

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÝ PORTÁL PRO ANOTACI OBRAZOVÝCH DAT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ PETRUŽELKA

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÝ PORTÁL PRO ANOTACI OBRAZOVÝCH DAT

WEB PORTAL FOR VISUAL DATA ANNOTATING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ PETRUŽELKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ALEŠ LÁNÍK

BRNO 2012

Abstrakt

Cílem práce je navržení a implementování integrovatelného webového rozhraní pro anotování obrazových dat. Toto rozhraní bude interagovat s daným serverem za účelem načtení a ukládání anotací a získávání doporučených anotací.

Abstract

The purpose of this thesis is to design and implement an embeddable interface for visual data annotating. This interface would communicate with a given server for the purpose of acquiring and persisting data and receiving suggested annotations.

Klíčová slova

uživatelské rozhraní, webové rozhraní, grafické rozhraní, anotování, obrazová data, JavaScript

Keywords

user interface, web interface, graphical interface, annotating, visual data, JavaScript

Citace

Jiří Petruželka: Webový portál pro anotaci obrazových dat, bakalářská práce, Brno, FIT VUT v Brně, 2012

Webový portál pro anotaci obrazových dat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Aleše Láníka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jiří Petruželka
10. května 2012

Poděkování

Rád bych poděkoval vedoucímu práce panu Ing. Aleši Láníkovi za poskytnuté informace a rady, všem účastníkům testování a autorům nástrojů a knihoven, užitých při tvorbě bakalářské práce. V neposlední řadě své rodině za podporu a členům komunity freenode#bbg za četné rady.

© Jiří Petruželka, 2012.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Obdobná řešení	4
2.1	VIA	4
2.2	Google+	5
2.3	Szoter	5
3	Popis problematiky	6
3.1	Anotování	6
3.2	Uživatelská rozhraní	6
3.2.1	Přístupnost	7
3.2.2	Konzistence	8
3.2.3	Uživatelské chyby, prevence a korekce	8
3.2.4	Uživatelské akce	8
4	Vlastní řešení	9
4.1	Zaměření	9
4.2	Konzistence a terminologie	9
4.3	Vlastnosti	11
4.3.1	Přístupnost	11
4.3.2	Internacionalizace	11
4.3.3	Odezva	11
4.3.4	Rozložení	12
4.4	Uživatelské akce	13
4.4.1	Primární akce	13
4.4.2	Doplňkové akce	14
4.4.3	Uživatelské chyby, prevence a korekce	15
4.4.4	Zvýšení efektivity	16
4.5	Požadavky rozhraní	16
4.6	Implementace	17
4.6.1	Metoda zobrazování	17
4.6.2	JavaScript, CoffeeScript	18
4.6.3	Moduly	18
4.6.4	Další knihovny	19
4.6.5	Třídy	19
4.6.6	Datový formát	22
4.6.7	Nastavení a přizpůsobení	24
4.6.8	Accessor a použití	24

4.6.9	Optimalizace	24
5	Testování	25
5.1	Osobní testování	25
5.1.1	Schéma jednotlivého testování	25
5.1.2	První testovací sezení	26
5.1.3	Druhé testovací sezení	27
5.1.4	Třetí testovací sezení	28
5.2	Dotazník	29
5.2.1	Výsledky	29
6	Závěr	30
A	Obsah CD	33
B	Detaily použitých knihoven	34
C	Manuál	35
C.1	Použití z hlediska vývojáře	35
C.1.1	MarkArtAccessor	36
C.2	Ukázkový server a proxy	36
C.3	Výchozí klávesové zkratky	37
D	Dotazník	38
E	Testované prostředí	43
E.1	Klientská část	43
E.2	Serverová část	43

Kapitola 1

Úvod

Cílem této práce je návrh a vytvoření webového uživatelského rozhraní pro poloautomatické anotování obrazových dat. Toto má být schopno zobrazovat vyznačené části v anotovaném médiu a jim přiřazené anotace. Uživateli rozhraní má umožnit vytvářet nové značky a přiřazovat jim vlastní popisky. Ze serveru přijme na základě těchto přidáných anotací seznam doporučených anotací, které mají být nabídnuty uživateli k přidání. Data má být schopno předat zpět serveru pro uložení.

V dnešní době velmi narostla na objemu práce s digitálními daty a mutlimédií, a tím i potřeba jejich zařazení, prohledávání, stanovení kontextů či vztahů mezi nimi. Anotační rozhraní má vytváření těchto metainformací zprostředkovávat a usnadňovat. Anotovat je možno zcela automatizovaně, poloautomatizovaně (kdy člověk kontroluje či využívá anotací vybraných algoritmů) či manuálně.[24] Tato práce se zabývá poloautomatizovaným anotováním, kdy uživatel sám vkládá anotace (zde textové popisky), další jsou mu pak nabízeny jako návrhy na základě již přiřazených anotací (tento algoritmus není součástí této práce a může být výrazně pozměněn).

Webové rozhraní má pak výhodu dostupnosti z mnoha zařízení a platform bez nutnosti instalace zvláštního programu, s možností sdílení dat mezi mnoha uživateli a podpory ze strany prohlížeče.

Práce má za cíl vytvoření přístupného a nenáročného webového rozhraní pro doplňování anotací do obrazových dat a nabízení doporučených anotací. Uživatel má mít možnost rozhraní zkoušet a experimentovat a přitom být schopen provedené úkony opravit či vrátit zpět. Pokročilejší a seznámenější anotující mají mít rozšířené možnosti pro urychlení prováděných činností. Pro uživatele má být jednoduché k pochopení a klást jen minimální nároky. Rozhraní vytvořené v rámci této práce má být jednoduše integrovatelné do webových stránek a poskytovat maximální možnost přizpůsobení pro své účely.

Ve druhé kapitole budou popsány vybraná stávající řešení rozhraní anotace obrazových dat. Ve třetí kapitole je nastíněna základní teorie, ze které řešení vytvořené v této práci vycházelo. Čtvrtá kapitola je rozdělena na první sekce zabývající se návrhem řešení, tj. jak se v řešení odráží teorie ze třetí kapitoly, a následně sekcí popisující implementaci z technického hlediska. Pátá kapitola pojednává o uživatelském testování tohoto rozhraní, rozebírá jednotlivá sezení a schéma a analýzu dotazníku.

Kapitola 2

Obdobná řešení

V této kapitole budou stručně popsány příklady stávajících řešení.

2.1 VIA

VIA je desktopový program pro anotování obrázků a videí od CERTH-ITI. V popisu se zaměřím výhradně na funkcionality spojené s anotováním obrázků. Údaje vycházejí z osobního testování a z [4].

V záložce **Image** umožňuje nastavit globální informace o obrázku (název, soubor), anotujícím a přiřadit anotace pro celý obrázek. Záložka **Regions** slouží k definici oblastí, pro které se vztahují anotace. Anotace pro celý obrázek i oblast se přidávají v příslušné kartě jednotně kliknutím na tlačítko s popiskem „+“ (odebrání pak „-“), stejně tak vytváření nových oblastí.

VIA nabízí tvary oblasti obdélník, kružnici, elipsu, polygon, volný tvar a segment (ten je pouze generován). Prvním třem typům oblastí je možno měnit rozměr, pozici a tvar. Rozměr a pozici lze bez výběru nástroje měnit přímo v obraze, pro změnu velikosti je nutno najet přesně na kontrolní značky. Měnit pozici je možno za celou vnitřní plochu oblasti, v případě překryvu značek je řešeno pomocí interní hodnoty určující hodnotu na pomyslné ose Z. To je možno ovlivnit tlačítky **Bring to Front**, **Sent to back**, přesná hodnota pro každou oblast však není z uživatelského rozhraní zjistitelná. VIA obsahuje příkaz pro slučování oblastí, ten ovšem konvertuje všechny tvary na segment (nepodařilo se mi funkci zcela pochopit ani nalézt dokumentaci). Oblast lze zvětšit nebo přesunout za hranice obrazu.

Při vytváření nové oblasti se interaguje s obrázkem pouze u tvarů polygon a volný tvar, kdy je nutno vybrat příslušné body. Tyto pak nelze přímo modifikovat, lze tvar zrušit a vytvořit znova. Při vytváření je tvar značen tenkou černou linkou. V bočním panelu jsou editovatelná pole pro hodnoty pozice a rozměru, mění se podle vybrané oblasti, avšak je nutno nejdříve dvojkliknout, aby byly následné změny registrovány.

Segmenty lze na požádání vygenerovat pomocí externí aplikace.

V obrázku jsou zobrazeny pouze vyznačené oblasti, anotace jsou vypsány po výběru příslušné oblasti v boxu v bočním panelu. Anotace lze kromě vepsání také přidat tažením z načteného seznamu anotací.

Program se ptá na uložení změn pouze při uzavření přes položku menu, nikoli u jiných způsoby ukončení. Neumožňuje vrácení akcí.

2.2 Google+

Google+ je webová sociální síť od společnosti Google Inc., která mimo jiné umožňuje nahrávání fotografií a jejich úpravu, doplňování informací, komentářů a označování osob. Nástroj je implementován pomocí JavaScriptu, anotované oblasti jsou vizualizovány pomocí CSS. Údaje vycházejí z vlastního testování.

Jakmile máme otevřený anotovaný obrázek, po vybrání nástroje pro označení osob je možné vložit kruhovou oblast a k ní přiřadit referenci na daného člověka. Vkládá se jediným kliknutím na konkrétní místo obrazu, tažení myši neprovede žádnou akci (pokud si člověk akci rozmyslí, lze vytvoření takto zrušit). Kliknutím do vnitřního prostoru existující oblasti se neprovede vložení, je třeba kliknout jinde a oblast poté přesunout zpět na dané místo. Oblast je vždy kruhová a nelze zvětšit nebo přesunout za hranice obrázku. Z toho také vyplývá, že by měla být vždy dostupná plocha kam kliknout pro vytvoření oblasti. Oblast je možné tažením za kontrolní body zvětšit či zmenšit, tažením za vnitřní plochu lze změnit pozici.

Všechny anotace jsou standardně skryty, je možné je zobrazit po najetí kurzorem na jejich plochu (tudíž je dobré si pamatovat kde jsou) nebo najetím kurzorem na příslušnou položku na bočním panelu (zde se zobrazí výrazněji, okolí je ztmaveno).

K celému obrázku je taktéž možno přidávat komentáře, tj. textový popis obsahující také autora a čas přidání. Neumožňuje vrácení akci. Dokument je ihned ukládán a tak není třeba upozorňovat na neuložený dokument a možno záložku uzavřít či obnovit.

2.3 Szoter

Szoter je online nástroj pro anotaci obrázků implementovaný pomocí technologie Adobe Flash. Údaje vycházejí z vlastního testování.

Tato aplikace umožňuje do editovaného rámce přidat více fotografií a tyto rotovat (po 90°) nebo zrcadlit horizontálně či vertikálně. Je také možné přesouvat či měnit velikost samotného rámce a změnit míru přiblížení.

Szoter podporuje tvary čáru, rovnou úsečku, kružnici/elipsu, obdélník a šipku. Text se vkládá přímo jako objekt text. K dispozici je paleta barev a také výběr přesné barvy. Značkám je možno měnit pozici, velikost a rotovat je a měnit tloušťku jejich linky. Je možno je přesouvat po pomyslné ose Z. Značky se vybírají kliknutím, ovšem obrázek pod značkami je také vybrat rovněž. Szoter umožňuje vrácení a obnovování akci. Neupozorňuje při uzavření aplikace, je tak možno ztratit anotační data.

Kapitola 3

Popis problematiky

3.1 Anotování

Následující anotační teorie vychází z [21] a [24].

Anotování je proces, kdy přiřazujeme k objektu či datům (často multimediálním) sémantické informace, které mohou shrnovat obsah, sloužit k filtrování, třídění a prohledávání dat, poskytovat doplňující informace o autoru, okolnostech vzniku dat, jejich vztahu k jiným objektům, nebo se může jednat například o překlad mluveného slova ve videu či vět v textu. Tyto anotace mohou být opět multimediálního rázu, textové popisky či odkazy na jiná data a metadata aj. Je možné je dělit na popisná (kterou se zabývá většina nástrojů), strukturní (pro navigaci) a administrativní (pro správu a zpracování). Anotace se může vztahovat k celému objektu, nebo (vyšší granularita) k určité části objektu, například části obrázku nebo časového úseku videa.

Anotační proces se dělí na manuální, poloautomatizovaný a automatizovaný. Manuální anotování předpokládá uživatele anotačního rozhraní, který pomocí tohoto zadává příslušná pomocná data či odkazy na základě vlastních vědomostí nebo analýzy dat (zde prohlédnutí příslušného obrazu). Automatické anotační programy jsou ty, které samy přiřazují anotace na základě analýzy daných dat, dostupných metadat a jiných informací, ty však nejsou součástí této práce. poloautomatické anotační systémy urychlují proces anotování doporučováním tagů na základě analýzy obrazu, již přiřazených tagů nebo jiného algoritmu. Zároveň jsou přesnější než zcela automatizované anotátory, jelikož jsou jejich výsledky kontrolovány člověkem.

V této práci bude řešena poloautomatizované anotování, kdy uživatel bude zadávat textové popisky a na základě přiřazených anotací mu budou nabízeny (výběr však není součástí práce) další popisky k přidání.

3.2 Uživatelská rozhraní

Následující teorie uživatelských rozhraní vychází z [22], [19] a [25].

