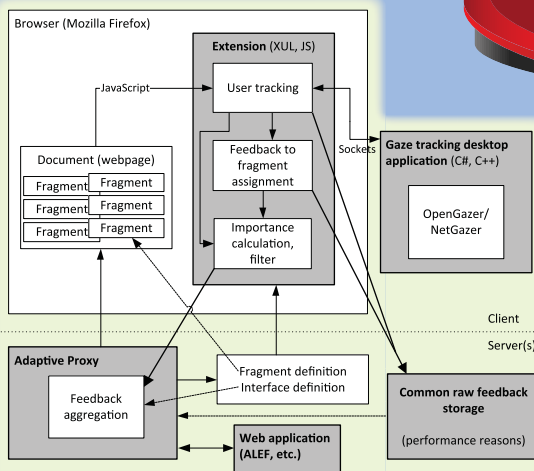
Document fragments (B)

$$AI_j = \frac{\sum (w_G \sum w_r v_r)}{|I|}$$

Visualization & summarization - highlight/select fragments according to their Attention Index

Adaptive guide to educational system (C) - provide tips & ask questions according to user's gaze

Collaboration - augment messaging with users' actions



Evaluation

Model	Specs	Price incl. VAT (€)
LOGITECH QuickCam Pro 9000 VID	2 mpix, autofocus, previous model	78,40 x2
	QuickCam 4000 was used with OpenGazer	

Microsoft LifeCam HD-5000 USB	720p, autofocus, 16:9	29,58 x2
LOGITECH MP Webcam C500	RightLight technology	32,45 x2
GENIUS Eye 320 SE	640x480, manual focus	12,96 x2
Microsoft LifeCam VX-700	(cheapest)	8,51 x2

Model	Positioning	X accuracy	Y accuracy
LOGITECH QuickCam Pro 9000 VID	+++	97 px	66 px
Microsoft LifeCam HD-5000 USB	++	102 px	70 px
LOGITECH MP Webcam C500	++	103 px	76 px
GENIUS Eye 320 SE	+	99 px	70 px
Microsoft LifeCam VX-700	-	96 px	56 px

Mouse	Yes, working	Only looking	Partly looking	Not even looked	Did not respond	Total asked
Have you been just working with part X of ALEF system?	52.9 %	17.4 %	8.7 %	11.6 %	9.4 %	115

Gaze	Yes, working	Only looking	Partly looking	Not even looked	Did not respond	Total asked
Have you been just working with part X of ALEF system?	43.5 %	4.3 %	20.7 %	25 %	6.5 %	92

ALEF experiment

- 34 selected students of PSI course (10 borrowed cameras)

of which:

- 33 students used the extension

of which:

- 17 students used webcam gaze tracking
- 23 students used mouse-gaze simulation

Gaze tracking

- "safe" accuracy 200 px

- calibration 12 to 35 seconds (light cond.)
- glasses - antireflection matters

Project	Platform	License	Hardware	Image capture	Data access
ITU GazeTracker	Emgu CV; Windows XP SP2	GPLv3 & commercial	nightvision DV cam, IR webcam	remote: zoomed on eye near head/stand mount	UDP/TCP API, library, source code
OpenGazer, (NetGazer)	OpenCV; Linux, Mac, (Windows)	GPLv2 (GPL)	webcam	remote: under display	library, source code
openEyes	Matlab/Open CV; multi-platform	GPLv2	webcam/IR webcam	near: head mount	source code
myEye	OpenCV; Windows XP SP2	CC BY-NC-ND 2.5, no source	IR webcam	near: head/stand mount	not known
FTR_BY_WLP	undisclosed	CC BY-NC 2.0, no source	webcam with LED	remote: under display	no

	Gaze from webcam	Simulated gaze
Asked questions	110	421
Answered	103	385
Asked-to-answered ratio	93.64 %	91.45 %

PRÍLOHA E – TECHNICKÁ DOKUMENTÁCIA

1 Komunikácia s aplikáciou GazerTracker

Aplikácia GazerTracker načúva na lokálnom porte 56453 a umožňuje pripojenie neobmedzeného množstva klientov prostredníctvom socketov pre preberanie informácií o pohľade. Aplikácia prijíma príkazy vo formáte uvedenom v tab. 1 a reaguje na ne správami vo formáte uvedenom v tab. 2. Na príkazy *EXIT* a *STOP* aplikácia neodpovedá a ukončí spojenie.

Tabuľka 1: Správy prijímané aplikáciou GazerTracker od pripojených klientov.

Správa	Význam
NAME {name}\n	Identifikácia klienta pre zobrazenie v zozname klientov.
SUBSCRIBE\n	Prihlásenie sa na odber udalostí týkajúcich sa pohľadu – klient bude automaticky dostávať stavové informácie (správa g,...).
UNSUBSCRIBE\n	Odhlásenie sa z odberu udalostí pohľadu.
STATUS\n	Požiadanie o aktuálny stav pohľadu (správa g,...).
VERSION\n	Požiadanie o vnútornú verziu aplikácie.
SHOWWINDOW\n	Požiadanie o zobrazenie konfiguračného okna.
EXIT\n	Ukončenie komunikácie zo strany klienta.
STOP\n	Ukončenie komunikácie zo strany klienta a vypnutie aplikácie GazerTracker.

Tabuľka 2: Správy odosielané aplikáciou GazerTracker pripojeným klientom.

Správa	Význam
GazerTracker v0.99\n	Úvodná správa aplikácie po pripojení klienta, obsahuje vonkajšiu verziu aplikácie.
HELLO {name}\n	Privítanie klienta pri prvom pripojení a odpoveď na príkaz <i>NAME</i> . Klient, ktorý ešte nezaslal svoje meno prostredníctvom príkazu <i>NAME</i> je oslovený poradovým číslom.
SUBSCRIBED\n	Klient bol úspešne prihlásený na automatické zasielanie stavu pohľadu.
UNSUBSCRIBED\n	Klient bol úspešne odhlásený z odberu stavu pohľadu.
g, {x}, {y}, {cstat}, {type}\n	Stav pohľadu používateľa: <i>x, y</i> súradnice pohľadu na obrazovke, <i>cstat</i> stav kalibrácie (<i>1</i> = v poriadku/ <i>0</i> = kalibrácia je neplatná), <i>type</i> typ zdroja pohľadu (<i>gaze</i> = webová kamera/ <i>mouse</i> = simulácia z myši).
{a}. {b}. {c}. {d}\n	Vnútorná verzia aplikácie GazerTracker, <i>a, b, c, d</i> sú čísla.
SHOWN\n	Konfiguračné okno bolo úspešne zobrazené.
NOTUNDERSTOOD\n	Neplatný príkaz.

2 Definičný súbor systému ALEF pre rozšírenie UTrack

Systém ALEF poskytuje na adrese http://adresa.servera/utrack_def.txt nasledujúci súbor definujúci používateľské a aplikačné rozhranie systému pre rozšírenie UTrack.

Tabuľka 3: Definičný súbor `utrack_def.txt` poskytovaný systémom ALEF.

Hodnota	Význam
REALUTRACKDEF!	Predpísaná hlavička súboru.
userdef^#user-identification	Identifikácia používateľa je uložená v elemente s identifikátorom „ <i>user-identification</i> “.
documentdef^learning_objects\[([0-9]+).*	Identifikácia dokumentu je reťazec číslíc nasledujúci za reťazcom „ <i>learning_objects</i> “ v URL adrese dokumentu
fragmenturl^utrack/fragments	Spätnú väzbu opisujúcu fragmenty a aktivitu používateľa v záložkách posielat' v predpísanom formáte na adresu „ http://adresa.servera/utrack/fragments “
overgazeurl^utrack/overgaze	Okamžitú spätnú väzbu vznikajúcu pri nadmernom záujme o aplikačný fragment posielat' v predpísanom formáte na adresu „ http://adresa.servera/utrack/overgaze “
fragment^recommender^#recommender	Aplikačný fragment „ <i>recommender</i> “ predstavuje element s identifikátorom „ <i>recommender</i> “.
fragment^menu^#menu	Aplikačný fragment „ <i>menu</i> “ predstavuje element s identifikátorom „ <i>menu</i> “.
...	...
fragment^rating^#userRatingWidget	Aplikačný fragment „ <i>rating</i> “ predstavuje element s identifikátorom „ <i>userRatingWidget</i> “.
fragment^document^.learning_object	Sledovaný dokument predstavuje element s triedou „ <i>learning_object</i> “.
fragment^documentparent^#main_content_container	Sledovaný dokument sa posúva v rodičovskom prvku s identifikátorom „ <i>main_content_container</i> “.
fragment^documentstop^selectionParent	Rekurziu fragmentov dokumentu zastaviť na prvkoch s triedou „ <i>selectionParent</i> “ (obaľovač textových uzlov pre používateľské anotácie v systéme ALEF).

3 Komunikácia spätnej väzby webovej aplikácii

Rozšírenie UTrack v momente zmeny lokality v záložke, zatvorenia záložky alebo celého prehliadača odošle webovej aplikácii, ktorej dokument bol v záložke otvorený spätnú väzbu na webovou aplikáciou špecifikovanú adresu (*fragmenturl*) POST požiadavkou s parametrami opísanými v tab. 4.

Tabuľka 4: Parametre požiadavky posielanej rozšírením webovej aplikácii pri opustení stránky.

Parameter	Význam
userid	Identifikácia používateľa prevzatá spôsobom definovaným webovou aplikáciou (<i>userdef</i>) z dokumentu.
docid	Identifikácia dokumentu prevzatá spôsobom definovaným webovou aplikáciou (<i>documentdef</i>) z URL adresy.
location	Navštívená adresa.
totaltime	Celkový čas otvorenia dokumentu v záložke (sekundy).
activetime	Čas, po ktorý bol používateľ na záložke aktívny (sekundy).
gazed	Informácia, či bol použitý pohľad: <i>gaze</i> pohľad zistený z webovej kamery, <i>mouse</i> pohľad simulovaný z polohy myši, <i>none</i> bez pohľadu.
ext_version	Verzia rozšírenia prehliadača.
track_version	Verzia aplikácie GazerTracker pripojenej k rozšíreniu prehliadača.
docfrag	Spätná väzba pre fragmenty dokumentu vo formáte JSON.
appfrag	Spätná väzba pre fragmenty aplikácie vo formáte JSON.