Uživatelské rozhraní ve výpočetní technice je rozhraní zaměřené na uživatele, který tak nemusí znát podrobnosti implementace programu (příp. zařízení), které má obsluhovat, a odbornou terminologií a znalostí k ní se vážící. Má uživateli umožnit vykonávání jeho práce, a to jednoduše a efektivně. Rozhraní využívá k předání informací vjemy zachytitelné lidskými smyslovými orgány, jsou to primárně zrak (možno předat velké množství

informací zaráz) a sluch (doplňkově, o to může mít i při nízké intenzitě dobrou zaznamatelnost). Uživatel pak může obsluhovat rozhraní pohybem (často za pomoci ovládacích zařízení typu myš, klávesnice a jiné), řečí (příkazy) a hmatem (gesta, mimika). Webové uživatelské rozhraní je pak rozhraní, jež je součástí webových stránek na Internetu. U toho je pak primárním výstupním zařízením monitor různorodých zařízení (u nevidomých pak hlasová čtečka), sekundárně reproduktory.

Jelikož uživatelské rozhraní má zásadní vliv na samotnou možnost obsluhy a příjemnost a efektivitu práce s daným programem, zařízením či informací, je třeba věnovat návrhu a zpracování náležitou pozornost. Hodnocení použitelnosti rozhraní lze dělit na pět kategorií: dobu potřebnou pro naučení se práce s rozhraním; rychlost vykonávání umožněné činnosti; druh a četnost chyb, kterých se uživatel může při práci dopouštět; uchování znalostí o práci s programem; subjektivní vjem z práce s programem [22]. Není vždy zcela možné uspět ve všech kritériích, některá se mohou se mohou vyvažovat, jinak je třeba si stanovit priority.

Hlavní doporučené zásady, tzv. „Zlatá pravidla“ (Golden Rules) návrhu uživatelského rozhraní jsou:

1. Konzistence
2. Univerzální použitelnost
3. Zpětná vazba systému
4. Dojem začátku, průběhu a zakončení akcí
5. Prevence chyb
6. Umožnit vrácení či opravu akcí
7. Uživatel má mít kontrolu nad systémem
8. Minimalizovat nároky na krátkodobou paměť

3.2.1 Přístupnost

Důležitou vlastností rozhraní je jeho přístupnost – umožnit obsluhu napříč různými zařízeními a lidem bez ohledu na schopnosti, znalosti, věk, původ či vjemové, fyzické, emociální nebo psychické nedokonalosti. Zvyšování přístupnosti klade v mnoha případech zvýšené nároky na implementaci a návrh, kdy může být nutné, aby se přizpůsoboval podle uživatele, případně hledat nejmenší společnou použitelnou množinu pro daný případ, nebo zcela změnit koncept daného aspektu tak, aby zahrnoval širší skupinu potenciálních uživatelů. Vylepšení přístupnosti pro menšinové skupiny má nezřídka pozitivní dopad i na zbývající část.

S přístupností souvisí nároky na uživatele. Odborný časopis *Economics Abstracts* zmiňuje tyto kognitivní procesy: [22]

- Krátkodobá a pracovní paměť
- Dlouhodobá a sémantická paměť
- Řešení problémů a uvažování
- Rozhodování se a zvažování rizik

- Jazyková komunikace a pochopení
- Hledání, vybavování si, senzorická paměť
- Učení se, osvojování si dovedností, znalostí a přijetí konceptů

Kromě těchto schopností se lidé liší zkušenostmi, věkem i charakterem. Na člověka mohou také působit četné vnitřní i vnější vlivy.

3.2.2 Konzistence

Aby byl uživatel schopen se dobře naučit ovládání rozhraní, udržet tuto znalost v paměti a pracovat efektivně, je třeba aby bylo rozhraní konzistentní. Konzistence se týká velké řady aspektů: použité terminologie, způsobů ovládání, vizuálního stylu. U vizuálního stylu se pak přesněji jedná o barvy, rozložení, ikony, font, velikost tlačítek aj. [22] Nekonzistence navýší množství chyb uživatele během používání rozhraní.

Pokud je ovšem prvek napříč systémem konzistentní, porušení v jednom určitém případě může způsobit jeho zvýraznění a případně zapamatování. Také se může jednat o prevenci potenciálně nebezpečné operace (více viz 4.4.3). Veškerá porušení doporučení vzešlých z teorie uživatelských rozhraní je ovšem nutno mít řádně promyšlené, zdůvodněné a následně provést testování, zda má toto vybočení opravdu očekávaný vliv. Čím zažitější a standardnější konkrétní prvek je, tím důležitější je udržet konzistenci a tím mnohem více narůstají nároky na přínos porušení daného pravidla, mělo-li by se porušit. Pro udržení konzistence jsou vytvářeny směrnice a terminologické slovníky, pro tento projekt je naleznete v následující kapitole v sekci 4.2.

3.2.3 Uživatelské chyby, prevence a korekce

Uživatelé mají tendenci používat zařízení a rozhraní často bez snahy o pochopení jejich fungování. Místo studování příruček raději ovládání zkouší. Rozhraní by toto mělo respektovat a umožnit uživateli experimentovat bez toho, aby se musel obávat jakýchkoli následků. K tomu lze užít mechanismus historie a vracení akcí a poskytnutí nástrojů pro manuální upravení stavu na zamýšlený. Ke zvýšení pravděpodobnosti správného odhadu funkčnosti lze zařadit stručnou nápovědu přímo do aplikace, snadno pochopitelné označení ovládacích prvků a výrazné odlišení prvků s destruktivními následky (odstraňování, blokování. . .), kdy je porušení jistých pravidel (např. příliš malá velikost tlačítka) obhájitelné.

3.2.4 Uživatelské akce

U rozhraní je definován soubor možných akcí, které je s ním možno provádět. Tyto akce mají své předpoklady, způsob (či více způsobů) vykonání a definovaný stav po svém dokončení. Část definovaných akcí tvoří primární zájem rozhraní, další mohou sloužit ke korekci či být efektivnější, ale třeba náročnější, variací primárních akcí. Akce se také mohou lišit předpokládanou četností jejich využívání. Akce má mít uživatelem rozlišitelné zahájení, průběh a zakončení. Uživatel má být také průběžně informován systémem o jeho stavu. I zkušený uživatelé se často dopouští chyb a proto je úkolem rozhraní míru chybovosti snížit, upozornit na nežádoucí stav a poskytnout nástroje k nápravě stavu způsobeného chybnou akcí.

Kapitola 4

Vlastní řešení

Tato kapitola nejdříve seznámí s cílem a zaměřením rozhraní a používanou terminologií, dále popíše návrh a následně implementaci a možnosti nastavení a přizpůsobení.

4.1 Zaměření

Při tvorbě tohoto anotačního rozhraní byl kladen důraz na jeho přívětivost, přístupnost a dostatečnou možnost korekce. Jelikož uživatelé rádi experimentují a zkoušejí než dlouze zkoumají (viz 3.2.3), byla snaha, aby uživatel mohl jakoukoli akci co nejjednodušeji vrátit či opravit. Mělo by být maximálně konfigurovatelné a upravitelné, aby šlo využít na co nejširším spektru stránek. Dále by mělo rozhraní zefektivňovat proces anotování.

4.2 Konzistence a terminologie

Z důvodu potřeby konzistence (nastíněné v 3.2.2) byl vytvořen terminologický slovník (tabulka 4.1) a dále směrnice pro vizuální styl, jenž jsou použity pro výchozí nabízený vzhled a jsou následující:

Písmo

velikost je 13 pixelů, nemělo by jít níže jak 12 pixelů

barva je `rgb(40,40,40)`, v případě tmavého pozadí `rgb(255,255,255)`; pro zvýraznění na tmavém pozadí lze použít `rgb(255,165,65)`

rodina písma je Segoe UI, při nedostupnosti Arial či Sans-serif

Barvy

V základu je užito světlé pozadí. Aktivní a aktivovatelné či výrazné prvky jsou barvou `rgb(40,100,160)`, při najetí při možnosti aktivovat `rgb(25,55,85)`. Pro zvýraznění neaktivních prvků se užívá `rgb(80,80,80)`. V každém případě je však nutný vysoký kontrast.

Barevné konvence jsou na několika místech záměrně porušeny z důvodu příliš výrazné barvy na příliš rozsáhlé ploše (menu, přiřazené tagy v seznamu pod obrázkem) či pokud prvek nemá být výrazný (box oznamující komunikaci se serverem).

Tvary

Je používáno širokých obdélníků s vnitřním okrajem standardně 3 pixely shora a zezdola, 12 pixelů ze stran. Rohy jsou mírně zaobleny. Všechny klikatelné tvary musí být dostatečně velké, aby bylo možné je pohodlně vybrat kurzorem myši.

Kurzor

Veškeré texty uživatelského rozhraní mají standardní kurzor. Editovatelné texty a pole mají kurzor textový. Všechny prvky aktivovatelné kliknutím (převážně tlačítka) mají kurzor v podobě ruky. Speciální nástroje mohou aktivovat vlastní kurzor, ale jen pro situace, kdy uživatel bez dalších úkonů může danou akci provést. Při načítání se použije čekající kurzor.

Interakce

Každý nástroj (kromě **Prohlédnout**) lze použít tažením myši po obrázku. Kromě tohoto základu může nástroj definovat vlastní alternativní možnosti použití.

Tento vzhled je možno modifikovat pomocí vlastních kaskádových stylů (například pro různé kulturní oblasti).

Význam	Český termín	Anglický termín
vyznačený tvar a všechny přidružené tagy	značka	mark
vizuální část značky / tvar	tvar značky	mark's shape
anotační text	tag (popisek)	tag
anotovaný prvek	obrázek	picture
vytvořit nový prvek	vytvořit	create
přidat prvek	přidat	add
odstranit prvek	odstranit	remove
změnit vlastnost prvku	změnit	change
změnit lokaci prvku	přesunout	move
změnit velikost prvku	zvětšit/zmenšit	scale
změnit rotaci prvku	otočit	rotate
uvést data do stavu před poslední akci	vrátit	undo
uvést data do stavu po poslední akci	obnovit	redo
uživatelská akce	akce	action
stisknout tlačítko myši	kliknout	click
držení stisknutého tlačítka a pohyb myši	tažení	dragging
ukončení stisku tlačítka	uvolnění	release
značka v množině aktuálně upravovaných značek	vybraná značka	selected mark
odstranit značky z množiny aktuálně upravovaných	zrušit výběr	deselect all
získávat data ze serveru	načítání dat	loading data

Tabulka 4.1: Přehled používaných termínů

4.3 Vlastnosti

4.3.1 Přístupnost

Aplikaci je možno kompletně ovládat pomocí klávesnice. Nebylo snahou o dosažení ekvivalentní efektivity, tato možnost tu však je. Mnoho funkčnosti se spoléhá na využití (standardně) klávesy **Tab** pro přesun mezi aktivovatelnými prvky, některé mají definovány klávesové zkratky (viz příloha C.3). Všechny prvky, které lze aktivovat (např. tlačítka), obsahují textový popis. Ikony pro výběr nástrojů a akce pro historii v rychlém menu mají vyplněn text také, pouze je skryt – ale pouze v případě, že je zobrazeno hlavní menu (a dále mají vyplněn atribut `title`) resp. prvky pro vrácení/obnovení akce, a tak je uživateli k dispozici textový význam ikon. Všechny prvky s definovanou odezvou na najetí myši (**hover**) mají definovanou i reakci na výběr klávesnicí (**focus**).

4.3.2 Internacionalizace

Každý textový prvek rozhraní je možné lokalizovat pro daný jazyk. Řešení umožňuje kaskádovitou definici překladů, kdy se nejdříve použije definice z co nejspecifičtější jazykové větve (např. `en-uk-london`) a postupně zkouší až po nejobecnější (`en`). Je možno přizpůsobit i formátovací funkce (výpis času). Do rozhraní je možno doplňovat vlastní jazykové definice, v základu je k dispozici čeština, angličtina a velmi lehce britská angličtina.

Hlavnímu kontejneru pro rozhraní jsou přiřazeny třídy s názvy ekvivalentními zadaným kulturním identifikátorům, a tudíž je možné dělat verze vizuálně odlišné pro jiné kultury. S použitím vlastního kódu v JavaScriptu jde detekci přítomnosti těchto tříd (případně i použitím jiných kritérií) vytvořit i odlišné chování či rozložení rozhraní.

4.3.3 Odezva

Vždy, když uživatel vykoná v implementovaném rozhraní akci, je o tomto informován prostřednictvím přiřazení záznamu do historie akcí a samozřejmě samotným projevem akce. Pokud se jedná pouze o změnu nastavení rozhraní, je přítomna vizuální identifikace (změna ikony a textu u vypnutí automatického ukládání, změna barvy a písma u výběru nástroje atp.). Aktivovatelné prvky mají nastaven stejný kurzor jako odkazy na webové stránky (tj. *ruka*) a změnu barvy při najetí myši či zaměření pomocí klávesnice, klávesy **Tab**.