Spätná väzba pre dokumentové fragmenty má nasledujúci formát zoznamu objektov pre jednotlivé fragmenty:

```
[{"gaze": sekundy pohľadu,  
  "wear": sekundy zobrazenia,  
  "other": súčet ostatných indikátorov,  
  "lengthX": šírka v pixeloch,  
  "lengthY": výška v pixeloch},  
  
  {ďalší fragment v poradí},  
  
  ...]
```

Spätná väzba pre aplikačné fragmenty má obdobný vnútorný formát (*gaze*, *wear*, *other*, *lengthX*, *lengthY*), ale spätná väzba k jednotlivým fragmentom nie je uložená v prvkoch zoznamu, ale vo forme pomenovaných prvkov objektu.

```
{„menu“:{hodnoty fragmentu},  
  
  „rating“:{hodnoty fragmentu},  
  
  ...}
```


4 Spracovanie prijatých príkazov (aplikácia GazerTracker, C#)

```
1. private string processNetCommand(string command, int clientNumber)
2. {
3.     if (command == null) return null;
4.
5.     String com = command.Trim('\0');
6.     com = com.Trim('\n');
7.     com = com.Trim('\r');
8.     if (com == "") return null;
9.
10.    char[] delimiters = { ' ', ',', '.', ':', '\t' };
11.    string[] words = com.Split(delimiters);
12.
13.    String response = "NOTUNDERSTOOD\n";
14.
15.    if (words[0] == "NAME" && words.Length == 2)
16.    {
17.        clientNames[clientNumber] = words[1];
18.        updateClientList();
19.        response = "HELLO " + clientNames[clientNumber] + "\n";
20.    }
21.    else if (words[0] == "SUBSCRIBE" && words.Length == 1)
22.    {
23.        clientSubscribers[clientNumber] = true;
24.        updateClientList();
25.        response = "SUBSCRIBED\n";
26.    }
27.    else if (words[0] == "UNSUBSCRIBE" && words.Length == 1)
28.    {
29.        clientSubscribers[clientNumber] = false;
30.        updateClientList();
31.        response = "UNSUBSCRIBED\n";
32.    }
33.    else if (words[0] == "STATUS" && words.Length == 1)
34.    {
35.        response = messageStatus();
36.    }
37.    else if (words[0] == "SHOWWINDOW" && words.Length == 1)
38.    {
39.        updateShowWindow();
40.        response = "SHOWN\n";
41.    }
42.    else if (words[0] == "VERSION" && words.Length == 1)
43.    {
44.        response = Application.ProductVersion+"\n";
45.    }
46.    else if (words[0] == "EXIT" && words.Length == 1)
47.    {
48.        ((Socket)clientSockets[clientNumber]).Close();
49.        clientSockets[clientNumber] = null;
50.        updateClientList();
51.        response = null;
52.    }
53.    else if (words[0] == "STOP" && words.Length == 1)
54.    {
55.        notifyIconTray.Visible = false;
56.        Application.Exit();
57.    }
58.
59.    return response;
60. }
```

5 Sledovanie nadmerného záujmu o aplikačný fragment (rozšírenie UTrack, JavaScript)

```
1. for (var name1 in tracked) {
2.   if (name1 == "document") continue;
3.   var obj = $jUT(tracked[name1], tab.contentDocument);
4.   if (obj == null || obj.length == 0) continue;
5.
6.   var xleft = obj.offset().left;
7.   var xright = xleft + obj.width();
8.   var ytop = obj.offset().top;
9.   var ybottom = ytop + obj.height();
10.
11.  var gaze = 0.0;
12.
13.  if (!obj.is(':visible') || obj.width() == 0 || obj.height() == 0) {
14.    gaze = 0.0
15.  } else if (TrackingHandlers.is_in_box(e_x,e_y,xleft,ytop,xright,ybottom)) {
16.    gaze = 1.0;
17.  } else {
18.    var dist = TrackingHandlers.distance_from_box(e_x, e_y, xleft, ytop, xright,
19.                                                  ybottom);
20.
21.    if (dist >= constants.gaze_assignment_precision /
22.        constants.gaze_assignment_appf_ratio) {
23.      gaze = 0.0;
24.    } else {
25.      gaze = 1 - (dist / (constants.gaze_assignment_precision /
26.                          constants.gaze_assignment_appf_ratio));
27.    }
28.  }
29.
30.  overgazen[name1] += gaze * time_diff * 2;
31.
32.  if (fbAppFragments[name1] == undefined) {
33.    fbAppFragments[name1] = {
34.      gaze : 0.0,
35.      wear : 0.0,
36.      other: 0.0,
37.      lengthX : 0,
38.      lengthY : 0
39.    };
40.    fbAppFragments[name1].lengthX = obj.width();
41.    fbAppFragments[name1].lengthY = obj.height();
42.  }
43.
44.  fbAppFragments[name1].gaze += gaze * time_diff / 1000.0;
45. }
46.
47. // DevelSupport.Log(JSON.stringify(fbAppFragments));
48. // DevelSupport.Log(JSON.stringify(fbFragments));
49.
50. for (var name2 in overgazen) {
51.   if (overgazen[name2] >= constants.time_to_be_app_considered_overgazen) {
52.     overgazen[name2] = 0;
53.
54.     FeedbackCommunicator.sendOvergazen(tab, name2);
55.   }
56. }
```

6 Rekurzívne prehliadanie fragmentov pri zvýrazňovaní (systém ALEF, JavaScript)

Nasledujúci kód rekurzívne zvýrazní fragmenty zadaného objektu podľa zoznamu dôležitosti.

```
1. highlightRecursive:function(obj, importance_list) {
2.   if (typeof obj == "undefined" || obj.size() === 0) {
3.     return;
4.   }
5.
6.   obj.each(function() {
7.     var child = $j(this);
8.     var child_children = child.children();
9.
10.    if (child_children.size() == 0) {
11.      EvaluatorAlef.highlight(child, importance_list);
12.    } else if (child.hasClass("selectionParent")) {
13.      // we hit stopper fragment
14.      EvaluatorAlef.highlight(child, importance_list);
15.    } else {
16.      // go deeper
17.      EvaluatorAlef.highlightRecursive(child_children, importance_list);
18.    }
19.  });
20. },
21.
22. highlight:function(obj, importance_list) {
23.   if (!obj.is(':visible') || obj.width() == 0 || obj.height() == 0) {
24.     // hidden
25.     return;
26.   }
27.
28.   var children = obj.children();
29.   children.each(function() {
30.     $j(this).removeClass('HighlightAnnotation');
31.   });
32.
33.   var imp = importance_list[evalA_licounter++];
34.
35.   if (imp == undefined) {
36.     if (!evalA_firstalert) {
37.       alert("Error in data! Importance list shorter than actual document.");
38.       evalA_firstalert = true;
39.     }
40.     imp = 0;
41.   }
42.
43.   if (obj.is("img")) {
44.     obj.css('outline-style', 'solid');
45.     obj.css('outline-color', 'rgba(217,254,119,' + imp.toString() + ')');
46.     obj.css('outline-width', '5px');
47.   } else {
48.     obj.css('background-color', 'rgba(217,254,119,' + imp.toString() + ')');
49.   }
50. },
```

7 Parametrizácia vyhodnocovacích otázok (systém ALEF, Ruby)

Uvedený kód umožňuje parametrizáciu textu vyhodnocovacích otázok tagmi a náhodných udalostí ponúknutia otázky podmienkami. V kóde je využitý príkaz `eval`, avšak jeho vstupy nie sú modifikovateľné používateľmi a sú predspracované.

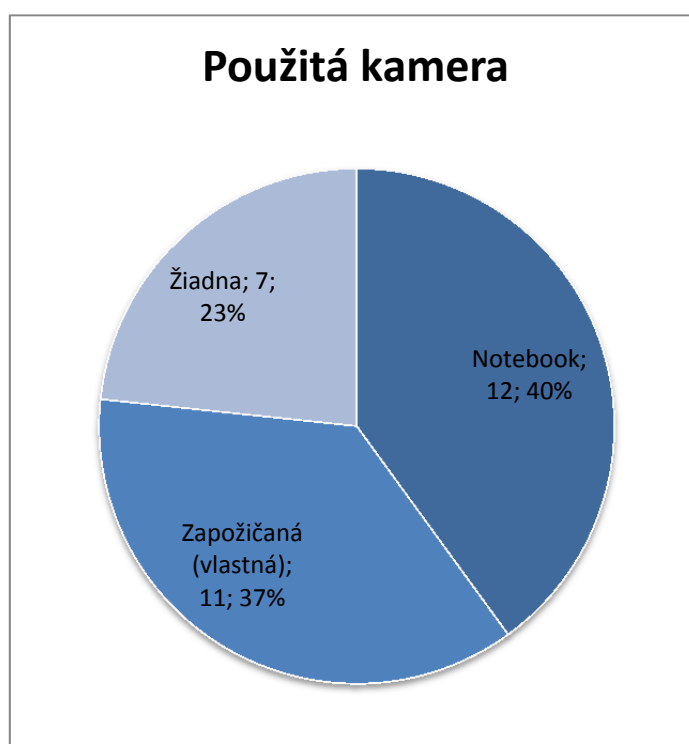
```
1. # replaces tags in questions, if resolving of single tag fails, it is replaced
2. # with replacement string
3. # example: "xyz%USERNAME%abc", ["answer1", "person %USERNAME%", "answer4"]
4. def replace_tags(str, replacement = "")
5.   if str.is_a?(String)
6.     matches = str.scan(/\%([A-Z]*?)\%/)
7.
8.     matches.each do |m|
9.       str.gsub!(/\%#{m[0]}\%/, resolve_tag(m[0], replacement))
10.    end
11.   elsif str.is_a?(Array)
12.     str.collect do |x|
13.       replace_tags(x)
14.     end
15.   end
16.
17.   return str
18. end
19.
20. def resolve_tag(str, replacement = "")
21.   tag = str.downcase
22.   begin
23.     return eval("tag_#{tag}")
24.   rescue Exception => exc
25.     return replacement
26.   end
27. end
28.
29. # evaluates whether question trigger condition string is true
30. # both, %TAGS% and ^CONDITIONS^ can be used
31. # example: "%USERNAME% = 'Janko Hrasko' || ^RAND^ < 0.5"
32. # "^SOMEFUNCTION^(%TAGARGUMENT%, ^FUNCTIONARGUMENT^)"
33. def eval_conds(str)
34.   condstr = replace_tags(str, "nil")
35.   condstr = replace_conds(condstr)
36.
37.   begin
38.     return eval(condstr)
39.   rescue Exception => exc
40.     return false
41.   end
42. end
43.
44. def replace_conds(str)
45.   matches = str.scan(/\^[A-Z]*?\^/)
46.
47.   matches.each do |m|
48.     str.gsub!(/\^#{m[0]}\^/, "cond_#{m[0].downcase}")
49.   end
50.
51.   return str
52. end
```


PRÍLOHA F – DOKUMENTÁCIA K EXPERIMENTOM

1 Dotazník predložený študentom

Študentov zúčastnených v experimente sme po ukončení experimentu požiadali o dobrovoľné, ale motivované vyplnenie nasledujúceho dotazníka. Dotazník vyplnilo 28 z 34 študentov zúčastnených v experimente. Úplné odpovede na textové otázky sú dostupné v elektronickej prílohe práce.