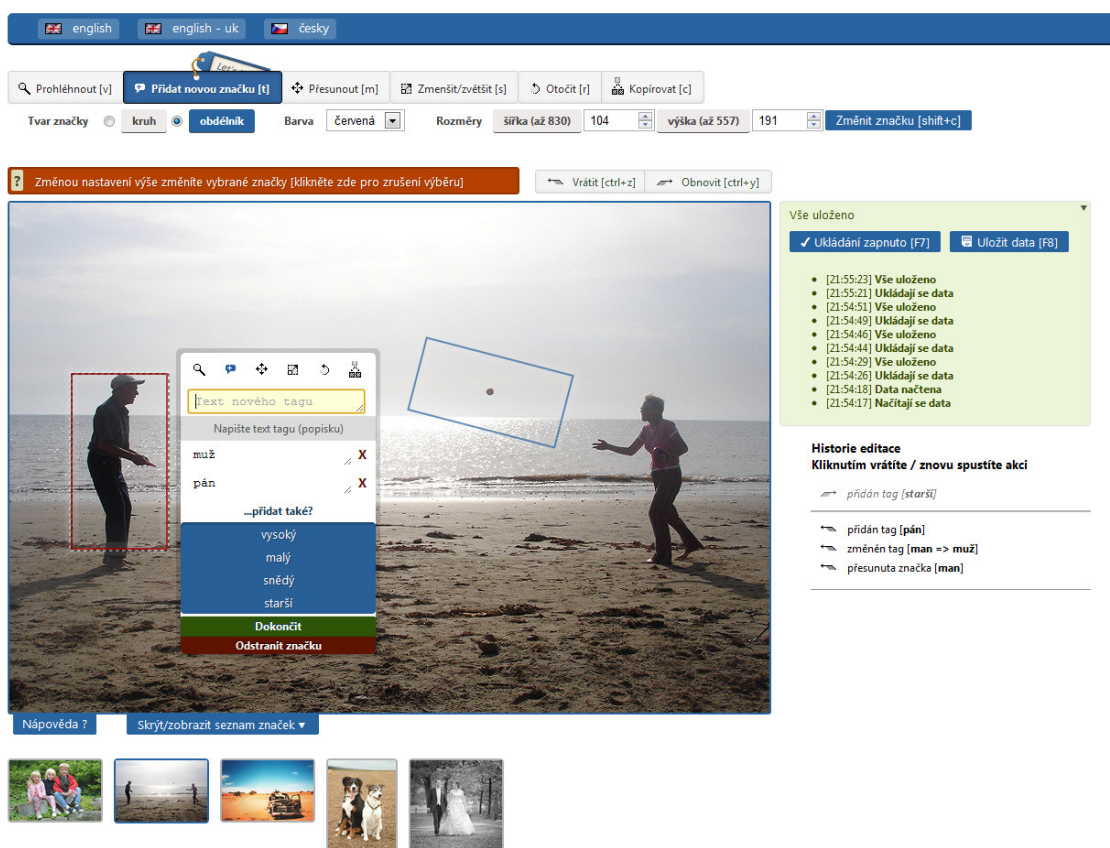
Zahajovací načítání je pojato výrazně a anotace a anotovaný obrázek jsou překryty poloprůhledným elementem s vycentrovanou informující textovou hláškou a je nastaven čekací kurzor. Při následných načítáních doporučených tagů se již načítá méně výrazně, není důvod blokovat.

Během vytváření značky je uživateli podávána informace o finálním vzhledu značky. Stejně tak při úpravách značky, navíc se zobrazuje i náhled tagů. Tento náhled pro všechny značky si uživatel může kdykoli zobrazit podržením příslušné klávesy (viz příloha C.3).

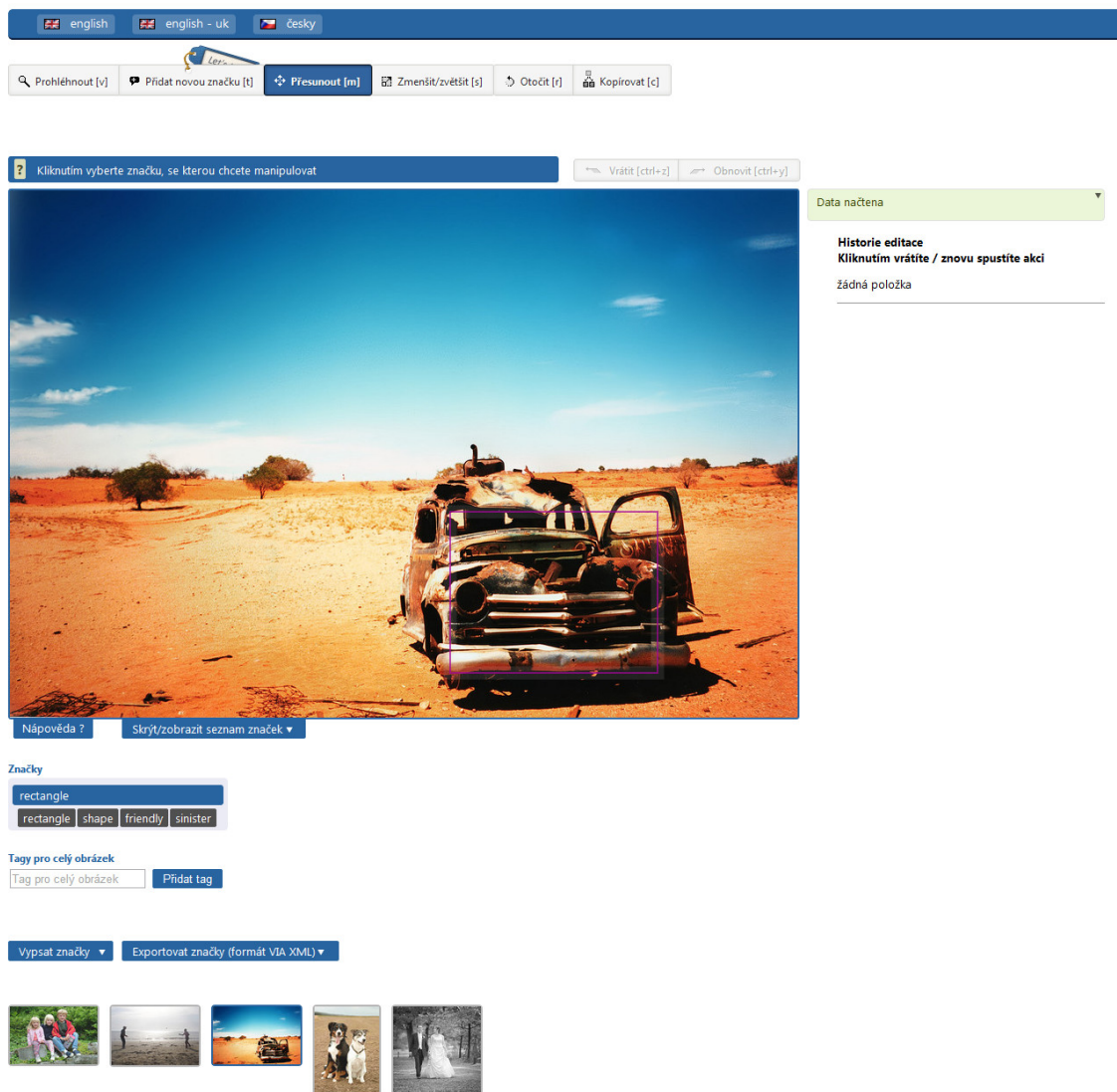
Anotující je průběžně informován o komunikaci se serverem, konkrétně o načítání a ukládání. V případě selhání je zobrazena textová hláška s výraznou barvou. Tyto textové informace mají definovanou minimální dobu zobrazení, je však možno panel rozkliknout a vidět výpis proběhlých stavů pro případ, že přece jen uživatel konkrétní stav přehlédl. Při opětovném navázání komunikace je uživatel znovu informován. Dále je přítomen decentní indikátor stavu uložení dokumentu v titulku stránky.

4.3.4 Rozložení

Webová stránka s implementovaným rozhraním je zachycena na obrázcích 4.1 a 4.2. V horní části rozhraní je menu, kde je uživateli očekáváno a kde uživatel okamžitě vidí, ve kterém režimu se nachází a které základní akce má k dispozici. Následují případně specifická nastavení související s aktuálním nástrojem (režimem) následované mezerou, díky tomuto vidí uživatel spojitost. Dalším prvkem je pruh s nápovědným textem a vpravo na jeho boku tlačítka pro vrácení resp. obnovení akce. Tyto prvky jsou velmi blízko anotovanému obrázku, aby byly co nejlépe vidět a nebyly příliš vzdálené pro aktivaci myši. Následuje anotovaný obrázek a na jeho boční straně (při úzkém zobrazovacím zařízení pak pod obrázkem) je blok popisující aktuální stav komunikace se serverem, pod ním výpis provedených akcí. Tyto prvky jsou na boku, jelikož jsou to povahou sloupcové prvky a jiné umístění by příliš odsunovalo ostatní prvky. V obrázku jsou potom vizualizovány tvary jednotlivých značek, při výběru také ovládací panel značky, při aktivaci také rychlé menu. V režimu prohlížení pak mohou být zobrazeny přímo anotace. Pod obrázkem je tlačítka pro otevření nápovědy a skrývatelný blok vypisující jednotlivé značky a jejich tagy. První tag značky je zvýrazněn. Na boku by byly příliš odsunut (nebo naopak příliš odsouvaly) seznam historie akcí a nad obrázkem by velmi odsunuly obrázek samotný, vzdálenost anotovaného obrázku od horní hrany by byla příliš velká, mohlo by se stát, že při načtení by nebyl vůbec vidět. Rozhraní je zakončeno galerií, jelikož je rozhraní zaměřeno na anotování, nikoli však procházení obrázku, tudíž by změna obrázku měla být výjimečnou akcí.



Obrázek 4.1: Snímek implementovaného rozhraní s výchozím stylem, editace značky



Obrázek 4.2: Snímek implementovaného rozhraní s výchozím stylem, viditelné rozložení

4.4 Uživatelské akce

Následuje rozbor akcí, které může uživatel provádět. Primární akce jsou ty, jež přímo souvisejí s hlavní náplní rozhraní. Doplnkové akce jsou akce, které slouží ke zpřesnění, úpravě či opravě stavu. Veškeré následující akce je možno vrátit pomocí mechanismu historie.

4.4.1 Primární akce

Vytvoření značky

Tato akce je dostupná přes nástroj **Přidat novou značku**. Existují v zásadě čtyři způsoby provedení. Jednotný způsob je tažením myši po obrázku z výchozího do koncového bodu. Během tohoto může vytváření zrušit klávesou **Esc**. Zda vytváření probíhá ze středu či z okraje je indikován pomocným tvarem kolem kurzoru. Druhá možnost je jediný klik do plochy obrázku (mimo přítomné tvary značek a jejich okraje)

a druhým klikem do koncového bodu. Uživatel nemusí takto stále udržovat tlačítko stisknuté, může i zcela myš pustit, případně akci zrušit stiskem pravého tlačítka nebo klávesou **Esc**. Při pohybu myši je zobrazen náhled finální značky. Třetí možnost je jediný klik za současného držení klávesy **Shift**, kdy se okamžitě značka přidá s tvarem odpovídajícím pomocnému tvaru kolem kurzoru. Čtvrtý způsob je vytvoření kliknutím na tlačítko **Přidat značku** či stisknutím příslušné klávesové zkratky **Shift+c**, kdy je vytvořena značka s rozměry uvedenými v nastavení nástroje a pozicí ve středu obrázku. Jedná se o způsob jak přidat značku čistě pomocí klávesnice.

Po dokončení akce je přidána nová značka bez přiřazených tagů. Její barva a tvar jsou stanoveny podle nastavení tohoto nástroje.

Přiřazení nového tagu

Po přidání značky je aktivováno políčko pro vyplnění nového tagu pro tuto značku. Po vyplnění a stisknutí klávesy **Enter** nebo kliknutím na tlačítko **Přidat nový tag (popisek)** nebo tlačítko **Dokončit** je tento tag přidán a server je dotázán na doporučené tagy.

Přiřazení navrženého tagu

Tato akce realizuje poloautomatizovaný aspekt aplikace. Po dokončení akce **Přiřazení nového tagu** jsou uživateli nabídnuty serverem doporučené tagy (jsou-li nějaké) ve formě seznamu pod již přiřazenými tagy. Pokud je seznam už přítomen, jsou do něj přiřazeny získané položky (nikoli pokud by u některé nastala duplicita). Uživatel poté jediným kliknutím tlačítkem myši – anebo stisknutím klávesy **Enter** či mezerníku, pokud je daný tag vybrán – doporučený tag přidá mezi přiřazené, vizuálně je odebrán ze seznamu doporučených a je vytvořena položka v přiřazených tazích.

Výběr značky

Akce důležitá pro opětovnou editaci již vytvořených značek. Výběr značky se provádí kliknutím na tvar nebo okraj tvaru dané značky (ten je vizuálně zvýrazněn) nebo kliknutím na položku dané značky (nebo použitím klávesy **Enter** či mezerníku) ve výpisu značek pod obrázkem (pokud je rozbalen). Mezi položkou ve výpisu a tvarem v obrázku je vizuální provázanost, při najetí myši či vybrání klávesnicí je zvýrazněn prvek v obrázku a naopak. To je také důležité pro ovládání čistě pomocí klávesnice, a tím spíše pokud mají některé značky stejné tagy, lze tak rozlišit, o kterou značku se jedná. Výběr je možno rozšiřovat popsáním způsobem za držení klávesy **Ctrl**. Kliknutím do prostoru obrázku při momentálně editovaném textovém poli toto pole ztratí fokus. Pokud žádné editováno není, je množina vybraných značek vyprázdněna. Dále jsou k dispozici klávesové zkratky pro zrušení a invertování výběru, výběru všech, výběru posledně vytvořené/editované značky a výběru následující/předchozí značky.

4.4.2 Doplnkové akce

Odstranění značky

Uživatel má možnost odstranit celou značku včetně všech jejích tagů, a to pomocí tlačítka **Odstranit značku** v ovládacím panelu dané značky (to je záměrně až dole a zmenšeno). Dále je možné použít klávesu **Delete**. Po provedení je značka kompletně odstraněna.

Odstranění tagu

Také je možno odstranit jednotlivé tagy aktivováním tlačítka s písmenem **X** (a skrytým

textem **Odstranit tag**). Po použití je tag odebrán, je však přidán mezi doporučené tagy a tak je ho možno okamžitě zase vrátit.

Upravit tag

Tag je možno i změnit, přiřazené tagy jsou vepsány do editačních formulářových polí a jejich přepsáním se po opuštění pole nebo stisknutím klávesy **Enter** či **Tab** nový text značky uloží a aktualizuje se seznam doporučených tagů.