1. Aký model kamery ste použili?



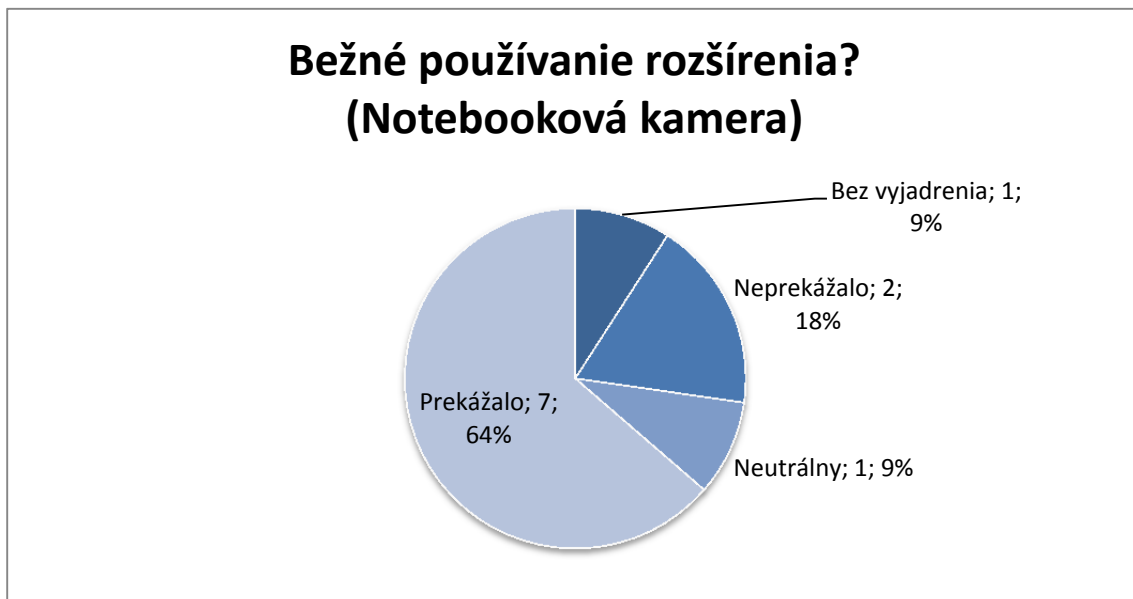
Obrázok 1: Použitie kamier používateľmi.

Na obr. 1 sú uvedené odpovede študentov na použitú kameru – teda či použili samostatnú kameru (vlastnú alebo nami zapožičanú) alebo kameru na notebooku, ktorú nie je vždy možné správne nasmerovať a umiestniť. Dvaja študenti použili aj zapožičanú kameru, aj kameru na notebooku, preto sa nachádzajú v dvoch skupinách.

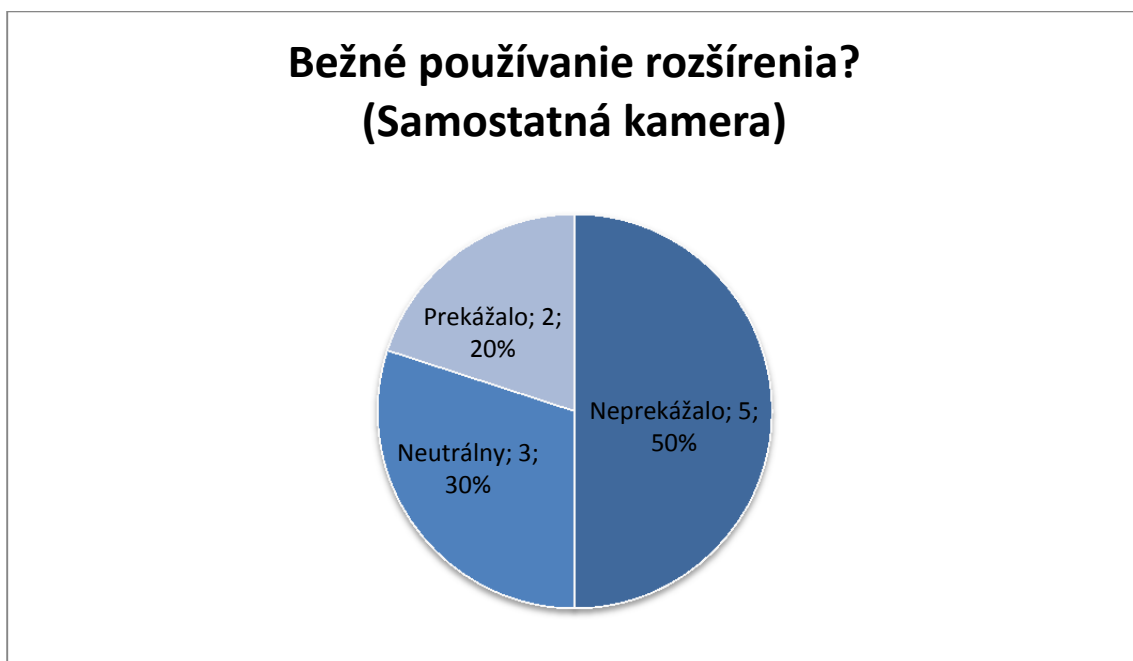
2. Kam ste umiestnili kameru?

Pretože experiment prebiehal v neriadenom prostredí, touto otázkou sme zisťovali, či používatelia správne vykonali pokyny pre umiestnenie kamery. Používatelia, ktorí uviedli použitie dedikovanej kamery (nie zabudovanej v notebooku) správne uviedli „pod displej“, používatelia kamier zabudovaných v notebookoch uviedli „nad displej“.

3. Pri sledovaní pohľadu kamerou - predstavovalo pre vás použitie rozšírenia činnosť, ktorú by ste boli ochotní akceptovať pri bežnom používaní webových aplikácií?



Obrázok 2: Boli by ste ochotní akceptovať používanie rozšírenia (so sledovaním pohľadu) pri bežnom používaní webových aplikácií? Odpovede študentov, ktorí použili kameru, ktorá bola súčasťou notebooku.



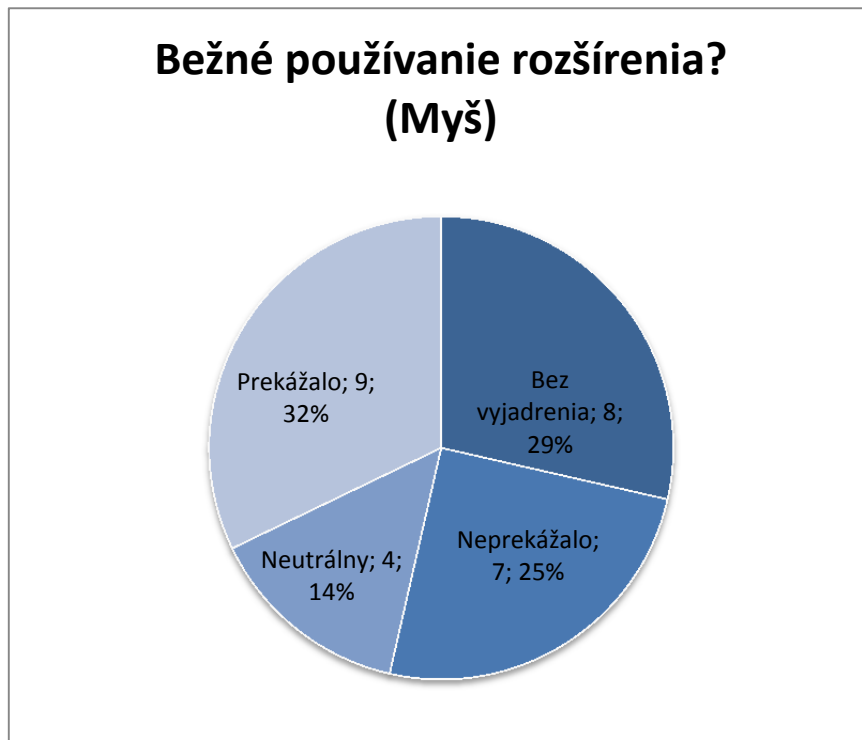
Obrázok 3: Boli by ste ochotní akceptovať používanie rozšírenia (so sledovaním pohľadu) pri bežnom používaní webových aplikácií? Odpovede študentov, ktorí použili samostatnú kameru.

Na obr. 2 sú uvedené odpovede študentov, ktorí používali kameru zabudovanú v notebooku, na obr. 3 sú uvedené odpovede študentov, ktorí používali samostatnú kameru.

3a. Ak ste v predchádzajúcej otázke vybrali možnosť "Prekážalo by mi to", prečo?

- Používatelia kamier zabudovaných v notebooku
 - Nepohodlie (5x), strata kalibrácie (4x)
 - Súkromie (1x)
- Používatelia samostatných kamier
 - Nepohodlie (2x)
 - Problémy s kamerou v operačnom systéme (1x)

4. Pri sledovaní pohľadu myšou - predstavovalo pre vás použitie rozšírenia činnosť, ktorú by ste boli ochotní akceptovať pri bežnom používaní webových aplikácií?



Obrázok 4: Boli by ste ochotní akceptovať používanie rozšírenia (so sledovaním pohľadu pomocou pozície myši) pri bežnom používaní webových aplikácií? Odpovede všetkých študentov.

4a. Ak ste v predchádzajúcej otázke vybrali možnosť "Prekážalo by mi to", prečo?

- Strata súkromia, obava zo zneužitia na reklamné účely (1x)
- Spomalenie prehliadača, načítavania stránok (1x)

5. Vyskytli sa počas používania problémy?

- Straty kalibrácie, pohodlie statickej polohy (11x)
- Chyba aplikácie (4x; po e-mailovej konzultácii opravená chyba v programe)
- Neintuitívne ručné ukončovanie sledovacej aplikácie GazerTracker nezávisle od rozšírenia UTrack (3x; pridaná funkcia automatického ukončenia)
- Zníženie výkonu prehliadača (3x; čiastočne súvisí s dočasnými výkonovými problémami servera systému ALEF, nezávislými od použitia rozšírenia)
- Sledovanie pohľadu nefungovalo dobre s okuliarmi kvôli odleskom (1x)

2 Umiestňovanie použitých webových kamier

V tab. 1 je uvedený prehľad webových kamier použitých pri experimentoch. Ako najlepšie umiestniteľná sa ukázala kamera Logitech QuickCam Pro 9000, ktorú bolo možné s pomocou dvojkľbového stojana nielen jednoducho nastaviť správne na tvár používateľa z pozície pod displejom podľa požiadaviek použitých algoritmov, ale zároveň bolo možné upraviť jej výšku tak, aby nebola zatienená v prípade použitia vyšších modelov klávesníc nachádzajúcich sa medzi používateľom a klávesnicou.