Přesunout tvar značky

Pokud uživatel neumístil značku přesně kam chtěl, může použít nástroj **Přesunout**. Tažením myši kdekoli po obrázku nebo použitím klávesových šipek může přesunout vybrané značky na libovolné místo. Také tažením za tvar či okraj značky může přesouvat i bez výběru, tato značka je následně vybrána. Při držení **Shift** je posunováno jen v jednom směru.

Zmenšit či zvětšit tvar značky

Pokud uživatel nevytvořil značku s požadovanou velikostí, může použít nástroj **Zmenšit/zvětšit**. Tažením myši kdekoli po obrázku nebo použitím klávesových šipek může změnit velikost vybrané značky. Také tažením za tvar či okraj značky může měnit velikost i bez výběru, tato značka je následně vybrána. Při držení **Shift** je tvar měněn horizontálně i vertikálně stejnou měrou, nehledě na nastavení stejné funkcionality nástroje.

Otočit tvar značky

Dalším doplňujícím nástrojem je **Otočit**. Tažením myši kdekoli po obrázku nebo použitím klávesových šipek se provede rotace vybraných značek. Také tažením za tvar či okraj značky lze provést rotaci i bez výběru, tato značka je následně vybrána. U kružnice pak určuje pouze výchozí rotaci značek.

Kopírování značky

Pokud je třeba vytvořit stejnou či podobnou značku s podobnými tagy, může být jednodušší kopírovat stávající značku a tu pak jen upravit. Tento nástroj funguje stejným způsobem jako nástroj **Přesunout**, avšak původní značka setrvává na svém místě a přesunována je nově vytvořená. Nová značka má stejný tvar, barvu, zvětšení, rotaci i navržené a doporučené tagy jako původní. Při držení **Shift** je posunováno jen v jednom směru.

Přidání tagu pro celý obrázek

Tato akce je řešena pomocí textového pole a tlačítka ve spodní části rozhraní, jelikož se předpokládá primárně použití pozicovaných značek. Vyplněním pole a stisknutím klávesy **Enter** či tlačítka je přidán nový tag a vypsány doporučené tagy.

4.4.3 Uživatelské chyby, prevence a korekce

Z důvodu nastíněných v 3.2.3 je toto řešení zaměřeno na umožnění experimentování a možnost úprav a vracení akcí. K ruční úpravě daného stavu značek slouží doplňkové akce popsané v předchozí části 4.4.2. Pokročilejším uživatelům je umožněno zkoušení za pomoci ovládání ukládání, tj. může zrušit automatické ukládání a jakmile je spokojen, provede uložení. Kliknutí pravým tlačítkem do prostoru obrázku vyvolá rychlé menu, kde je možno přepínat nástroje, měnit nastavení či vracet akce. Je zobrazeno v blízkosti myši a jeho použití je tedy poměrně rychlé.

Uživatel je chráněn před ztrátou dat v důsledku opuštění stránky v neuloženém stavu pomocí potvrzujícího dialogu. Ten je zobrazen i v případě pokusu o uzavření záložky či celého prohlížeče. Zda je stav uložen či ne, je indikováno v popisku záložky (a pokud ta je aktivní, tak i popisku okna rozhraní operačního systému). Jsou implementovány také interní mechanismy snižující šanci uložení neplatného stavu v případě chyby v kódu. Jinak jsou anotace průběžně automaticky ukládány.

Důležitou částí je historie akcí, uživatel může vracet či znovu provádět úkony. Toto může provést buď po jednom (pomocí tlačítek či standardních klávesových zkratk `Ctrl+z`, `Ctrl+y`) nebo může vrátit i obnovit větší množství akcí zároveň. Při provedení akce je však smazán seznam akcí k obnovení, jelikož by vedl k neplatnému stavu. Uživateli je označen stav, ve kterém se v rámci historie nachází. Při najetí myši či výběru klávesnici jsou označeny všechny akce, které budou ovlivněny.

Pro mírné snížení množství překlepů u tagů je implementováno našeptávání, více o něm v části 4.4.4.

Další možností vylepšení tohoto aspektu by byla implementace ukládání jednotlivých stavů a možnost vracení se k nim, nejlépe včetně lokálního uložení.

4.4.4 Zvýšení efektivity

Zde budou popsány implementovaná řešení vedoucí urychlení vytváření a úprav značek. Při vytvoření i vybrání značky jsou rozměry, typ a barva pro následující značku nastaveny podle ní a pomocí vytváření s klávesou `Shift` (viz 4.4.1) jde takto rychle přidávat značky se stejným vzhledem. Nástroj kopírování umožňuje rychlé vytvoření mnoha značek. Všechny manipulační nástroje a akce **Odstranění značky** lze provádět nad všemi vybranými značkami zároveň. Jsou implementovány klávesové zkratky (seznam naleznete v příloze C.3) pro rychlejší provedení dané akce či umožnění speciálních akcí výběru.

K dispozici je našeptávání tagů při přidávání nového tagu (ať už znače nebo celému obrázku). Tento seznam je uchováván pomocí `cookies`, tudíž jsou našeptávány i napříč více obrázky. Při zadávání stejných tagů více značkám tak nemusí uživatel vypsát celý výraz, stačí několik písmen. Vyhledávání v zadaných tazích probíhá pomocí algoritmu, kdy jsou nalezeny i výrazy s překlepy nebo vynechanými písmeny, a na základě míry a způsobu shody je stanoveno pořadí nabídnutí. Lze tak i předejít překlepům, kdy uživatel může vidět, že nabídnutý tag je onen zamýšlený, avšak liší se od jím napsaného. Po proběhlém testování byl seznam nabídnutých tagů nastaven pouze na ty, které uživatel přidal během aktuálního sezení, jinak bylo chování matoucí, tím spíše, že už jeden nabídnutý seznam – doporučení serveru – byl přítomen. Díky tomu, že se nabízejí jen ty, jež před chvílí přidal, je lépe vidět korelace a význam tohoto mechanismu.

Výrazným zlepšením by byla možnost kopírovat značky napříč různými anotovanými obrázky.

4.5 Požadavky rozhraní

Pro plnohodnotné využití rozhraní předpokládá alespoň schopnost vidět, číst a ovládat klávesnici, případně prostředek se stejnými možnostmi jako klávesnice (hlasová zařízení...). Je třeba základní znalost práce s počítačem a webovými prvky a nejzákladnější počítačovou terminologii.

Po technické stránce je vyžadován prohlížeč Microsoft Internet Explorer, Google Chrome, Mozilla Firefox nebo Opera (konkrétní testované verze viz příloha E). Je třeba zobrazovací zařízení, ideálně schopno zobrazovat barvy. Větší rozměry povedou k vyšší efektivitě práce, velmi vhodné je mít rozměry alespoň velikosti anotovaného obrázku.

4.6 Implementace

Tato sekce pojednává o konkrétní technologické implementaci výše popsaného rozhraní. Projekt nese označení „MarkArt“ a je takto v některých případech referován.

4.6.1 Metoda zobrazování

Dle zadání práce je na anotovaných fotografiích třeba zobrazit plochy či okraje jednotlivých značek. Je tudíž nutné vybrat jednu z několika metod umožňujících vizualizaci tvarů ve webové stránce. V této oblasti jde o **Adobe Flash** a dále nové technologie – **Canvas**, **SVG** a **CSS transformace**.

Adobe Flash je platforma pro tvorbu dynamických aplikací, používaných ve webových stránkách [2]. Na webu je tuto aplikaci možno spustit pomocí multiplatformního běhového prostředí **Adobe Flash Player**, při tvorbě aplikace je využíván jazyk **ActionScript**. Tuto aplikaci je nutno instalovat, avšak **Adobe Flash Player** vykazuje 99% pokrytí [3], tudíž je z pohledu vývojáře z hlediska rozšíření použití bezpečné.

CSS transform je pravidlo kaskádových stylů verze 3 umožňující přesunutí, změnu velikosti, rotaci anebo zkosení **HTML DOM** elementu [7]. **HTML** element je vždy obdélníkového tvaru, lze dosáhnout kružnice nebo obdélníku se dvěma vypouklými stranami pomocí vysokých hodnot **CSS** vlastnosti **border-radius** (viz implementace Google+ 2.2). K současnému datu 10. května 2012 jde o standard ve fázi „working draft“, což je první z pěti fází vývoje standardu **W3C** [10].

Canvas je nová technologie definovaná ve standardu **HTML5** [11] umožňující vykreslování objektů a bitmap na rastrové bázi (v případě 2d kontextu) do elementu **canvas** [7]. **Canvas** reprezentuje plochu primitivních obrazových bodů, nedisponujících žádnou logikou [23]. Při změně vykreslovaného obsahu je proto v mnoha případech plochu – případně její část – vymazat a vykreslit všechen obsah, nezměněný i nový. K současnému datu jde o standard ve fázi „working draft“.

Základem **Scalable Vector Graphics** jsou objekty reprezentované vektory. Tyto objekty lze parametricky nastavovat a měnit a jejich vykreslení se automaticky aktualizuje. Dokáží přijímat události, být skryty, seskupovány, transformovány, nést data atd. [23] K současnému datu 10. května 2012 jde o standard ve fázi „recommendation“, což je poslední z pěti fází standardu **W3C** [10].

Ze zmíněných technologií jsem zvolil **Scalable Vector Graphics**. V porovnání s **Adobe Flash** poskytuje podobné možnosti ve smyslu zobrazování tvarů [1][23], důvodem volby byl jazyk **JavaScript**, který je v prohlížečích standardně a je ho možno používat i pro interakci s **DOM** stromem a je tak možno použít jednotný programovací jazyk. V případě, že by se **Adobe Flash** využil i na ovládací prvky, nebylo by již možné ovlivňovat tyto prvky pomocí kaskádových stylů a byla by tak byla ztížena integrace projektu do konkrétních webových stránek vývojáře, používající toto rozhraní. Ve volbě mezi technologiemi popsanými **W3C** rozhodlo, že základní vektorové prvky povahou nejvíce odpovídají anotačním značkám, v návrhu nebyla stanovena potřeba práce na bázi jednotlivých pixelů. Dále je možné vykreslit

různé tvary (základem jsou kružnice, elipsy a obdélníky, další tvary je možné definovat pomocí cest). Díky parametrům a objektové povaze je jednodušší vytváření i manipulace s prvky [23]. K SVG jsou k dispozici prověřené knihovny. Jedná se standard výrazně nejbližší fázi finálního vydání.

4.6.2 JavaScript, CoffeeScript

Finálním jazykem práce, interpretovaným v internetovém prohlížeči, je JavaScript. Tento jazyk umožňuje reagovat na události a dynamicky měnit obsah zobrazených webových stránek. Jde o jazyk objektově orientovaný a slabě typovaný. Vznikl roku 1995 pod vedením Brendana Eich, původně pod názvem „LiveScript“, jako jednodušší jazyk pro vývojáře neseznámené s Javou a pro webové designery[15]. Přes svůj potenciál byl zpočátku pro nízkou bariéru vstupu, absenci kompilátoru a své nerestriktivní vlastnosti přehlížen jako jazyk pro začátečníky[15][16].

Syntakticky vychází z C, C++ a Javy. Je inspirován jazykem Self z hlediska mechanismu dědičnosti – mechanismu **prototypů**. Práce s poli a regulárními výrazy je pak více podobná Perlu [16].

Veškerá data jsou v JavaScriptu objektem. Čísla, pole, dokonce i funkce jsou objekty. Všechny nepřímo dědí od společného objektu `Object`[7].

Pro psaní zdrojového kódu jsem zvolil jazyk CoffeeScript. Jedná se o jazyk vytvořený v roce 2009 Jeremy Ashkenasem upravením JavaScriptu, do nějž se také následně překládá[13], a tudíž není třeba se zabývat kompatibilitou s interpretujícími zařízeními – s CoffeeScriptem do kontaktu nepřijdou¹. Je zaměřený na stručnější syntaxi a jednodušší zápis, odklánějící se od C nebo Javy směrem k Pythonu. Odstraňuje pro řadu případů kulaté či složené závorky, zjednodušuje a zkracuje zápis iterace, definice tříd² a funkcí, práci s poli aj.[13] Kompilovaný JavaScriptový kód je zaměřen na korektnost a efektivitu, díky tomu, že již není třeba řešit čitelnost těchto zápisů. Překlad byl prováděn oficiálním kompilátorem, který dokáže automaticky zpracovat změněný soubor ve hlídaném adresáři³. Kompilace je provedena tak rychle, že se ve vývojovém procesu z pohledu vývojáře vůbec neprojeví.

4.6.3 Moduly

Z hlediska přehlednosti a efektivity psaní zdrojového kódu je vhodné projekt dělit na více částí, jediný soubor může být nepřehledný a ztěžovat týmovou práci[18]. Pro co nejrychlejší načítání stránek je naopak výhodné pouze minimální počet požadavků na server [5][12]. Je navíc třeba se zabývat posloupností načítání souborů, aby byly před použitím již potřebné identifikátory a proměnné definovány. Řešením je AMD (Asynchronous Module Definition), které řídí načítání souborů (resp. modulů) včetně volby pořadí. Modul je v tomto projektu téměř vždy synonymem pro soubor i třídu. Deklaruje své závislosti, uvádí svoji definici a volitelně svůj název. [14]

Vybral jsem knihovnu **RequireJS** (autorem James Burke). Tato knihovna zaručuje načítání potřebných modulů ve správném pořadí [14], zamezuje cyklické (příp. rekurzivní) požadavky na načtení. Má (volitelně) integrovanou knihovnu jQuery (taktéž použitou v práci, viz 4.6.4). RequireJS spjata s optimalizačním nástrojem r.js od stejného autora (viz 4.6.9).