V prípade použitia notebooku, kde sa kamera musí vojsť do tesného priestoru medzi klávesnicou a displejom, sa naopak ako výhodnejšie javili menšie modely kamier (Genius Eye 320 SE a Microsoft LifeCam VX-700), ktoré pri stolných počítačoch nedosahovali spoza klávesníc.

Tabuľka 1: Webové kamery použité pri overovaní riešenia.

		
LOGITECH QuickCam Pro 9000	Microsoft LifeCam HD-5000	LOGITECH MP Webcam C500
		
GENIUS Eye 320 SE	Microsoft LifeCam VX-700	

PRÍLOHA G – POUŽÍVATEĽSKÁ PRÍRUČKA SYSTÉMU ALEF

Na vytvorení používateľskej príručky systému ALEF sa spoločne podieľali viacerí autori systému.

**Slovenská Technická Univerzita
Fakulta informatiky a informačných technológií**

Používateľská príručka systému ALEF

**Autori projektu: Mária Bieliková, Marián Šimko, Michal Barla,
Pavel Michlík, Martin Labaj, Vladimír Mihál,
Maroš Unčík, Jakub Ševcech, Máté Fejes**



2011

Obsah

1 PRÍSTUP DO SYSTÉMU ALEF	1
1.1 Prihlásenie do systému	1
2 PRÁCA SO SYSTÉMOM	2
2.1 Navigačná časť	2
2.2 Obsahová časť	4
2.2.1 Odpovedanie na otázku	4
2.2.2 Odpovedanie na cvičenie	6
2.2.3 Výučbový text	7
2.2.4 Poznámky priradené ku kľúčovým slovám v texte	7
2.3 Anotačný pásik	9
2.4 Zvýraznenie dôležitých fragmentov	10
2.5 Časť so zásuvnými modulmi	12
2.6 Pridávanie poznámok	12
2.7 Komponent pre zobrazovanie skóre	13
2.8 Komponent pre zobrazenie nahlásených chýb	13
2.9 Komponent pre prácu s externými zdrojmi	14
2.10 Adaptívne systémové otázky	15

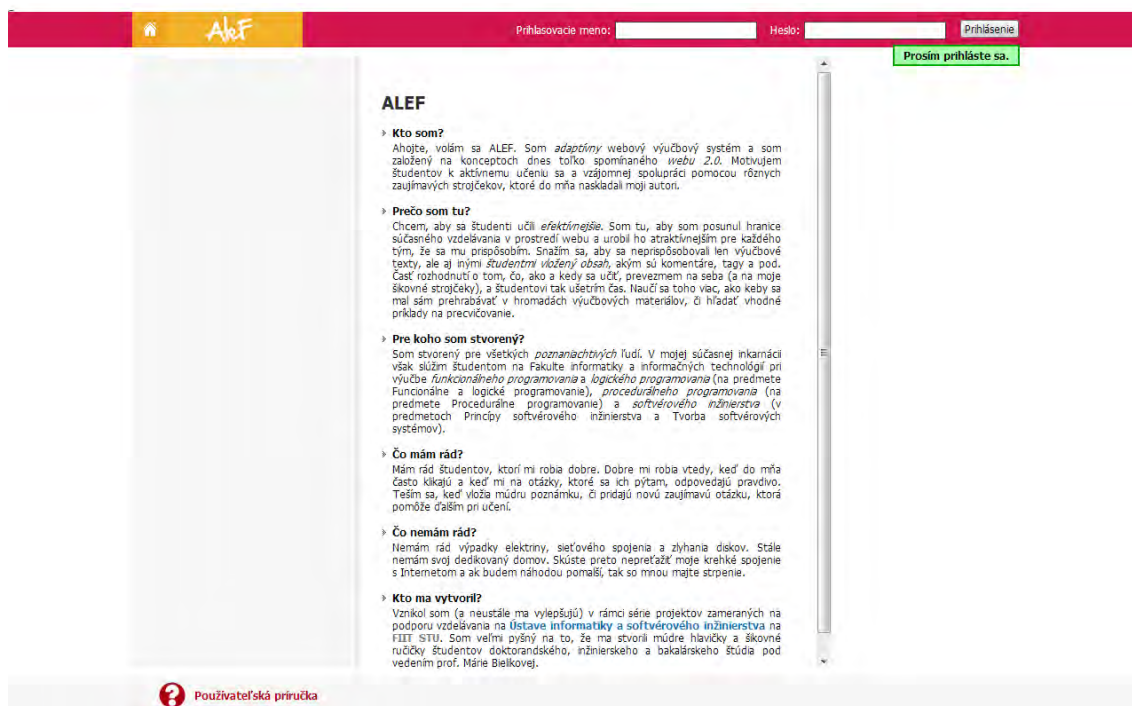
1 Prístup do systému ALEF

System ALEF (Adaptive Learning Framework) je adaptívny výučbový systém na podporu výučby na predmetoch Funkcionálne a logické programovanie a Procedurálne programovanie na Fakulte informatiky a informačných technológií na Slovenskej Technickej Univerzite v Bratislave. Aktuálnu ukážkovú verzia tohto systému možno nájsť na webovej adrese

<http://alef.fiit.stuba.sk/>

1.1 Prihlásenie do systému

Po návšteve webovej adresy výučbového systému ALEF sa zobrazí úvodná obrazovka, kde sa vo vrchnej časti nachádza prihlasovací formulár (Obr. 1). V prípade, že máte vytvorené konto v systéme ALEF, zadajte svoje prihlasovacie meno a heslo.



Obr. 1. Úvodná obrazovka a prihlasovací formulár.

2 Práca so systémom

Po prihlásení do systému sa nezávisle od role používateľa zobrazí úvodná obrazovka systému (Obr. 2). Ak nie je pri komponente uvedené inak, zobrazuje sa používateľom oboch rolí. Úvodná obrazovka je rozdelená na štyri časti:

1. navigačná časť
 - zobrazuje odporúčanie pre študentov
 - zobrazuje menu pre navigáciu
2. obsahová časť
 - zobrazuje aktuálne zvolený výučbový materiál
3. anotačný pásik znázorňujúci poznámky a nahlásené chyby
4. časť so zásuvnými modulmi
 - zobrazuje zásuvné moduly v systéme

The screenshot displays the AleF system interface. At the top, there is a navigation bar with the AleF logo and menu items: Administrácia, Lisp, Prolog, and C. The user is identified as Vladimír Mihal (študent) with an 'Odhlásiť' button. The main content area is divided into several sections:

- Podobné témy**: A sidebar on the left showing a list of topics. Item 1, 'Bílé znaky', is highlighted with a yellow circle labeled '1'.
- Bílé znaky**: The main content area displays the title 'Bílé znaky' and a text snippet. A yellow circle labeled '2' is placed over the text.
- Anotačný pásik**: A horizontal bar below the main content showing a note from Pavel Herout. A yellow circle labeled '3' is placed over this bar.
- Právnosti**: A sidebar on the right showing 'Tvoje skóre' (51.9), 'Nahlásené chyby', 'Externé zdroje', and 'Tagy'. A yellow circle labeled '4' is placed over the 'Tagy' section.

At the bottom left, there is a 'Používateľská príručka' button.

Obr. 2. Úvodná obrazovka.

2.1 Navigačná časť

Hlavný komponent navigačnej časti tvorí menu. Jeho obsahom je zoznam všetkých vzdelávacích objektov. Zoznam je usporiadaný podľa kapitol, t.j. v rámci daných

vzdelávacích objektov sa môžu nachádzať aj ďalšie. Kliknutím na kapitolu sa vybraná položka rozbalí a zobrazia sa k nej patriace podkapitoly.

Menu (Obr. 3) sa skladá z troch záložiek:

- **Texty** – záložka obsahuje zoznam výučbových textov,
- **Otázky** – záložka obsahuje zoznam otázok, na ktoré je možné odpovedať,
- **Cvičenia** – záložka obsahuje zoznam cvičení, ktoré je možné vyriešiť.

The screenshot shows a web interface for a course. At the top, there is a navigation bar with a home icon, the course name 'AleF', and tabs for 'Administrácia', 'Lisp', 'Prolog', and 'C'. The user is identified as 'Vladimír Mihal (Student) | Odlásiť'. On the left, a sidebar menu is visible with sections for 'Podobné témy', 'Odporúčame', and a list of topics including '9 Funkce a práce s pamětí'. The main content area displays the selected lesson, '9 Funkce a práce s pamětí', with a filter bar and introductory text. On the right, a sidebar contains 'Tvoje skóre' (51.9), 'Nahlásené chyby', 'Externé zdroje', and 'Tagy'.

Obr. 3. Ukážka navigačnej časti.

V navigačnej časti sa okrem menu nachádza oblak kľúčových slov (nad menu, Obr. X), ktorý dopĺňa funkcionality menu. V oblaku sú zobrazené kľúčové pojmy, ktoré reprezentujú určité témy učiva.

Oblak slov pozostáva z dvoch častí:

- **Podobné témy** – obsahuje kľúčové slová, ktoré opisujú aktuálne otvorený vzdelávací objekt,
- **Odporúčame** – koncepty v tejto časti odporúčajú témy, ktoré sú určené na základe aktuálnych vedomostí študenta.

Konkrétne vzdelávacie objekty z danej témy sa získavajú kliknutím na vybrané kľúčové slovo v oblaku. Po kliknutí sa v menu vyznačia vzdelávacie objekty relevantné so zvoleným pojmom.

Po kliknutí na koncept sa pri vyznačených položkách v menu zobrazí jedna z anotácií uvedených nižšie:

- – vzdelávací objekt je do veľkej miery relevantný s vybraným slovom,

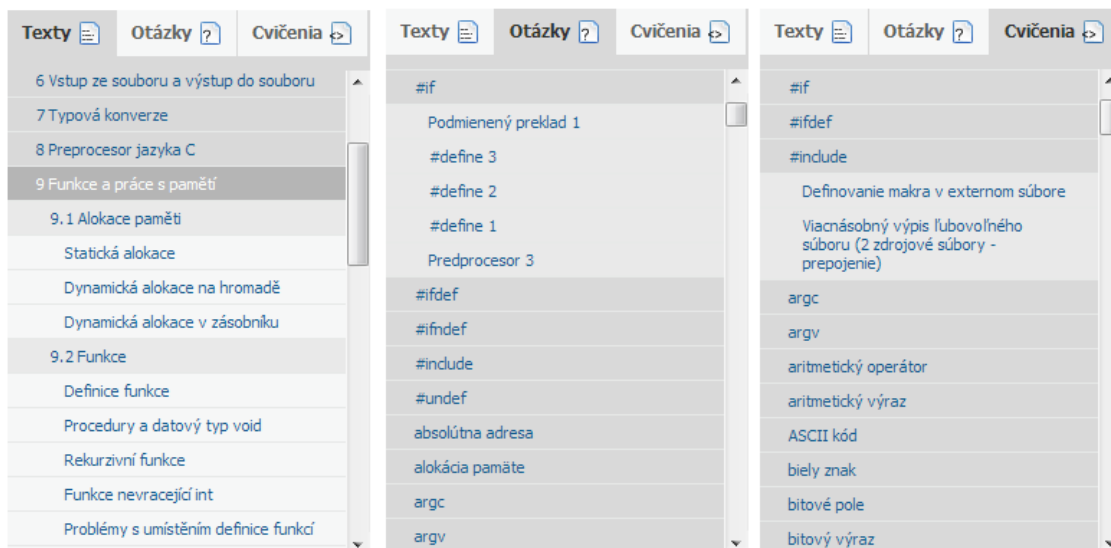
- 🌟 – vzdelávací objekt je mierne relevantný s vybraným slovom,
- ⭐ – relevancia vzdelávacieho objektu je minimálna,
- 🚫 – vzdelávací objekt nie je relevantný s výberom, ale v rámci neho sa nachádzajú podkapitoly, pri ktorých je zobrazená jedna z prvých troch anotácií, t.j. treba ho rozbaľiť, aby sme sa dostali k požadovaným materiálom.