¹Je sice možné vložit do stránky element `script` s atributem `type=text/coffeescript`, avšak i ten je kompilován

²resp. implementace tradičního mechanismu tříd v beztržním JavaScriptu

³či podadresáři

Dovoluje také použití pluginů[14], v programu je použit plugin `text` (autorem taktéž James Burke), který umožňuje načítání i standardních textových souborů, jakými mohou být např. soubory kaskádových stylů.

4.6.4 Další knihovny

S výjimkou RequireJS, kterému byla věnována zvláštní pozornost z důvodu všeobecnosti a vlivu na celý kód, následuje popis a zdůvodnění použitých knihoven. Výpis přesných verzí lze nalézt v příloze B.

Raphaël (autorem Dmitry Baranovskiy) je knihovna zapouzdřující práci se SVG elementem a v něm obsaženými objekty. Knihovnu Raphaël jsem vybral z důvodu – oproti jQuery SVG – jednodušší API a přehlednější dokumentace.

jQuery je velmi populární [9] obecná knihovna pro JavaScript (původně od Johna Resiga). Umožňuje výběr DOM elementů v dokumentu pomocí identifikátorů se syntaxí velmi podobnou CSS selektorům. Zjednodušuje manipulaci s těmito elementy, procházení DOM struktury, práci s událostmi, využívání AJAXu a jiné[6]. V práci je používána pro vytváření a manipulaci s veškerým uživatelským rozhraním a jeho propojení s logickou částí aplikace pomocí událostí a pro AJAXové načítání a ukládání.

Knihovna **Underscore** je použita jako balíček pomocných metod pro práci s poli, objekty a funkcemi. Původně byla použita knihovna **SugarJS**, která ovšem modifikuje základní objekty JavaScriptu a mohla by tak potenciálně nastat kolize s jinými skripty, využívanými na stejné webové stránce jako uživatelské rozhraní, které je součástí této práce. Pro řešení byla knihovna Underscore rozšířena o metody pro práci s řetězci (`firstUpper`, `uniqChars`, `capitalize`, `stringSearch`) a poli (`duplicateObjectArray`, `combine`, `mapSort`, `valueSort`). Řada z byla vytvořena pro funkcionalitu pokročilého našeptávání, viz 4.4.4.

4.6.5 Třídy

MarkArt

Je iniciujícím a konfiguračním objektem, který spravuje nastavení celého systému a uchovává informace o anotovaném prvku a reference na primární DOM kontejnery.

Při své inicializaci přijme DOM elementy, ve kterých má být rozhraní vytvořeno, konfiguraci (více viz 4.6.7) a data. Obsahem dat jsou údaje o aktuálně anotované fotografii a volitelně také data potřebná k vykreslení a obsluze minigalerie (tj. odkazů na následující/předchozí fotografie). Pokud je konfiguračně povolena persistence nastavení, provede MarkArt načtení uloženého nastavení z `localStorage` a pokud je odstraněn (nebo je vyvolána událost `unload` na objektu `window`), provede uložení současného nastavení. Z tohoto vyplývá[7], že nastavení je uloženo pouze lokálně a při změně prohlížeče či zařízení bude jiné⁴. Nakonec provede spuštění Controlleru 4.6.5.

Při svém odstranění zároveň iniciuje řetězové ukončení přidružených objektů.

Uživatel⁵ skriptu nemusí přijít s objektem MarkArt ani do kontaktu, základní inicializace se děje přes objekt `markArtAccessor`. MarkArt je však zpřístupněn, stejně tak v něm obsažené objekty, tudíž je toto ponecháno na uživateli. Více viz 4.6.8.

⁴vyjma případů, kdy shodou okolností bylo i na tom odlišném zařízení při ukončení nastavení

⁵zde ve smyslu vývojáře používajícího toto řešení na webu, který vytváří či spravuje

Controller

Controller je centrální objekt, který řídí základní logické procesy aplikace a nakládání s daty. Obsahuje primárně metody pro vytváření a odstranění značek a tagů, správu načítání a ukládání dat (přes **ServerAccess** 4.6.5), spouštění nástrojů (viz 4.6.5) a zpřístupňuje další obslužné objekty jako **History** (4.6.5) a **MouseInput**. Dále řídí logování, spravuje selekci (podmnožina značek vybraná uživatelem, na které budou aplikovány jeho/její akce).

Pokud by byl projekt dále rozšiřován, **Controller** by byl prvním kandidátem pro dekomponování.

Tool a potomci

Nástroje (**Tools**) jsou skupina tříd pro interakci s užitelem. Jsou úzce spojeny s **Controller**em. Určují akce prováděné při kliknutí nebo tažení kurzoru po fotografii. Obsahují také nastavení (realizované objektem **ToolOptions**), které může ovlivňovat výsledky prováděné manipulace, přímo upravovat vybrané objekty nebo např. zobrazení scény.

Podskupinou jsou třídy dědící od **DraggingTool**, které manipulují vybranými značkami tažením kurzoru myši po fotografii (alternativně i pomocí klávesnice). Všechny z této skupiny dokáží interagovat s více vybranými značkami zároveň. Po dobu manipulace je nad SVG elementem vytvořen speciální DOM element, který deleguje všechny události na objekt fotografie (na nějž jsou navázány manipulační události), a tím zabraňuje nechtěné interakci s ostatními prvky (ovládací prvky značky, značky samotné) během procesu.

V jeden moment může být vybrán pouze jeden nástroj, souběžně jsou aktivní i další nástroje, všechny ze skupiny **universal**.

Nástroje nejsou výlučným prostředkem uživatele pro ovládání aplikace, další interakce je možná pomocí ovládacích prvků výpisu tagů, historie aj.

Nástroj zpřístupňuje metody:

- **initialize** volaná při aktivování daného nástroje
- **release** volaná při deaktivování daného nástroje
- **preRender** volaná před renderovací fází (viz 4.6.5)
- **postRender** volaná po renderovací fází (viz 4.6.5)
- **setOptions** volaná při změně nastavení
- objekty **control**, **imageWrapper**, **workspace** obsahující metody, jejichž názvy odpovídají názvu události, jež má ona metoda na daném objektu⁶ obsluhovat; událostí může být pro stejnou metodu více, pak se oddělují znakem „-“

Obslužné metody pro **control**, **imageWrapper** a **workspace** jsou navázány při aktivaci a při deaktivaci opět odpojeny. Stejně tak veškeré další obsluhy událostí na zmíněných DOM objektech, objektech nenáležících do kompetence **MarkArtu**.

⁶tj. **control** (DOM objekt SVG), **imageWrapper** (DOM element obalující fotografii a SVG), nebo **workspace** (DOM element obalující veškeré uživatelské rozhraní)

Mark, Entity

Mark je třída reprezentující konkrétní anotaci. Může jít o značku nebo i tag. Značka je charakteristická tím, že obsahuje referenci na vizuální **Entity**, nemusí mít (nemívá) vyplněnou vlastnost **text**. Objekt třídy **Mark** může obsahovat další zanořené objekty třídy **Mark**. Jinak jde o jednoduchý objekt, obsahuje dále pouze identifikaci, zda je o skutečný nebo navržený tag a zda je nebo není smazán.

Entity je vizuální reprezentace značky. Obsahuje definici jak konkrétně vytvořit a nastavit SVG (příp. DOM) elementy pro požadovanou podobu. Poskytuje metody pro skrývání a zobrazování, aktualizaci stavu, změnu velikosti, rotace, pozice, duplikování a jiné. Reaguje také na změny objektu **Mark**, ve kterém je zapouzdřena. Objekt **Mark** může mít přiřazenu nejvýše jednu **Entity**, avšak **Entity** může zapouzdřovat libovolnou kolekci elementů.

ServerAccess a dekorátory

ServerAccess je třída obstarávající veškerou komunikaci se serverem. Využívá knihovnu **jQuery** 4.6.4 pro realizaci komunikace samotné. Dovoluje navázání obslužných funkcí pro úspěšné či chybné dokončení požadavku. Požadavky i odpovědi umožňuje modifikovat či doplnit dodanými metodami, případně odeslání požadavku zrušit. Implementuje také funkcionalitu opakování požadavku, pokud nebyl dokončen úspěšně. Případně dokáže komunikovat přes pomocný proxy skript, běžící na lokálním serveru. Pro účely práce byl implementován jednoduchý skript pomocí jazyka PHP a knihovny **cURL**, podrobněji v C.2. V nastavení je vhodné uvést jak **URL** cílového serveru, tak **URL proxy**, aby mohla být **proxy** implementována flexibilně a přijmat cílovou adresu jako argument.

BulkServerAccess přidává funkcionalitu odložených požadavků pro aktualizace. Požadavek na aktualizaci zařadí do čekání a provede ho až v případě, že se neobjevil nový. Pokud nový požadavek přijde, čekající požadavek je zrušen a nahrazen tímto příchozím. Pokud jsou tedy data dokumentu rychle měněna (což nastává často při použití funkcionality hromadného vracení či obnovování akcí), byly by požadavky na změnu vydávány příliš rychle a také zbytečně. Před uzavřením tabu prohlížeče či prohlížeče celého v neuloženém stavu je uživatel chráněn, viz 4.4.3.

CachedServerAccess doplňuje **ServerAccess** o ukládání výsledků pro požadavky na data. Využívá k tomu objektu **Cache**. Ten má nastavenou expirační dobu, tudíž je možné, že k opětovnému požadavku na server během anotování dojde. Velmi přínosné je to při opakovaných rychlých požadavcích na již získaná data, což nastává často při využívání funkcionality hromadného vracení/obnovování akcí nebo při akci kopírování.

Renderer

Renderer má na starosti správu uživatelského rozhraní a správu vizuálních entit. Během své inicializace vytvoří a nastaví prvky rozhraní a připojí je do DOM modelu. Načte kaskádové styly a spustí vykreslovací smyčku.

Pomocí objektu **Internationalizer** vypisuje texty, časy a jiné údaje lokalizovaně.

Název „vykreslovací smyčka“ není zcela přesný, jelikož objekty SVG jsou vytvářeny jednorázově a pak měněny parametricky. V této smyčce jsou však skutečně volány

metody, které tyto objekty vytvoří. V dalších iteracích pak kontrolují stav entity a například provedou skrytí, pokud byla entita smazána, ale z nějakého důvodu není v daný moment skryta. Pro fáze před a po hlavním cyklu ve vykreslovací smyčce je možné registrovat funkce, které se mají v tyto časy spouštět. Toto je využito pro vykreslení a přesun kurzoru u nástroje pro tvorbu značek.

Hlavní funkce poskytované mimo inicializaci a smyčky jsou obnovení uživatelského rozhraní na základě dodaných dat a navázání příslušných obslužných událostí, nastavení aktuální zprávy o komunikaci se serverem, nastavení aktuální nápovědy a výpis entit.

History

Funkcionalita vrácení a obnovování akcí je implementována objektem **History**. Uchovává zásobník jednotlivých záznamů, při vrácení akce pak upravuje ukazatel na vrchol. Je možné nastavit maximální počet položek v jednom okamžiku v zásobníku. Základní rozhraní je následující:

- **happened** uloží záznam o proběhnuvší akci
- **undo** vrátí zadaný počet akcí zpět
- **redo** znovu provede zadaný počet akcí

Jednotlivý záznam má pak tyto vlastnosti:

- **name** název akce
- **actionDo** funkce, která při zavolání provede danou akci
- **actionUndo** funkce, která při zavolání navrátí aplikaci do předchozího stavu
- **identifier** identifikátor, přidáván do popisu jako doplňující informace (o akci či objektu, kterého se akce týká)

Mechanismus vrácení a obnovování akcí historie je založen na vlastnosti JavaScriptu, kdy funkce při své definici přejímá kontext, v němž byla vytvořena (lexikální uzávěr) [7]. Díky tomu mají **actionDo** i **actionUndo** vždy k dispozici prostředky a data, jak provést svoji činnost. Alternativně by bylo možno využít **Command pattern**⁷.

Pro odstranění duplicity existuje metoda **happen**, která po uložení navíc provede funkci **actionDo**, není tedy třeba popisovat akci dvakrát.

4.6.6 Datový formát

Základním formátem dat v MarkArt je JSON (JavaScript Object Notation), formát pro výměnu dat, založený na syntaxi JavaScriptu [7]. Je v něm nativně podporován, je stručný, řetězce jsou kódované pomocí **unicode** [17]. Skupina značek nebo tagů je reprezentována polem záznamů o následující struktuře:

- **id** id značky (pokud je přiřazeno serverem)
- **tags** pole názvů tagů

⁷ <<http://c2.com/cgi/wiki?CommandPattern>>

- **shape** název tvaru značky
- **color** pole složek barvy schématu RGBA
- **rotation** míra rotace v radiánech
- **scaleX** míra změny tvaru ve vodorovné ose
- **scaleY** míra změny tvaru ve svislé ose
- **center** objekt s vlastnostmi x,y typu float, označují střed objektu
- **boundaries** pole objektů s vlastnostmi x,y typu float, označují další body objektu (jejich interpretace je uložena v příslušném objektu typu **Entity** viz 4.6.5)

Použití MarkArtu ovšem nevynucuje užití tohoto formátu na serveru, data lze jednoduše modifikovat pomocí dvojice funkcí **remote.requestDataFilter** a **remote.responseFilter** a mít tak dva zcela odlišné datové systémy.

V tomto formátu jsou také vypisovány do elementu ovládacího rozhraní, zobrazujícího výpis dat značek. Data před výpisem je možno modifikovat funkcí **rendering.exportFilter** nebo **rendering.exportStringFilter**. Jako příklad byl implementován konvertor do formátu XML ve formě užívané anotačním programem VIA⁸.

Následující výpis je formátem dat o anotovaném obrazovém prvku samotném. Tato data jsou přijímána pouze při instanciaci a tudíž je případná modifikace ponechána na uživateli⁹ tohoto rozhraní.