Vyznačenie objektov sa dá zrušiť kliknutím na ikonu 🚫 v pravom dolnom rohu oblaku.

2.2 Obsahová časť

V závislosti od vybraného výučbového materiálu sa môžu v obsahovej časti nachádzať tri typy zobrazených materiálov:

1. **otázka,**
2. **cvičenie,**
3. **výučbový text.**



Obr. 4. Ukážky menu. Zľava záložka výučbových textov, v strede záložka otázok a vpravo záložka príkladov.

2.2.1 Odpovedanie na otázku

Otázka je typ výučbového materiálu, na ktorý sa dá odpovedať a systém vyhodnotí odpoveď študenta. V závislosti od typu otázok sa zobrazí príslušný formulár na jej zodpovedanie (Obr. 5). V každom formulári študent vyplní správnu odpoveď a klikne na tlačidlo *Odpovedaj*. Systém následne vyhodnotí odpoveď študenta a zobrazí výsledok (Obr. 6).

Makro s parametrom 1 Vaše aktuálne hodnotenie:

Čo platí pre makro s parametrom?

niekedy sa takéto makrá nazývajú aj vkladané funkcie

pred použitím makra v programe, sa makro nahradí konkrétnym textom

sú veľmi vhodné na nahradenie dlhých funkcií, lebo sa výpočet v nich prebieha rýchlejšie

syntakticky ich môžeme zapísať ako: #define meno_makra(arg1, arg2 ..., argN) hodnota_makra

Obr. 5. Formulár pre zodpovedanie otázky.

Makro s parametrom 1 Vaše aktuálne hodnotenie:

Táto otázka bola vyriešená

Čo platí pre makro s parametrom?

niekedy sa takéto makrá nazývajú aj vkladané funkcie

pred použitím makra v programe, sa makro nahradí konkrétnym textom

sú veľmi vhodné na nahradenie dlhých funkcií, lebo sa výpočet v nich prebieha rýchlejšie

syntakticky ich môžeme zapísať ako: #define meno_makra(arg1, arg2 ..., argN) hodnota_makra

Odpovedali ste nesprávne!

Odhodnoďte užitočnosť:

  
 Neúžitočný Neutrálny Užitočný

Odporúčame:

Tieto materiály vám môžu pomôcť:

- › [8.1 Makra bez parametru – príkaz define \(#define\)](#)
- › [8.2 Makra s parametrom \(makro\)](#)

Obr. 6. Zodpovedaná otázka.

Zelenou farbou sú vyznačené správne odpovede, červenou sú označené nesprávne odpovede. Čiernym rámčekom sú vyznačené odpovede, ktoré označil študent.

Pre rýchlu navigáciu možno použiť tlačidlo *Predchádzajúci príklad* resp. *Ďalší príklad*, ktoré zobrazia ďalšiu otázku na riešenie.

2.2.2 Odpovedanie na cvičenie

Cvičenie je typ výučbového materiálu, na ktorý sa dá vyriešiť a následne oznámiť systému postup v riešení (Obr. 7). Študent postupne kliká na jednotlivé možnosti, ktoré mu systém ponúka pri riešení. Možnosti sú nasledovné:

1. Poznám riešenie

Systém zobrazí správne riešenie. Následne si študent vyberie z možností:

- a. Moje riešenie je také isté, ako uvádza vzorové riešenie*
- b. Moje riešenie je iné, ale myslím, že správne*
- c. Moje riešenie je nesprávne, ale teraz už tomu rozumiem*
- d. Moje riešenie je nesprávne a stále tomu nerozumiem*

2. Nepoznám riešenie

Systém zobrazí pomôcku. Následne si študent vyberie z možností:

a. Už poznám riešenie

Systém zobrazí správne riešenie. Následne si študent vyberie z možností:

- i. Moje riešenie je také isté, ako vzorové riešenie*
- ii. Moje riešenie je iné, ale myslím, že správne*
- iii. Moje riešenie je nesprávne, ale už tomu rozumiem*
- iv. Moje riešenie je nesprávne a stále tomu nerozumiem*

b. Stále nepoznám riešenie

Systém zobrazí správne riešenie. Následne si študent vyberie z možností:

- i. Riešeniu rozumiem*
- ii. Riešeniu nerozumiem*

Pre rýchlu navigáciu je možné použiť tlačidlo *Predchádzajúci príklad* resp. *Ďalší príklad*, ktoré zobrazia ďalšie cvičenie na riešenie.

Definovanie makra na test veľkosti písmena

Vaše aktuálne hodnotenie:

Zadanie:

Definujte makro `je_veľké(x)`, ktoré vráti hodnotu `0` ak znak nie je veľké písmeno a hodnotu `1` ak je znak veľké písmeno.

Poznám riešenie

Nepoznám riešenie

Pavel Herout: Učebnice jazyka C - knihu si môžete zakúpiť v internetovom obchode www.eplanet.sk.

◀ Predchádzajúci príklad

Nasledujúci príklad ▶

Obr. 7. Formulár pre zodpovedanie cvičenia.

2.2.3 Výučbový text

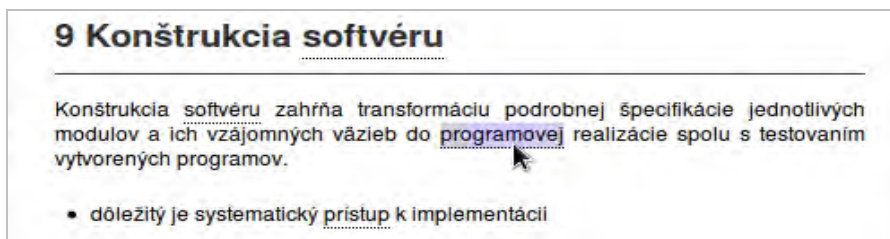
Výučbový text je statický text, ktorý môže obsahovať tabuľky, obrázky, grafy, hypertextové odkazy a ďalšie prvky. V niektorých výučbových textoch sú vložené vnorené cvičenia a otázky, ktoré sa viažu k danému výučbovému textu (Obr. 9). Tieto otázky a cvičenia je možné vyriešiť.

- Šípkami vľavo a vpravo sa možno pohybovať medzi vloženými otázkami a príkladmi,
- Samotné odpovedanie na otázku alebo príklad prebieha rovnako ako je to opísané v častiach Odpovedanie na otázka a Odpovedanie na cvičenie.

2.2.4 Poznámky priradené ku kľúčovým slovám v texte



K niektorým kľúčovým slovám vo výučbovom texte sú priradené automaticky generované poznámky. Tieto poznámky rozširujú informácie o danom kľúčovom slove tým že poskytujú jeho definíciu a odkazy na súvisiace stránky.

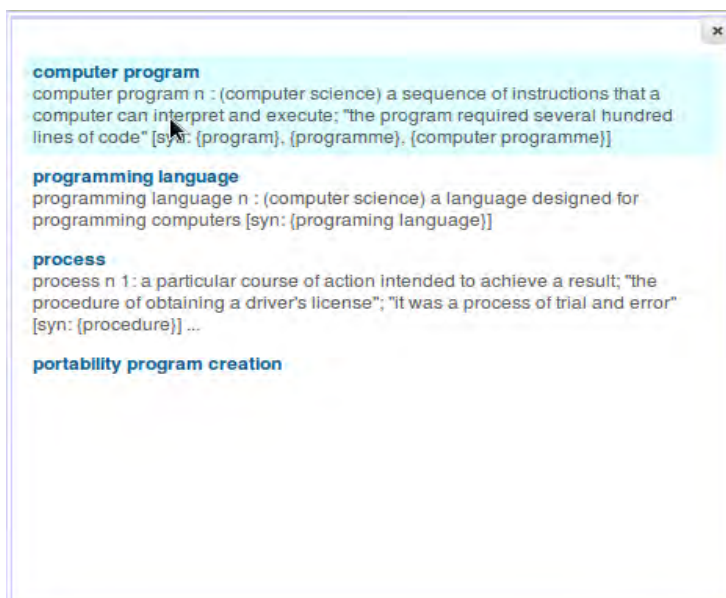
Slová, ku ktorým je takáto poznámka priradená sú v texte zvýraznené podčiarknutím a zmenou farby po prechode myšou nad takýmto slovom (Obr. 8).



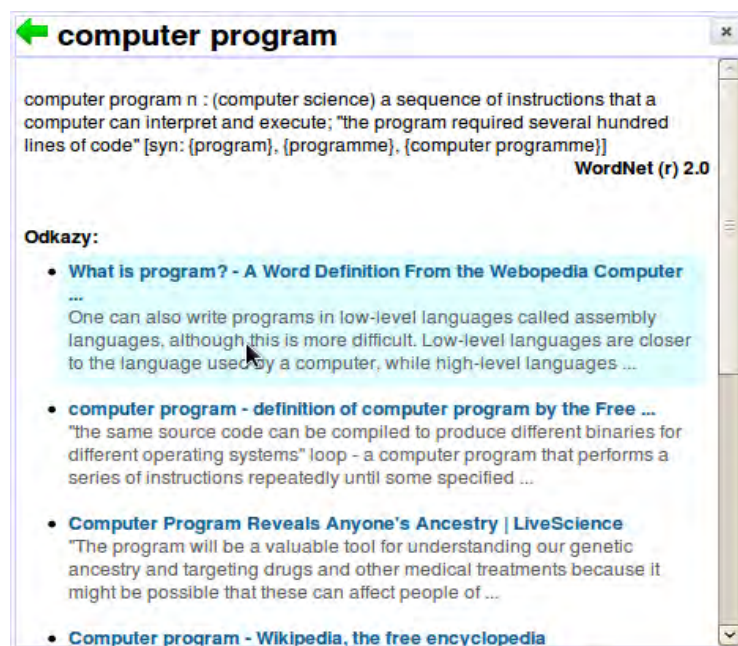
Obr. 8. Zvýraznenie slov ku ktorým je pripojená automaticky generovaná poznámka

Poznámka sa zobrazí vo forme tooltipu po kliknutí na zvýraznené slovo. Tooltip obsahujúci poznámku má viacero stavov medzi ktorými je možné prechádzať. Po kliknutí na zvýraznené slovo v texte sa zobrazí zoznam poznámok priradených k tomuto slovu (Obr. 9). Zvolením jednej z týchto poznámok sa tooltip presunie do stavu, kde zobrazuje obsah tejto poznámky (Obr. 10). Obsah poznámky tvorí definícia slova, zoznam odkazov na súvisiace stránky a zoznam pripojených prezentácií. Po kliknutí na odkaz na súvisiacu stránku sa na novej karte prehliadača otvorí táto stránka. Po kliknutí na odkaz zo zoznamu prezentácií sa tooltip presunie do ďalšieho stavu, kde bude v jeho tele zobrazená zvolená prezentácia.

V každom stave tooltipu je možné vrátiť sa o krok späť pomocou tlačidla . Zatvoriť tooltip je možné kliknutím mimo tooltipu do výučbového textu, kliknutím na tlačidlo  alebo stlačením klávesy Esc.



Obr. 9. Tooltip so zoznamom poznámok priradených k zvolenému slovu

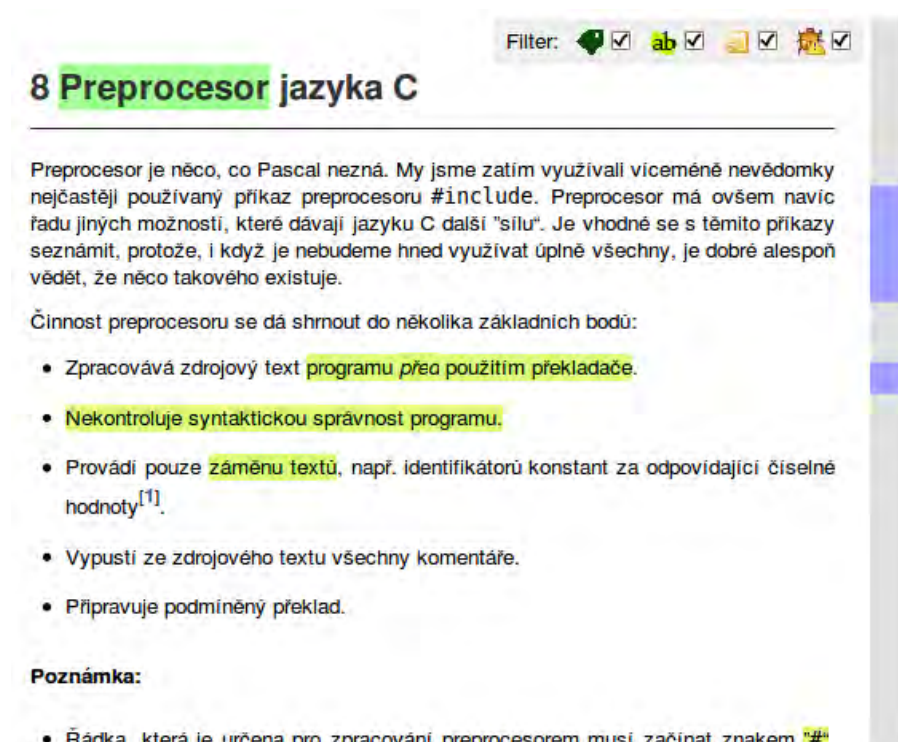


Obr. 10. Tooltip zobrazujúci obsah poznámky

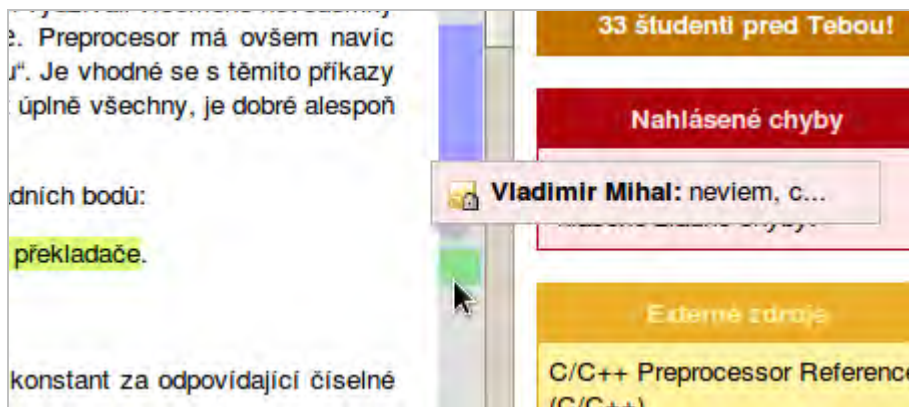
2.3 Anotačný pásik

Anotačný pásik (Obr. 11) znázorňuje komentáre vytvorené používateľmi a nahlásené chyby vo výučbových objektoch. Ich prítomnosť reprezentuje modrý obdĺžnik. Anotácia je zobrazená ukázaním kurzorom na modrú oblasť. Kliknutím alebo podržaním myši nad modrou oblasťou zostane anotácia zobrazená trvalo. Ďalším kliknutím ju opäť uvoľníme.

Pre filtrovanie medzi poznámkami a nahlásenými chybami slúži filter (Obr. 11 vpravo hore). Prvá ikona predstavuje zobrazenie tagov vložených z textu, druhá ikona predstavuje zvýraznené výseky textu, tretia predstavuje zobrazenie poznámok a posledná zobrazenie nahlásených chýb. Používateľ si tak môže zvoliť, ktoré anotácie chce v danom výučbovom materiáli vidieť.



Obr. 11. Anotčný pásik. Nachádza sa napravo od výučbového materiálu. Modrou sú znázornené anotácie vložené používateľmi.



Obr. 12. Zobrazenie anotácie.

2.4 Zvýraznenie dôležitých fragmentov

Zvýrazňovač dôležitosti fragmentov umožňuje podfarbiť časti zobrazeného výučbového objektu podľa informácií o tom, ako používatelia v minulosti pracovali s týmto objektom. Predstavuje tak pomôcku pri zrýchlenom učení, revízii, opakovaní už naučeného textu a pod.

Ovládacie tlačidlá (Obr. 13) sa zobrazia nad výučbovým objektom vľavo hore v prípade, že sú informácie o činnosti používateľov pre daný objekt dostupné a je

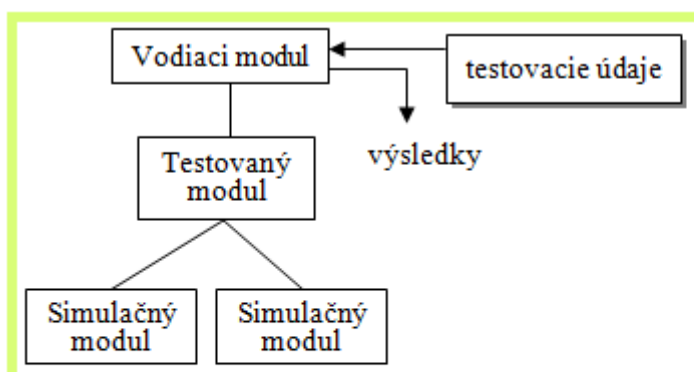
použitý podporovaný prehliadač (*Mozilla Firefox*). Jednotlivé tlačidlá umožňujú zobraziť zvýraznenie podľa primárneho zdroja informácií:

1. *Pohľad*, *Pohľad odhadnutý podľa myši* – podľa času, ktorý sa používateľia pozerali na dané časti dokumentu,
2. *Opotrebovanie dokumentu* – podľa virtuálneho opotrebovania dokumentu, čiže ako dlho boli jednotlivé časti zobrazené (vhodné pre dlhšie texty, ktorých obsah bolo potrebné posúvať),
3. *Zvýraznenia používateľov* – podľa ručného označovania dôležitých častí používateľmi.



10.4 Stratégie testovania

- prístup k testovaniu celého systému
 - testovanie zdola-nahor
 - testovanie zhora-nadol
 - jednofázové testovanie
 - sendvičové testovanie
 - testovanie porovnávaním



Testovanie zdola-nahor
angl. bottom-up testing

- vhodné, ak veľa modulov "nižšej" úrovne predstavuje všeobecné moduly, ktoré iné časti systému často používajú

Obr. 13. Zvýrazňovač dôležitosti fragmentov (vľavo hore).






2.5 Časť so zásuvnými modulmi

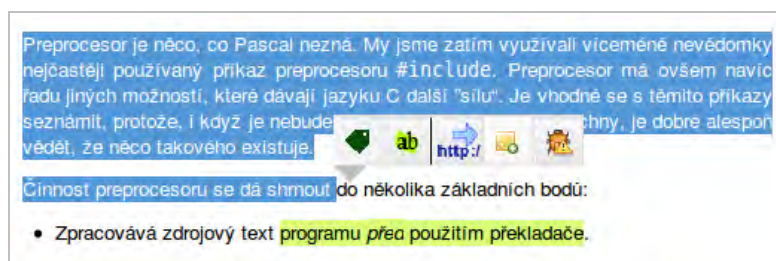
V pravej časti obrazovky sa nachádzajú zásuvné moduly systému. V aktuálnej verzii systému sa nachádzajú štyri moduly:

1. *Komponent pre zobrazenie skóre* – zobrazuje bodový zisk prihláseného študenta. Body sú študentovi pripisované v závislosti od aktivít, ktoré vykoná v súvislosti s rôznymi časťami systému (pridá tag do textu, nahlási chybu, pridá poznámku ...).
2. *Komponent pre zobrazovanie nahlásených chýb* – zobrazuje nahlásené chyby vo výučbových materiáloch od používateľov.
3. *Komponent pre zobrazenie a pridávanie externých odkazov* – zobrazuje externé zdroje, ktoré pridali študenti alebo vyučujúci k danému výučbovému materiálu. Umožňuje tiež pridať ďalšie externé zdroje.
4. *Komponent pre zobrazenie a pridávanie tagov* – zobrazuje tagy priradené k danému výučbovému materiálu. Umožňuje tiež pridať ďalšie tagy či už priamo jeho napísaním alebo označením časti textu, ktorý sa má použiť ako tag.