- **id** identifikátor anotovaného prvku pro komunikaci se serverem
- **src** HTTP URI anotovaného prvku (obrázku)
- **description** (nepovinné) popis daného prvku, používán pro atribut alt (pokud již není vyplněn) a exportery

Nakonec je možno uvést data pro galerii jako pole o libovolném počtu objektů s těmito vlastnostmi:

- **id** identifikátor položky galerie pro porovnání s právě anotovaným
- **url** HTTP URI adresy, na kterou se má přesměrovat při kliknutí na položku galerie
- **thumb** HTTP URI náhledu pro danou položku
- **description** (nepovinné) popis dané položky, používán pro atribut alt

⁸Hlavní stránka na <<http://mklab.itι.gr/via/>> a ukázka formátu na <<http://mklab.itι.gr/via/files/ImageTest.XML>>

⁹zde ve smyslu vývojáře používajícího toto řešení na webu, který vytváří či spravuje

4.6.7 Nastavení a přizpůsobení

MarkArt má **značnou** možnost nastavení, nabízí v základu téměř 80 konfiguračních direktiv. Všechny z nich jsou volitelné. Podrobný popis každé direktivy lze nalézt v přiloženém souboru `konfigurace.pdf` (viz příloha A).

Konfigurace je rozdělena do několika sekcí. Většina odpovídá názvu třídy, ke které se nejvíce vztahují, přičemž nastavení nástrojů je uvedeno v části `controller`. Sekce `hooks` obsahuje funkce, které mohou reagovat na události vyvolané během vytvoření, činnosti nebo zakončení běhu MarkArtu. V položce `ajax` lze pak uvést veškeré direktivy z `<http://api.jquery.com/jquery.ajax/>`, rezervované jsou pouze `success` a `error`. Nejzákladnější nastavení pak v samostatné sekci nejsou, jsou uvedeny přímo v konfiguračním objektu. Z těchto nastavení je (pokud je povoleno) persistována část, konkrétně nastavení týkající se aktuálního stavu uživatelského rozhraní – aktuální položka menu, stav nastavení nástrojů, zapnutí/vypnutí automatického ukládání aj. Při znovunačtení je tak možné pokračovat přesně v bodě, kde uživatel přestal.

Nastavení vizuální stránky a rozložení je spravováno pomocí CSS. Doplněním vlastního CSS souboru, elementu nebo pomocí skriptování je tak možné upravit vzhled nebo rozmístění ovládacích prvků. MarkArt do základního elementu doplňuje třídy určující jazyk a kulturu. Tyto údaje jsou získány z konfigurace a pro položku `culture` jsou vypsány postupně všechny. U `language` je vypsán daný jazyk a pokud je více specifický (např. `en-uk-london`), jsou vypsány i všechny obecnější varianty (`en-uk` a `en`).

4.6.8 Accessor a použití

Vzhledem k asynchronní povaze načítání (viz 4.6.3) nelze přímo vrátit objekt `MarkArt`, jelikož v době interpretace nemusí být ještě dostupný. Z tohoto důvodu je pro přístup z vnější exportován objekt `markArtAccessor`, který zapouzdřuje volání `require` knihovny `RequireJS`. Interakci s vráceným objektem pak umožňuje nastavením přes nastavení své vlastnosti `initialized` na funkci. Pro stejný účel jde pak použít konfiguračních direktiv ze skupiny `hooks`. Dále poskytuje metodu pro zrušení předané instance a informace o verzi, autoru a poslední aktualizaci. Podrobnosti o použití jsou vypsány v příloze C.1.

4.6.9 Optimalizace

Součástí zpracování projektu jsou i některé formy optimalizace, viz tabulka 4.2. V práci je používán optimalizační nástroj `r.js` a varianta optimalizace `UglifyJS`. Je provedeno výrazné snížení počtu nutných požadavků na server a snížení celkové velikosti.

Druh zpracování	Velikost souborů ^a	Počet souborů
Žádná optimalizace	854 kB	45
Spojení souborů	762 kB	2
Předchozí + optimalizace <code>UglifyJS</code>	442 kB	2
Předchozí + <code>MOD_DEFLATE</code> ^b	132 kB	2

Tabulka 4.2: Výsledky provedených optimalizací

^avčetně knihovny `RequireJS`

^bnení součástí řešení, jedná se o prostředek populárního[8] HTTP webového serveru Apache, viz `<http://httpd.apache.org/docs/2.4/mod/mod_deflate.html>`

Kapitola 5

Testování

Důležitou součástí práce bylo také provedení uživatelského testování. Bylo prováděno ve dvou formách – primárně osobním uživatelským testováním s jednotlivými účastíky testování, sekundárně pomocí veřejně přístupné instance a k ní přidruženému dotazníku. Některá testování byla prováděna osobním testováním s vyplněním dotazníku.

Principy testování, jeho schéma, postupy a přístup vychází z [19].

5.1 Osobní testování

Jednalo se o osobní setkání s daným účastníkem. Forma byla zvolena z důvodu možnosti přímo vidět přesné reakce dané osoby, postupy které aplikuje, emoce vzešlé z rozhraní, možnost interagovat a dozvědět se zdůvodnění jeho úkonů, ujasnění názorů, zpětnou analýzu. Také opakování testování pro sledování změn, retence nebo naopak hladšího obsluhování. Tato forma tedy vykazuje prvky kvalitativního výzkumu [20].

Testování skutečnosti neodpovídalo úplně přesně, například tím, že účastníci byli skutečně jen testujícími osobami, a ne osobou, která má nějaký zájem na použití aplikace anotování. Dále v případě kontinuálního neúspěchu byla (po zaznamenání oné nastalé skutečnosti) poskytnuta pomoc, k čemuž by při reálném použití došlo jen nepravděpodobně. Reálně by to mohlo znamenat ztrátu uživatele a případně i šíření této negativní skutečnosti k ostatním potenciálním uživatelům.

Cílem osobního testování bylo získat pohled na aplikaci očima osoby neseznámené se samotným projektem a případně bez odborného vzdělání v oblasti informačních technologií či bez zkušeností s anotačními nástroji; vyhodnocení přehlednosti a efektivitu uživatelského rozhraní, obtížnost pochopení účelu a principů jednotlivých ovládacích prvků, náročnost pochopení procesů ke splnění jednotlivých úkonů a celkový dojem z rozhraní a jejího používání.

Testováno bylo iterativně, po každém testu byla provedena analýza a vyhodnocení, včetně analýzy videozáznamu (u testů, kde byl pořízen). Byly stanoveny závěry a projekt byl důkladně upraven tak, aby řešil zjištěné problémy a nedostatky. Samozřejmě byly objeveny chyby i technického rázu, které byly rovněž zaznamenány a následně opraveny.

5.1.1 Schéma jednotlivého testování

Účastníkům před sezením nebylo rozhraní známo a ve většině případů ani celková povaha či účel aplikace. Vždy šlo o jejich první používání aplikace. Případná obtížnost, nejasnosti či nepřehlednost rozhraní by totiž mohla být značně zmírněna či odstraněna časem stráveným

zkoušením aplikace. Rozhraní by však mělo být použitelné jednoduše a co nejdříve, bez nutnosti dlouhého zkoumání. Zkoušející osoba nebyla v průběhu naváděna, nebyly zodpovídaný dotazy, pokud nebylo očividné, že by sama úlohu ve stanoveném krátkém časovém rozmezí nevyřešila.

Testující osobě byly podány základní informace o následujícím průběhu a poté bylo načteno rozhraní. Osoba byla dotázána na první dojem z rozhraní. Následovaly specifitější dotazy na přehlednost, pochopení rozložení a prvků. Byla vybídnuta, aby si aplikaci po velmi krátký časový úsek vyzkoušela. Následně již byla požádána o provedení akcí v aplikaci, tj. vytvoření značky a přidání popisku (tagů). V dalším kroku následovala lehce složitější manipulace s úpravou vytvořených značek. Bylo vítané, pokud testující sám projevil iniciativu a zkoumal další funkce či procesy. Nakonec byl účastník dotázán na dojem z rozhraní a práce s ním a na jeho podněty či případné dotazy.

5.1.2 První testovací sezení

První testovací sezení proběhlo s uživatelem znalým v oblasti informačních technologií a práci s počítačem, avšak bez znatelných zkušeností s anotováním médií či dokumentů. Účastník si všiml a okamžitě pochopil princip funkce všech okolních (tj. zobrazených mimo obrázků) ovládacích prvků, pouze nezaznamenal možnost otevření boxu s informacemi a ovládacími prvky pro stav, načtení a uložení (dále „stavový box“). Podařilo se mu vytvořit značku v místě, na kterém zamýšlel. Při přidávání tagu nebyl úplně jasný vztah mezi našeptávanými výrazy a navrženými výrazy. Při manipulaci se značkami se snažil manipulovat tažením za samotný tvar značky, což v té době nebylo funkční. Po přečtení rady nad obrázkem způsob manipulace okamžitě pochopil a byl schopen uplatnit i u všech dalších nástrojů. Intuitivně zkusil použít klávesu `delete` pro smazání značky, což bylo implementováno a fungovalo. Úspěšně také vyzkoušel funkcionalitu vrácení a obnovení akce. U nástroje změny velikosti se projevila chyba, kdy nešlo tvar zmenšit, pouze zvětšit. Rozhraní hodnotil jako přehledné a dobře zpracované. Navrhl úpravu změny pozice výpisu tagů a galerie blíže k hornímu okraji stránky, u galerie případně do bočního panelu.

První testovací sezení jsem vyhodnotil jako relativně úspěšné. Opravil jsem nástroj na změnu velikosti. Našeptávací box jsem umístil bokem od ovládacího panelu značky, aby nezakrýval přidávané a navržené tagy, zvýšil minimální počet napsaných znaků k zobrazení z 1 na 2 a zvýšil dobu do zobrazení ze 100 ms na 200 ms. Není to finální řešení, jelikož se nezabývá úplně příčinou. Pokud by bylo chování nadále matoucí, bude nutná větší úprava nebo kompletní odstranění vlastnosti.

Byla upravena manipulace, tj. od té doby již bylo možné manipulovat se značkami i za jejich tvar či okraj tvaru. Nadto jsem doplnil chování, kdy při tažení za tvar či okraj je objekt manipulován, i když nebyl předtím vybrán (tímto se vybere). Stejně jako mechanismus tažení u tvorby značky využívá direktivu `controller.timeToDrag` pro rozlišení kliknutí a držení tlačítka. Výpis tagů náležících značce jsem předřadil výpisu tagů vztahujících se k celému obrázku. Přesun galerie do pravého panelu jsem neprovedl, jelikož oba prvky v něm mají povahu seznamu a mohou být (v závislosti na nastavení a provedených akcích) značně dlouhé, takže by se jednak umístění neustále měnilo. Argument, že by se mohlo dostat velmi daleko nelze použít, protože i v současném návrhu jsou poměrně vzdálené s argumentací, že v anotačním módu není změna anotovaného prvku primární akcí. Více viz 4.3.4.

5.1.3 Druhé testovací sezení

Ve druhém testu byla účastnice s nižšími znalostmi informačních technologií, na běžné úrovni práce s počítačem, bez znatelných znalostí anotačních nástrojů a některé základní terminologie. Pochopení funkce okolních ovládacích prvků proběhlo bez problémů a zorientování se v rozhraní v pořádku. Pouze tam byl pokus o použití nástroje Přesunout v situaci, kdy nebyla vytvořena žádná značka – zde jsem udělal chybu a o této skutečnosti nezjistil více informací. První nejasnost byla u způsobu tvorby značky, kdy jako primární byly považovány formulářové prvky `input` a následně zadány velice nízké hodnoty (také nepracoval dobře skript na předvyplnění daných polí, který toto mohl zmírnit). Poté ovšem byla úspěšně vytvořena značka za použití metody dvou kliknutí, včetně předchozího nastavení barvy a tvaru. Přes neznalost termínu „tag“ bylo vyplněno správné pole. Nebyla vzata v potaz možnost přidat tagů více. Účastníci chyběl nějaký prvek, který by potvrdil nebo zakončil proces vytvoření otagované značky. Následovalo několik kliknutí na tlačítko „přidat tag“, což mohlo být kvůli oné absenci jasnějšího zakončení, ne však nutně. Byla párkrát použita funkcionalita hromadné historie, zpočátku ne dobře, nebylo úplně jasné, zda označená akce bude vrácena, nebo zda na ni bude stav vrácen. Funkce malého menu v ovládacím prvku tagu byla pochopena, využity atributy `title`. Přesunutí značky proběhlo bez problému, mírné nesnáze u změny velikosti, rotace bez problémů. Nápo vědný pruh nad obrázkem nebyl příliš dobře postřehnut. Znak fajfky v jednotlivých `inputech` přidaných tagů byl popsán jako něco, na co by mohlo jít kliknout.

O něco méně úspěšný test, díky tomu bylo možné samozřejmě odhalit a vylepšit řadu aspektů rozhraní. Zatím ne zcela kompletně dořešena zůstává primárnost tvorby značek zadáním rozměrů. Po úpravě by již měly výchozí hodnoty indikovat, že je třeba použít vyšších hodnot rozměrů. Dále byl nastaven parametr `step`, to znamená že u prohlížečů podporujících typ `inputu number` bude hodnota měněna po 10 px (stále ale bude možné numericky zadat libovolnou hodnotu); nastaveno minimum na 0, maximum podle rozměrů. Dále byl do popisku přidán údaj o maximálním rozměru a uživatel tak bude moci porovnat vizuální velikost a zadávanou hodnotu a zjistit, že hodnota 2 je pro celou šířku obrázku 840 px velmi málo.

Především však byl upraven kurzor. Od této iterace již mnohem lépe reflektuje stav rozhraní. Pozicování nad značkou má stále kurzor typu `pointer`, značící možnost kliknout na prvek. Každý nástroj ale již definuje vlastní speciální kurzor, jenž se projevuje v oblastech přímo nad obrázkem. U nástroje tvoření tagů je to kříž, který by – v kombinaci s návodnou kružnicí či čtvercem – měl mnohem lépe navádět, že se může do plochy obrázku kliknout a provést tam akci. Nástroje přesunu a kopírování mají kurzor `move`, značící, že manipulace je provedena tažením v požadovaném směru. U nástrojů změny velikosti a rotace by technicky tento kurzor odpovídal také, avšak charakteristika vyznění úkonu mi subjektivně nepřišla zcela totožná s pohybem (například střed se u těchto manipulací nikdy nemění) a i ostatní aplikace používají v těchto případech jiný kurzor. Zvolil jsem tedy varianty, které byly – z nabídky standardně podporovaných ukazatelů – nejvíce podobné, a to `nw-resize` pro změnu tvaru a `ne-resize` pro rotaci.

Tento speciální kurzor je ovšem u nástrojů typu `DraggingTool` aktivní pouze v případě, že je vybrána značka. Pokud je výběr prázdný a manipulace by nic neměnila, bude kurzor typu `default`.

Dále byly upraveny texty, aby alespoň některé naznačily význam termínu tag. Byl změněn nadpis u nabídnutých tagů na text více upoutávající a snad i jasnější. K položkám

u funkcionality hromadného vrácení/obnovení akcí byly doplněny popisky, zobrazované při najetí myši, které mají naznačit, že i aktuálně vybraná akce bude zahrnuta do vrácení/obnovení. Po úpravě atributu `title` u tlačítek pro vrácení (resp. obnovení) jedné akce byl připojen popisek oné konkrétní akce, která bude po kliknutí na dané tlačítko zpracována.

Bylo přidáno tlačítko s textem **Dokončit**, které pouze zruší výběr daného tagu a skryje jeho ovládací prvek, ovšem účel je primárně psychologický, aby měl uživatel určité zakončení oné krátké posloupnosti úkonů.

Do nápovědného pruhu byla přidána výrazná ikonka, která by měla přitáhnout pozornost (ne však až tak, aby to bylo rušivé). Také byla přidána nápověda pro situaci, kdy má uživatel vybrán nástroj typu `DraggingTool` (viz 4.6.5), ale nemá vybranou žádnou značku, a tak tažení po obrázku nebude mít na žádnou vliv.

Nakonec bylo upraveno menu ve smyslu zneaktivnění manipulačních nástrojů v případě, že v dokumentu nejsou definované značky. Tlačítko pro přidání zadaného tagu již také není aktivní v případě, že daný `input` nemá vyplněný text. Znak fajfky v `inputu` přidáného tagu se již zobrazuje pouze, pokud uživatel má tento `input` vybrán.

5.1.4 Třetí testovací sezení

Třetího testu se účastnila stejná osoba z testu druhého po třech dnech. Velmi brzy byla schopna doplnit značku, přidat tag a vybavila si hned všechny možnosti úpravy tagu či značky. Využila tlačítka **Dokončit**. Nyní již vyzkoušela doplnění i druhého tagu, ovšem za použití tlačítka **Dokončit** namísto **Přidat další popisek (tag)**. Přítomnost kurzorů bez ohledu na to, zda byly vybrány značky či ne, byl matoucí. Ovládání nástroje **Změny tvaru** bylo stále nepřívětivé. U nástroje kopírování se pokoušela provést akci pomocí jednoho kliknutí, po zaznamenání kurzoru a nápovědy nad obrázkem (tentokrát si nápověd všimla) použila nástroj bez problému. Při vytváření další značky nastala potíž v tom, že nastavení u nástroje **Přidat novou značku** lze využít pro vytváření následující značky, ale i úprave stávající, selekce na nich ale není nijak zobrazena, tudíž není jasné, jestli se bude nastavovat nebo upravovat. Dále nastavení rozměrů, narozdíl od dvou předchozích, stávající značku neovlivňuje (nekonzistence). Nastala bohužel chyba při kombinování funkcionality nástroje **Kopírovat** a změny tvaru pomocí nastavení, kdy objekt nereagoval na příkaz rotace, po obnovení stránky opět fungovalo. Byl prozkoumán prezentační mód (nástroj **Prohlédnout**), včetně nastavení a vazby výpisu značek na vizuální reprezentaci značek.

Z uvedeného vyplývá, že je stále potřeba nemálo úprav. Tlačítko „Dokončit“ nyní způsobí přidání tagu vyplněného do formulářového políčka. Dalším vlastním testováním bylo zjištěno, že kurzor skutečně má daný tvar akce pouze pokud je vybrána značka, jednalo se o chybu při rušení výběru. Byl přidán výrazný upozorňující popisek k nastavení nástroje **Přidat novou značku**, upozorňující, že je momentálně vybrána značka a úpravou nastavení bude změněna. Také se nastaví text nápovědného pruhu, popisující danou skutečnost.

Bylo doplněno provázání mezi formulářovými poli pro nastavení velikosti značky a vybranou značkou. Bylo vylepšeno nastavování těchto polí podle vybrané značky a podle naposledy vytvářené značky. Byla opravena funkcionality změny tvaru značky a vlastní testy zatím (i v kombinaci s kopírováním) vycházejí dobře.

U našeptávání je nyní standardně napovídáno to, co uživatel skutečně napíše, tedy nejsou napovídány tagy načtené z databáze. Zdroj našeptávání se také smaže při zavření prohlížeče (výchozí nastavení).

5.2 Dotazník

Druhá testovací část sestává ze zpřístupnění uživatelského rozhraní – a celkové testovací instance, včetně databáze, dat – testující osobě a následného vyplnění dotazníku. Po získání dat byla provedena jejich analýza a generalizace a vyvozeny závěry. Jedná se tedy o kvantitativní výzkum [20].

Dotazník byl rozdělen na pět sekcí. Principiálně byl rozdělen na část obecného pochopení rozhraní a poté interakci a provádění úkonů, podle [19]. V první části bylo úvodní poděkování a scénář, který byl testovanému doporučen pro vyzkoušení si projektu. Druhá část byla zaměřena na celkový dojem a pochopení rozhraní a názor na prezentační mód. Třetí část byla zaměřena na primární akce, tj. vytváření značek a přidávání tagů (více viz 4.4.1). Čtvrtá část žádala velmi stručné informace o účastníkovi (věk a odborná znalost) a umožňovala připsat názor či postřehy v textové formě (tj. nikoli výběr z nabídnutých hodnot) a zda se vyskytly v průběhu interakce nějaké chyby (kdy testující samozřejmě nemusí poznat, že jde v určitém případě o chybu, či naopak může za chybu považovat nějaké standardní chování). Poslední sekce byla označena jako volitelná a obsahovala názor na specifitější interakce a doplňkové funkcionality, viz 4.4.2.

Dotazník byl rozčleněn pro větší přehlednost a bylo použito nadpisů pro informování vyplňujícího, že postupuje a že se blíží konci. Dotazy na osobu testujícího byly přesunuty ke konci, aby na začátku, kdy je ještě pozornost a energie, vyplnil ty náročné. Volitelné dotazy byly umístěny jako poslední, aby se dotazník jevil vizuálně kratší. Nebyl také použit zcela formální jazyk.

U téměř všech položek bylo možno volit od tří do pěti možných hodnot – zde hodnocení dané akce či aspektu. Byla zvolena stupnice tří hodnot jako dostatečná pro účely práce, konkrétně: „dobré“, „bez znatelných problémů“, „chaotické, problémové“. Nebyla uvedena výrazně pozitivní hodnota, jelikož je cílem nalezení aspektů rozhraní, jenž potřebují vylepšit či odstranit, kde hodnota „výborné“ by nepřinesla přidanou hodnotu. Zároveň byly vynechány hodnoty mezi středním stupněm a špatným stupněm, pokud je v aspektu problém, je třeba ho řešit každopádně.

Kompletní dotazník je k nahlédnutí v příloze D.

5.2.1 Výsledky

Testování přes dotazník se zúčastnili především zkušenější uživatelé. Výsledky nebyly zcela pozitivní, ale je díky nim možno stanovit slabé stránky rozhraní a stanovit, kam se zaměřit při vylepšování. Po vizuální stránce by podle průzkumu bylo lepší zaměřit se na střídmost a jednoduchost. U interakce pak v obecné rovině především na lepší intuitivnost. Velmi pozitivně byla hodnocena plynulost rozhraní a přehlednost zobrazení anotací v obrázku. Při vytváření nových značek nebyly zásadní problémy. Nebyl také výrazný problém s úpravami značek a doplňujícími funkcionalitami typu našeptávání. Lehce rozporuplné bylo rychlé menu (kde v jednom případě šlo o potíž technického rázu) a potom především nástroj **Zmenšit/zvětšit**, který byl obecně hodnocen jako spíše neintuitivní. Implementován je procentuálním zvětšením a změnou probíhající směrem od výchozího bodu (u kružnice střed, u obdélníku levý horní roh), byla očekávána změna velikost v absolutním měřítku ve směru tažení myši. Tento nástroj by tedy potřeboval principiální změnu.

Přesné výsledky jsou přiloženy v samostatných souborech, viz příloha A. Zajímavostí je, že alespoň některé otázky z volitelné sekce zodpověděli všichni účastníci.

Kapitola 6

Závěr

Ve své práci jsem se zabýval návrhem a implementací webového rozhraní pro poloautomatické anotování obrazových dat. V jazyku `CoffeeScript` za použití technologie `Scalable Vector Graphics` byl dokončen konfigurovatelný skript, který ve vybraném elementu vytvoří anotační rozhraní a dokáže komunikovat se serverem. Součástí řešení je i krátký skript umožňující komunikaci se vzdáleným serverem. Díky webové povaze je vytvořené řešení použitelné na více platformách.

V práci jsou uvedena obdobná řešení této úlohy, popsána stručná teorie uživatelských rozhraní a rozhraní ve webovém kontextu, podrobně popsán a zdůvodněn návrh a implementace rozhraní a nakonec rozbor provedených testování.

Vytvořené řešení považuji za částečně úspěšné. Vzniklo použitelné a otestované webové rozhraní pro anotování. Na základě dotazníku, komunikace s uživateli a i osobního názoru by řešení mohlo být více intuitivní a jednodušší. Do budoucna bych tedy pokračoval upravením rozhraní tak, aby požadovalo méně ovládacích prvků a zároveň bylo intuitivnější. Vzhledem k současným trendům by bylo také vhodné se více zaměřit na zařízení s menším displayem a dotykovým ovládáním. Při všech formách vylepšení by jistě bylo vhodné spolupracovat s nástrojem pro automatické testování webových rozhraní, řešeným jako bakalářská práce na Fakultě Informačních Technologií VUT Brno. Řešení bylo přijato bylo obecně pozitivně a bylo ovládáno bez zásadních problémů i neodbornými uživateli, což považuji za významný aspekt pro zhodnocení výsledku práce.

Literatura

- [1] Adobe Systems Incorporated. *ActionScript 3.0 Developer's Guide: Using the drawing API*. [cit. 2012-05-05].
- [2] Adobe Systems Incorporated. Adobe Flash Platform. <http://www.adobe.com/cz/flashplatform/>. [cit. 2012-05-05].
- [3] Adobe Systems Incorporated. Adobe Flash Player. <http://www.adobe.com/cz/products/flashplayer.html>. [cit. 2012-05-05].
- [4] CERTH-ITI. Video Image Annotation Tool: Image Annotation Guide. <http://mklab.iti.gr/via/node/20>. 2009-10-22 [cit. 2012-06-06].
- [5] Google Incorporated. Web Performance Best Practices. https://developers.google.com/speed/docs/best-practices/rules_intro. 2012-03-28 [cit. 2012-04-19].
- [6] The jQuery Foundation. jQuery API. <http://api.jquery.com/>. 2012-04-19 [cit. 2012-04-22].
- [7] The Mozilla Foundation. Mozilla Developer Network. <https://developer.mozilla.org/>. 2012-04-12 [cit. 2012-04-19].
- [8] W3Techs. Usage of web servers for websites. http://w3techs.com/technologies/overview/web_server/all. 2012-04-04 [cit. 2012-04-26].
- [9] W3Techs. Usage Statistics and Market Share of JavaScript Libraries for Websites, April 2012. http://w3techs.com/technologies/overview/javascript_library/all. 2012-04-20 [cit. 2012-04-21].
- [10] World Wide Web Consortium. W3C Process Document: W3C Technical Report Development Process. <http://www.w3.org/2005/10/Process-20051014/tr.html>. 2005-10-14 [cit. 2012-04-19].
- [11] World Wide Web Consortium. HTML5: A vocabulary and associated APIs for HTML and XHTML. <http://www.w3.org/TR/html5/>. 2012-03-29 [cit. 2012-05-05].
- [12] Yahoo Incorporated. Exceptional Performance. <http://developer.yahoo.com/performance/>. [cit. 2012-04-19].
- [13] ASHKENAS, J. CoffeeScript. <http://coffeescript.org/>. 2012-04-10 [cit. 2012-04-19].