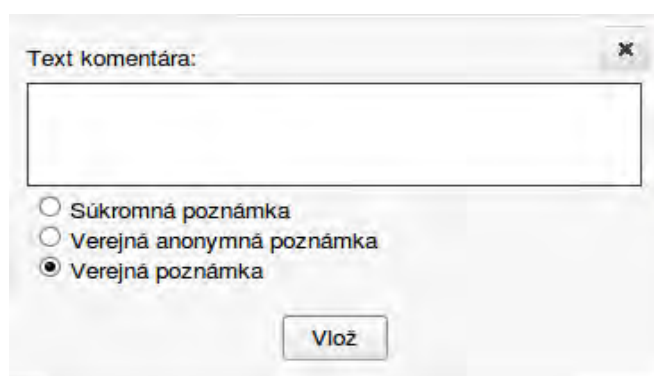
2.6 Pridávanie poznámok

Pridávať rôzne druhy poznámok sa v systéme dá k ľubovoľnému výučbovému materiálu (textom, otázkam a aj príkladom). Používateľ označí text, ku ktorému chce pridať poznámku. Následne si z kontextového menu môže vybrať jednu z možností (Obr. 14):

1.  *Pridanie tagu* – slúži na pridanie vyznačeného textu ako tag. Tagy sa zobrazujú zvýraznením pozadia otagovaného textu ako aj v komponente na pridávanie a zobrazovanie tagov.
2.  *Označenie textu* – vyznačenému textu sa zmení pozadie. Táto funkcia slúži na zvýraznenie dôležitých alebo inak zaujímavých častí textu.
3.  *Pridanie externého zdroju* – pomocou tejto voľby je možné k textu priradiť externý odkaz, ktorý sa bude zobrazovať v komponente na pridávanie a zobrazovanie externých odkazov.
4.  *Nový komentár* (Obr. 15) – slúži na pridanie poznámky k výučbovému materiálu. Poznámka sa zobrazuje na mieste, ku ktorému bola pridaná. Poznámka je viditeľná pre všetkých používateľov.
5.  *Nahlásenie chyby* – umožňuje pridať poznámku, ktorá upozorňuje ostatných študentov a učiteľa, že v danom výučbovom materiáli sa nachádza chyba.



Obr. 14. Označení textu.



Obr. 15. Rozhranie pre pridanie komentáru.

2.7 Komponent pre zobrazovanie skóre

Komponent pre zobrazenie skóre (Obr. 16) slúži ako informatívny prvok pre študentov, ktorý sa využíva pri hre za účelom motivácie. Hra spočíva v získavaní bodov a snahe študentov dosiahnuť čo najvyššie skóre. Komponent zobrazuje aktuálnu výšku bodov, ktoré študent získal za rôzne akcie v systéme. Komponent zároveň zisťuje poradie študenta v celkovom bodovom poradí a tento údaj zobrazuje, za účelom motivácie študentov.





Obr. 16. Komponent pre zobrazenie skóre.

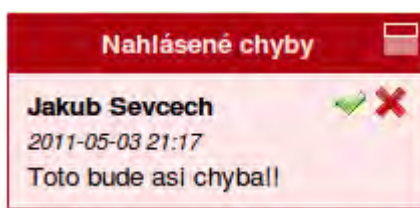
2.8 Komponent pre zobrazenie nahlásených chýb

Komponent pre zobrazenie nahlásených chýb (Obr. 17) slúži ako informatívny prvok. Komponent zobrazuje nahlásené chyby od používateľov v tabuľke v poradí:

používateľské meno autora nahlásenej chyby, čas nahlásenia chyby a text hlásenia resp. popis chyby. Po prechode myšou nad textom hlásenia, vo výučbovom texte sa zvýrazní časť textu, ku ktorému bola chyba nahlásená. V pravej hornej časti sú ku každej nahlásenej chybe zobrazené dve ikony:

-  Slúži na označenie chyby za vyriešenú, takto označená chyba sa ďalej nebude zobrazovať.
-  Slúži na zmazanie chybového hlásenia

Tieto ikony sú dostupné len pre používateľov s rolou *Administrátor* a *Učiteľ*.

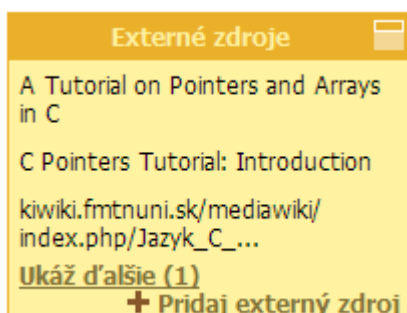


Obr. 17. Komponent pre zobrazenie nahlásených chýb.

2.9 Komponent pre prácu s externými zdrojmi

Komponent pre prácu s externými zdrojmi umožňuje používateľom vkladať odkazy na stránky s obsahom súvisiacim s obsahom aktuálne zobrazeného vzdelávacieho objektu. Používatelia môžu taktiež tieto odkazy hodnotiť. Rovnako ako u iných typov poznámok, pri vkladaní externých zdrojov môže používateľ špecifikovať ich viditeľnosť – teda či sú súkromné, anonymné alebo verejné. Formulár na vkladanie odkazu na externý zdroj sa zobrazí po kliknutí na odkaz „+ Pridaj externý zdroj“.

V komponente sú zobrazené všetky externé zdroje vložené k momentálne zobrazenému vzdelávaciemu objektu. Po zobrazení objektu sú v komponente zobrazené iba tri externé zdroje s najvyšším hodnotením od používateľov. Všetky externé zdroje použiteľ zobrazí kliknutím na odkaz „Ukáž ďalšie“.

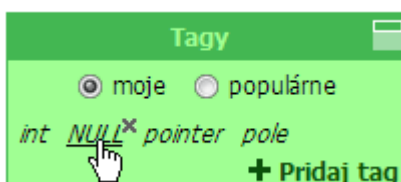


Obr. 18. Komponent pre prácu s externými zdrojmi.

2.10 Komponent pre zobrazenie a pridavanie tagov

Komponent pre zobrazenie a pridavanie tagov umožňuje používateľom pridávať tagy ku vzdelávacím objektom, prehliadať ich a odstraňovať. Používatelia môžu pri vkladaní tagu špecifikovať, či bude vložený ako verejný, anonymný alebo súkromný. Tag môže používateľ odstrániť nadíšením nad tag a kliknutím na tlačidlo „x“.

Používateľ môže prepínať medzi zobrazením jeho vložených tagov a populárnych tagov. Populárnym sa tag vtedy, pokiaľ je verejný a je k vzdelávaciemu objektu vložený minimálne tromi používateľmi.

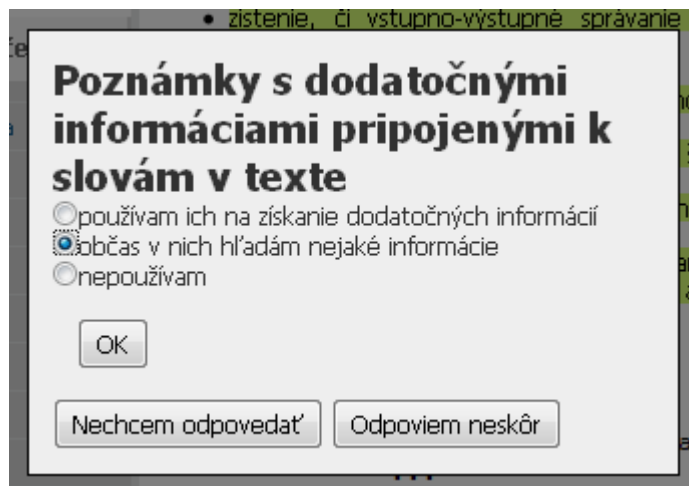


Obr. 19. Komponent pre zobrazenie a pridavanie tagov.

2.11 Adaptívne systémové otázky

System môže v prípade vhodnej situácie vygenerovať a zobraziť otázku (Obr. 20), slúžiacu väčšinou na získanie odozvy od používateľa na funkcie systému. Používateľ odpovie vyplnením odpovede podľa typu otázky (výber jednej možnosti, vpísanie reakcie a pod.) a stlačením tlačidla *OK* odpoveď odošle. Ak používateľ nechce na otázku odpovedať, môže stlačením tlačidla *Nechcem odpovedať* alebo *Odpoviem neskôr* odpoveď zrušiť, respektíve odložiť.

V prípade odmietnutia odpovede sa práve zobrazená otázka deaktivuje a už sa nezobrazí (čo však nevyklučuje, že systém neskôr nevygeneruje rovnakú otázku nanovo, ak to bude vhodné). V prípade odloženia odpovede sa otázka prestane zobrazovať, ale o krátku chvíľu sa zobrazí znova. Niektoré otázky sú špeciálne zamerané na aktuálne okamihy a preto sa už znova nezobrazia, ani ak ich používateľ iba odloží tlačidlom *Odpoviem neskôr*.



Obr. 20. Príklad systémovej otázky. Zobrazí sa v popredí pred výučbovým objektom.

PRÍLOHA H – POUŽÍVATEĽSKÁ PRÍRUČKA ROZŠÍRENIA UTRACK

Používateľská príručka k rozšíreniu UTrack zodpovedá jednoduchým pokynom, ktoré boli v rámci experimentu predvedené študentom v ústnej forme.

Slovenská Technická Univerzita
Fakulta informatiky a informačných technológií

Používateľská príručka rozšírenia UTrack

Martin Labaj

máj, 2011

Obsah

1 PREVZATIE ROZŠÍRENIA/INŠTALÁCIA.....	1
1.1 Inštalácia rozšírenia do prehliadača	1
1.2 Prevzatie prenosnej verzie prehliadača s rozšírením.....	1
1.3 Požadované knižnice	1
2 POUŽÍVANIE ROZŠÍRENIA.....	2

1 Prevzatie rozšírenia/inštalácia

Aplikácia UTrack je rozšírenie prehliadača Firefox umožňujúce získavanie implicitnej spätnej väzby o práci používateľa, najmä so zapojením sledovania pohľadu používateľa. Aktuálnu verziu a stručné pokyny možno nájsť na webovej adrese:

<http://utrack.labaj.sk/>

1.1 Inštalácia rozšírenia do prehliadača

Rozšírenie je možné nainštalovať štandardným spôsobom do nainštalovaného prehliadača Firefox, vyžadovaná je verzia 3.6. Vzhľadom na účely rozšírenia a niektoré zmeny vykonávané rozšírením je odporúčané využiť samostatný profil prehliadača alebo priamo prevziať samostatne fungujúcu prenosnú verziu prehliadača s priloženým rozšírením (pozri časť 1.2). Adresa pre inštaláciu aktuálnej verzie rozšírenia je:

http://utrack.labaj.sk/extension/utrack_current.xpi

1.2 Prevzatie prenosnej verzie prehliadača s rozšírením

Rozšírenie je tiež pripravené vo forme balíka s prenosnou verziou prehliadača Firefox 3.6 (nie je potrebné inštalovať). Stačí tak prevziať súbor na nasledujúcej adrese, rozbalíť ho a spustiť súbor *UTrackFirefox.bat*:

http://utrack.labaj.sk/utrack_firefox.zip

1.3 Požadované knižnice

Pre správnu funkciu získavania pohľadu z obrazu kamery je potrebné mať nainštalované aplikačné knižnice MS Visual C++ 2005 SP1 Runtime (verzie 2008 a 2010 sú oddelené a tieto verzie nie sú navzájom kompatibilné, nutná je verzia 2005), dostupné na adrese:

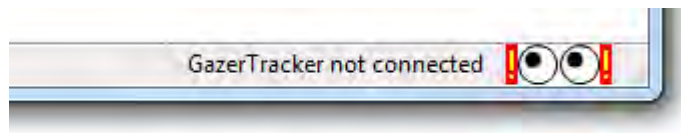
<http://www.microsoft.com/downloads/en/details.aspx?familyid=200b2fd9-ae1a-4a14-984d-389c36f85647>

Tiež je potrebný aplikačný rámec Microsoft .NET Framework 3.5 SP1, dostupný na adrese:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=ab99342f-5d1a-413d-8319-81da479ab0d7>

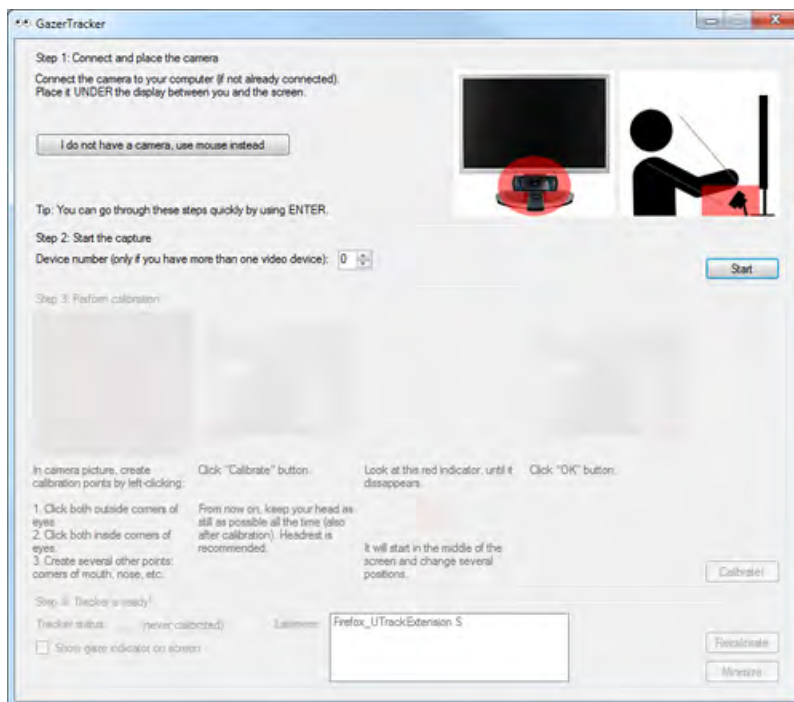
2 Používanie rozšírenia

Po spustení prehliadača s nainštalovaným rozšírením sa v stavovom paneli prehliadača v pravom dolnom rohu zobrazí ikona rozšírenia spolu s aktuálnym stavom kalibrácie (obr. 1).



Obrázok 1: Indikácia stavu rozšírenia v stavovom paneli prehliadača.

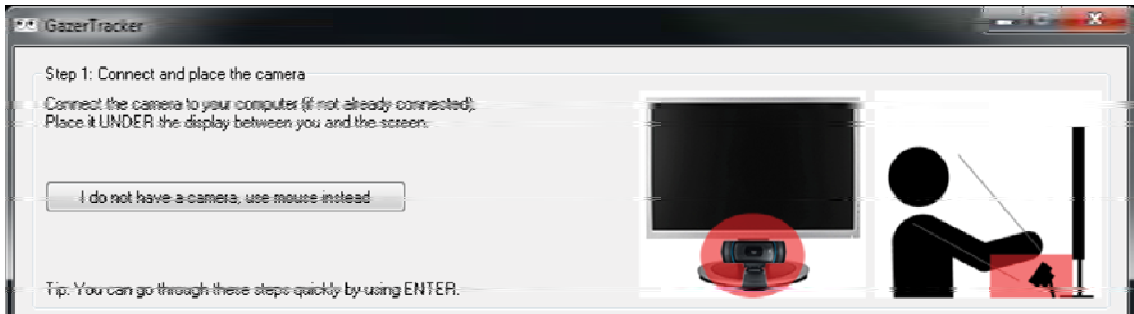
Pred pokračovaním v práci je potrebné vykonať kalibráciu sledovania pohľadu. Po spustení rozšírenia sa automaticky otvorí konfiguračné okno aplikácie GazerTracker pre sledovanie pohľadu (obr. 2), v ktorom je zvýraznený aktuálny krok a ostatné sú vizuálne potlačené (zobrazené sivou farbou).



Obrázok 2: Konfiguračné okno aplikácie GazerTracker pre sledovanie pohľadu.

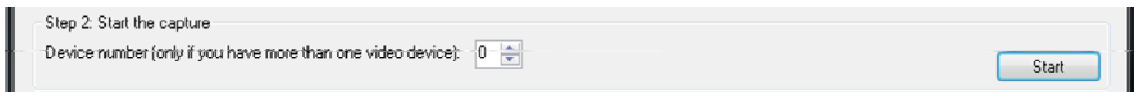
Následne stačí postupovať podľa postupne zvyrazňovaných pokynov na obrazovke. V prvom kroku je potrebné pripojiť a správne umiestniť kameru pod displej (obr. 3).

V prípade, že používateľ nemá k dispozícii kameru, vyberie možnosť „*I do not have a camera, use mouse instead*“ (Nemám kameru, použi namiesto nej myš) a na simuláciu sledovania pohľadu sa využije pozícia kurzora myši. V prípade zvolenia tejto simulácie pohľadu myšou už nie je potrebné vykonávať žiadne ďalšie kroky, používateľ môže okno minimalizovať a pokračovať v práci.



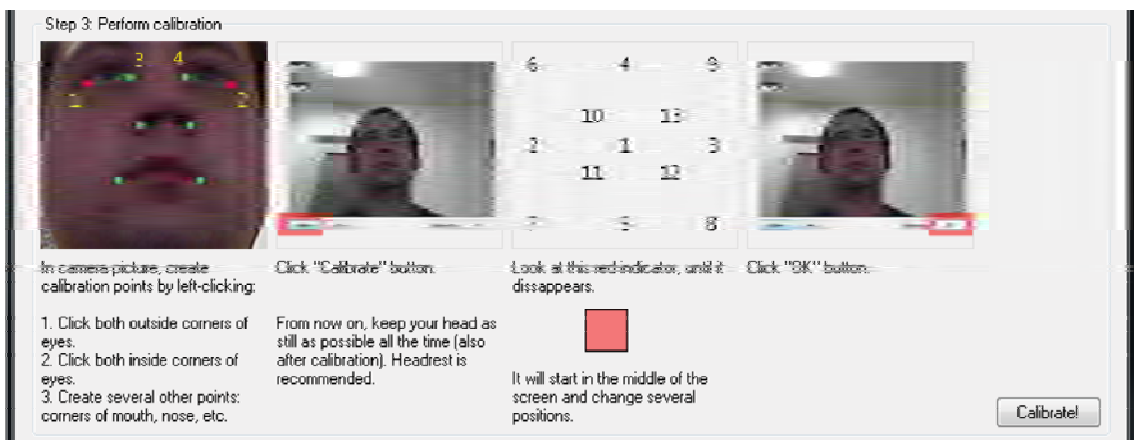
Obrázok 3: Krok 1, pripojenie a umiestnenie kamery.

Ďalej používateľ vyberie poradové číslo kamery v systéme (iba v prípade viacerých kamier pripojených súčasne) a stlačením tlačidla *Start* pripojí obraz kamery (obr. 4).



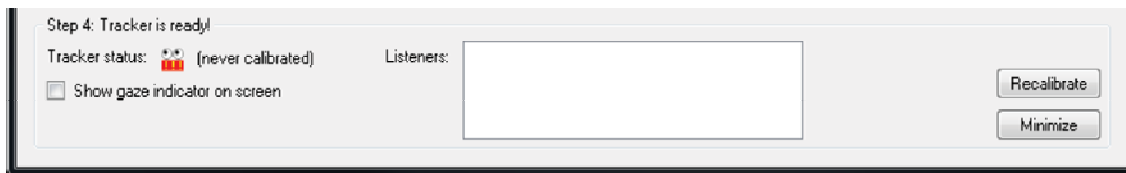
Obrázok 4: Krok 2, výber čísla kamery a pripojenie jej obrazu.

Následne je potrebné vykonať kalibráciu. Používateľ najskôr stlačí tlačidlo *Calibrate*, čím sa zobrazí kalibračné okno, klikne v obraze kamery na vonkajšie rohy očí, potom na vnútorné a vyberie ešte niekoľko charakteristických bodov na tvári. Následne stlačí tlačidlo *Calibrate* a sleduje pohľadom červený štvorec, ktorý postupne mení polohu na obrazovke. Po zmiznutí štvorca zavrie kalibračné okno stlačením tlačidla *OK*.



Obrázok 5: Krok 3, kalibrácia zamerania pohľadu.

Po správnom vykonaní kalibrácie sa zobrazí ikona stavu sledovača pohľadu zelenou farbou a používateľ môže konfiguračné okno minimalizovať do systémovej lišty tlačidlom *Minimize*.



Obrázok 6: Krok 4, minimalizácia konfiguračného okna.

Následne môže používateľ pokračovať v práci tak, ako zvyčajne.

PRÍLOHA I – OBSAH ELEKTRONICKEJ PRÍLOHY

Adresár	Obsah
<i>\Bin</i>	Vydaná aplikácia
<i>\!required</i>	Požadované knižnice (.NET, VC Redist)
<i>\UTrack_web</i>	Webová stránka obsahujúca rozšírenie UTrack so zahrnutým sledovačom pohľadu GazerTracker a prenosnú verziu prehliadača s predinštalovaným rozšírením
<i>\Doc</i>	Tento dokument
<i>\Eval</i>	Podklady z vyhodnotenia riešenia (dotazník, tabuľky), výpis databázy systému ALEF
<i>\Papers</i>	Články odoslané na zahraničné odborné konferencie
<i>\Src</i>	Zdrojové súbory
<i>\Alef</i>	Výučbový systém ALEF
<i>\GazerTracker</i>	Sledovač pohľadu
<i>\UTrack_extension</i>	Sledovacie rozšírenie prehliadača Firefox