- [14] BURKE, J. RequireJS. <http://requirejs.org/>. 2012-04-10 [cit. 2012-04-24].
- [15] CHAMPEON, S. JavaScript: How Did We Get Here?.
http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html.
2001-06-04 [cit. 2012-04-22].
- [16] FLANAGAN, D. *JavaScript: The Definitive Guide*. Definitive Guide Series. O'Reilly. 2006. ISBN 9780596101992.
- [17] Introducing JSON. <http://json.org/>. 2012-04-24 [cit. 2012-04-28].
- [18] KOLÁŘ, D. Principy programovacích jazyků a objektově orientovaného programování IPP – I. Vysoké Učení Technické v Brně, Fakulta Informačních Technologií. Říjen 2006. verze 1.1.
- [19] KRUG, S. *Don't Make Me Think! A Common Sense Approach to Web Usability*. Peachpit. druhé vydání. 2006. ISBN 0-321-34475-8.
- [20] OLECKÁ, I., IVANOVÁ, K. Metodologie vědecko-výzkumné činnosti.
http://www.mvso.cz/Files/WEB/APSYS/41Metodologie_vedecko-vyzkumne_cinnosti.pdf.
2010. ISBN 978-80-87240-33-5.
- [21] OSSENBRUGGEN, J. V., TRONCY, R., PAN, J. Z., aj. Image Annotation on the Semantic Web. Online. World Wide Web Consortium. Srpen 2007.
- [22] SHNEIDERMAN, B., PLAISANT, C. *Designing the User Interface*. Addison-Wesley. páté vydání. 2010. ISBN 0-321-53735-1.
- [23] SUCAN, M. SVG or Canvas? Choosing between the two.
<http://dev.opera.com/articles/view/svg-or-canvas-choosing-between-the-two/>.
2010-06-16 [cit. 2012-04-24].
- [24] WENYIN, L., DUMAIS, S., SUN, Y., aj. Semi-Automatic Image Annotation. In *Human-computer Interaction: INTERACT '01 : IFIP TC.13 International Conference on Human-Computer Interaction, 9th-13th July 2001, Tokyo, Japan*. IOS Press. 2001. ISBN 1586031880. s. 326–333.
- [25] ZEMČÍK, P. Tvorba uživatelských rozhraní. Vysoké Učení Technické v Brně, Fakulta Informačních Technologií. Listopad 2006.

Příloha A

Obsah CD

Součástí práce je CD, obsahující následující položky:

bp.pdf technická zpráva bakalářské práce ve formátu .pdf

demonstrace.avi demonstrační video

dokumentace.pdf popis instalace, použití a výpis všech konfiguračních direktiv

vysledky_dotazniku.xls kompletní výpis výsledků dotazníku ve formátu .xls (Microsoft Excel)

vysledky_dotazniku.pdf kompletní výpis výsledků dotazníku ve formátu .pdf

doplňky/

index.php příklad použití implementovaného rozhraní

remoteServer.php příkladová implementace vzdáleného serveru v PHP

proxy.php pomocný PHP skript pro přesměrovávání komunikace na/ze vzdáleného serveru

zdroj_text složka obsahující zdrojové texty a data technické zprávy

zdroj_kod složka obsahující zdrojové programové části práce

priklad_pouziti složka obsahující řešení k vyzkoušení, včetně serveru a dat

Příloha B

Detaily použitých knihoven

- *RequireJS* v1.0.2
- *Raphaël* v2.1.0¹
- *jQuery* v1.7.1
- *jQueryUI* v1.8.1
- *jQuery Cookie*²
- *Underscore.js* v1.3.1
- *r.js* v1.0.7

¹lehce upraveno pro kompatibilitu s RequireJS

²verze neuvedena, jde o commit `faa09dc38b`

Příloha C

Manuál

C.1 Použití z hlediska vývojáře

1. Do složky (příp. složek) přístupných serveru přesuneme soubory `require.js`, `markart.js` a pokud chceme, tak i soubory obrázků
2. V dokumentu HTML, kde bude skript používán, můžeme využít stávající strukturu nebo vytvořit nové elementy, kde vnější bude hrát roli kontejneru pro celé rozhraní (označované jako `workspace`), vnitřní pak obalovací prvek pro obrázek (`imageWrapper`) a uvnitř pak element `` (případně jiný, podle nastavení `element`)
3. Do hlavičky HTML dokumentu přidáme odkazy na skripty

```
<script type="text/javascript" src="$path$/require.js"></script>
<script type="text/javascript" src="$path$/markArt.js"></script>
```

kde `$path$` nahradíme skutečnou cestou ke skriptům. Cesta k obrázkům je nastavena v kroku 4 konfigurační direktivou `assetsPath`.

4. poté už můžeme rozhraní iniciovat pomocí

```
<script type="text/javascript">
markArt.run(containers, contents, config, immediately)
</script>
```

kde `containers` je objekt s prvky `workspace` a `imageWrapper`, obsahující řetězec ve formátu CSS selektoru nebo objekt `jQuery`, viz krok 2. Argument `contents` je objekt s prvky `current`, obsahující data o aktuálně anotovaném obrázku, a volitelně s `gallery`. Formát těchto dat je popsán podrobně v 4.6.6. Položka `config` je objekt s konfiguračními direktivami, popsány v příloženém souboru `konfigurace.pdf` (viz A), obecně o nastavení pak v 4.6.7. Poslední položka `immediately` je typu `bool`. Standardně metoda `markArtAccessor.run` čeká se spuštěním až na připravenost DOMu, pokud ale uživatel chce volat instanci okamžitě, zadá hodnotu `true`.

Pokud chceme provést určité úkony s nově vzniknuvší instancí (nebo jen v reakci na tuto událost), buď přiřadíme do `markArtAccessor.initialized` funkci (tato funkce bude pak volána pro každou instanci MarkArt, nehledě na její konfiguraci, dokud

nebude přepsána), nebo použijeme konfigurační direktivy `hooks.initialized` nebo `hooks.loaded`.

Toto je důležité, neboť kvůli asynchronní povaze 4.6.3 nelze z metody `markArtAccessor.run` přímo vrátit objekt rozhraní.

Příklad použití, využívaný při testování tohoto rozhraní, je k přiložen na CD, viz příloha A.

C.1.1 MarkArtAccessor

Objekt `markArt` exportovaný skriptem `markArt.js` je ve skutečnosti objektem `markArtAccessor`, objekt třídy `MarkArt` je zapouzdřen uvnitř. `markArtAccessor` má následující vlastnosti:

- `version` číslo verze datového formátu float
- `versionString` textová reprezentace verze
- `author` stručně o autorovi
- `lastUpdated` datum poslední změny, formát YYYY-MM-DD
- `instances` pole instancí objektů typu `MarkArt`
- `initialized` metoda spouštěná po každé instanciaci objektu `MarkArt` přes tento objekt

a tyto metody:

- `run` vytvoří novou instanci `MarkArt`, viz krok 4 použití.
- `destroy` zruší instanci `MarkArt`, uvedenou jako argument volání této metody

C.2 Ukázkový server a proxy

Pro účely vývoje a testování rozhraní byl vyvinut persistenční a napovídací server (soubor `remoteServer.php`) a proxy server (`proxy.php`), obojí psané jazykem PHP.

Skript `proxy.php` byl vytvořen kvůli „Same origin policy“, kdy se nelze dotazovat na zdroje z jiné domény, protokolu nebo portu [7]. Příjímá i předává data zároveň z POST i GET (pořadí dle nastavení serveru, viz [<http://www.php.net/manual/en/ini.core.php#ini.variables-order>](http://www.php.net/manual/en/ini.core.php#ini.variables-order)), adresu cílového serveru získává z klíče `finalUrl`.

Implementovaný vzdálený server řeší napovídání tagů pomocí asociačního pole `zadaný tag → doporučené tagy`, bere v potaz diakritiku, ignoruje velikost písmen. Persistence je řešena jednoduše textovým souborem a (de)serializací formátu JSON.

Oba jsou přiloženy na CD, viz příloha A.

C.3 Výchozí klávesové zkratky

Nástroj Prohlédnout v

Nástroj Přidat novou značku t

Nástroj Přesunout m

Nástroj Zmenšit/zvětšit s

Nástroj Rotovat r

Nástroj Kopírovat c

Vytvořit značku shift + c

Náhled p

Zrušení akce escape

Vrácení akce ctrl + z

Znovuprovedení akce ctrl + y

Odstranění značky delete

Vybrat vše a

Vybrat nic escape

Invertovat výběr i

Vybrat posledně editovanou/přidanou značku d

Vybrat následující ctrl + pravá šipka

Vybrat předcházející ctrl + levá šipka

(De)aktivovat automatické ukládání F7

Manuálně (vynuceně) uložit F8

Akce nástroje 'vlevo' šipka vlevo

Akce nástroje 'vpravo' šipka vpravo

Akce nástroje 'nahoru' šipka nahoru

Akce nástroje 'dolů' šipka dolů

Příloha D

Dotazník

Web Portal for Annotation of Visual Data / Webový portál pro anotaci obrazových dat

Thank you for helping me out! / Díky za pomoc!

URL: [ENG] <http://ibp_test.noxart.cz/> [CZ] <http://ibp_cz.noxart.cz/?lang=cz&id=1>

If you have a comment towards this questionnaire itself please let me know / Pokud máte připomínku přímo k tomuto dotazníku, budu rád když dáte vědět

Recommended scenario / doporučený scénář:

[ENG]

Create a mark and try to add tags (say person, man, friend). Another mark of different shape and color - try using a different way of using mouse as well - say person, woman, unknown. Try to create yet another mark just by using a keyboard. Add a tag for the whole image. Try changing a tag of the second mark from "unknown" to "happy". Select multiple marks and delete them. (You are surely free to experiment some more)

Thank you!

[CZ]

Vytvořte značku a zkuste přidat tagy (např. člověk, muž, kamarád). U další značky s jiným tvarem a barvou - zkuste použít odlišný způsob interakce myši - třeba člověk, žena, neznámá. Další značku zkuste prosím vytvořit čistě pomocí klávesnice. Zkuste přidat tag pro celý obrázek. Zkuste změnit tag u druhé značky z "neznámá" na "šťastná". Vyberte více značek naráz a smažte je. (Určitě můžete i dále experimentovat)

Díky!

How do you rate this user interface visually / Jak hodnotíte rozhraní po vizuální stránce *

Clarity / srozumitelnost

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Readability / přehlednost

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

View mode legibility, style / Styl a přehlednost módu prohlížení

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Rate the interaction / ohodnoťte interakci *

Smoothness / plynulost

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Simplicity / jednoduchost

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Intuitivness / intuitivnost

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Correction, modification / Korekce, úpravy

manual correction, undo, resizing, moving... / manuální korekce, vrácení akce, změna velikosti, přesunutí...

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Almost there... / Téměř hotovo...

Rate the mark creation / Ohodnotte tvorbu značek *

(the shape / tvaru)

Two distinct clicks / dvě jednotlivá kliknutí

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

- not discovered / neobjeveno
- not used / netestováno

Dragging / tažení

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not discovered / neobjeveno
- not used / netestováno

Shift + click / shift + kliknutí

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not discovered / neobjeveno
- not used / netestováno

Rate the tag adding / Ohodnotte přidávání tagů *

Adding a tag / přidání tagu

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not used / netestováno

Tag editing / editace tagu

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not used / netestováno

Adding suggested tags / přidávání navržených tagů

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not used / netestováno

Finish! A jsme u konce!

Your age / váš věk *

- 12-17
- 18-23
- 24-35
- 36-45
- 46-60
- 61-70
- 71+

Your computer knowledge / Vaše znalosti práce s počítačem *

- beginner / začátečník
- basic knowledge / základní znalosti
- average user / průměrný uživatel
- power user / pokročilý uživatel
- IT skilled / znalý v IT

Any note / jakákoli připomínka

What you liked, disliked or just was distinct / Co se Vám líbilo, nelíbilo, co nešlo přehlédnout...

plná textová odpověď

Any bug occurred? / vyskytla se chyba?

plná textová odpověď

Optional / nepovinné

Other features / Další funkčnosti

Autocomplete for tags (yellow box) / Napovídání při tvorbě tagů (žlutý blok)

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not discovered / neobjeveno

Shortcuts / klávesové zkratky

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not discovered / neobjeveno

Quick menu (right mouse in the picture) / Rychlé menu (pravé kliknutí do obrázku)

- pleasant / dobré

- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový
- not discovered / neobjeveno

Rate the interaction / ohodnoťte interakci

Selection / označování

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Mark editing by options / úprava značky nastavením

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Manipulation - move / manipulace - přesunutí

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Manipulation - scale / manipulace - změna velikost

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Manipulation - rotate / manipulace - rotace

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Manipulation - copy / manipulace - kopírování

- pleasant / dobré
- not a noticeable problem / bez znatelného problému
- chaotic, problematic / chaotický, problémový

Dotazník byl vytvořen v Google DOCS <<http://docs.google.com/>>.

Příloha E

Testované prostředí

E.1 Klientská část

Klientská část programu byla testována v programech:

- Google Chrome 18.0.1025.162, platforma Windows 7 64bit
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.162 Safari/535.19
- Google Chrome 18.0.1025.168, platforma Ubuntu 11.0 64bit
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.168 Safari/535.19
- Google Chrome 18.0.1025.168, platforma Ubuntu 12.04 32bit
- Opera 11.62 sestavení 1347, platforma Windows 7 64bit
Opera/9.80 (Windows NT 6.1; WOW64; U; en) Presto/2.10.229 Version/11.62
- Opera 12.00 beta sestavení 1387, platforma Windows 7 64bit
Opera/9.80 (Windows NT 6.1; Win64; x64; U; Edition Next; en) Presto/2.10.289 Version/12.00
- Mozilla Firefox 11.0, platforma Windows 7 64bit
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:11.0) Gecko/20100101 Firefox/11.0 FirePHP/0.7.1
- Mozilla Firefox 12.0, platforma Windows 7 64bit
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.0) Gecko/20100101 Firefox/12.0 FirePHP/0.7.1
- Mozilla Firefox 12.0, platforma Ubuntu 11.0 64bit
Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:12.0) Gecko/20100101 Firefox/12.0
- Microsoft Internet Explorer 9.0.8112.16421 (64-bit edice), platforma Windows 7 64bit
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)

E.2 Serverová část

Příložené testovací a pomocné PHP skripty byly testovány na PHP interpretu verze 5.3.9.