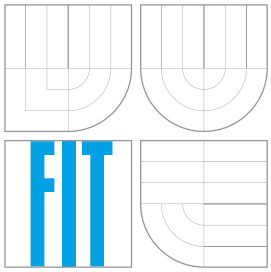
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SÉMANTICKÁ ANOTACE TEXTU

SEMANTIC ANNOTATION OF TEXT

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE
AUTHOR

Ing. JAROSLAV DYTRYCH

ŠKOLITEL
SUPERVISOR

Doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2016

Abstrakt

Tato práce se zabývá inteligentními systémy pro podporu sémantického anotování textu. Rozebírá motivaci vzniku takových systémů a stav poznání v oblastech souvisejících s jejich používáním. Popisuje také nově navržený a implementovaný anotační systém, realizující unikátním způsobem pokročilé funkce sémantického filtrování a prezentace alternativ anotací. Výsledky provedených experimentů jasně ukazují výhody zpracovaného řešení. Demonstrují také, že uživatelského rozhraní anotačních nástrojů významně ovlivňuje proces anotování. Následně je provedena optimalizace zobrazovaných informací pro úlohu desambiguace mnohoznačných jmen a jsou experimentálně vyhodnoceny přínosy metod navržených pro zvýšení rychlosti anotování a kvality vytvářených anotací. Srovnáním s obecným nástrojem Protégé je rovněž ukázán přínos vytvořeného systému pro kolaborativní tvorbu ontologií, které mají být "ukotveny" v textu. Na závěr jsou analyzovány a shrnuty veškeré dosažené výsledky.

Abstract

This thesis deals with intelligent systems for support of the semantic annotation of text. It discusses the motivation for creation of such systems and state of the art in the areas of their usage. The thesis also describes newly proposed and realised annotation system which realizes advanced functions of semantic filtering and presentation of annotation suggestion alternatives in a unique way. The results of finished experiments clearly show the advantages of proposed solution. They also prove that the user interface of the annotation tools affects the annotation process. The optimisation of displayed information for the task of disambiguation of ambiguous entity names was done and proposed methods to speedup and increase of quality of the created annotations was experimentally evaluated. The comparison with the Protégé general tool has proven the benefits of created system for collaborative ontology creation which should be anchored in the text. In the conclusion, all achieved results are analysed and summarized.

Klíčová slova

sémantické anotace, strukturované anotace, protokol pro přenos anotací, srovnání anotačních nástrojů, W3C Web Annotation, W3C Open Annotation, nabízení anotací, alternativní nabídky anotací, sémantické filtrování, vytváření ontologií na základě textu

Keywords

semantic annotations, structured annotations, protocol for annotation transmission, comparison of annotation tools, W3C Web Annotation, W3C Open Annotation, annotation suggestions, alternative suggestions of annotations, semantic filtering, ontology creation based on text

Citace

DYTRYCH, Jaroslav. *Sémantická anotace textu*. Brno, 2016. Disertační práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Školitel Pavel Smrž.

Sémantická anotace textu

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně pod vedením pana docenta Pavla Smrže. Další informace mi poskytli spolupracovníci z evropských projektů Decipher a MixedEmotions. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jaroslav Dytrych
29. srpna 2016

Poděkování

Děkuji Doc. RNDr. Pavlu Smržovi, Ph.D., za odborné vedení, vstřícnost a ochotu při řešení této práce. Děkuji také své rodině za podporu a trpělivost.

Obsah

1 Úvod **5**

2 Cíle práce **7**
2.1 Srovnání uživatelských rozhraní anotačních nástrojů . . . 7
2.2 Vytváření strukturovaných anotací . . . 7
2.3 Anotování kdekoliv a kdykoliv . . . 8
2.4 Zvýšení kvality vytvářených anotací . . . 8
2.5 Formalizace znalostí na základě anotování textu . . . 9

3 Stav poznání v oblasti tématu práce **10**
3.1 Definice pojmu anotace . . . 10
3.2 Kolaborativní strukturování znalostí . . . 11
3.3 Existující řešení pro anotování . . . 14

4 Zvolené metody řešení **21**
4.1 Metodika srovnávání anotačních nástrojů . . . 21
4.2 Anotační systém . . . 23
4.2.1 Architektura . . . 24
4.2.2 Formát anotace . . . 25
4.2.3 Protokol pro přenos anotací . . . 26
4.2.4 Server pro práci s anotacemi . . . 27
4.2.5 Klienti . . . 27
4.2.6 Nabízení anotací . . . 29
4.3 Zvýšení kvality vytvářených anotací . . . 29
4.4 Kolaborativní tvorba ontologií . . . 30

5 Realizovaný anotační systém **32**
5.1 Architektura . . . 32
5.2 Server pro práci s anotacemi . . . 33
5.3 Klienti . . . 36
5.4 Formát anotace . . . 43
5.5 Protokol pro přenos anotací . . . 48
5.6 Pokročilé vlastnosti systému . . . 50
5.7 Případy použití . . . 55
5.7.1 Nové způsoby publikování . . . 55
5.7.2 Příprava výstav v projektu Decipher . . . 56
5.7.3 Rozšiřování ontologie . . . 57
5.7.4 Správa anotovaných korpusových dat . . . 62

6 Provedené experimenty **64**
6.1 Testování prvotního prototypu systému . . . 68
6.2 Nasazení v projektu Decipher . . . 70
6.3 Výběr textů pro pokročilé experimenty . . . 72
6.4 První série pokročilých experimentů . . . 73
6.4.1 Srovnání nástrojů . . . 74
6.4.2 Optimalizace množství zobrazovaných informací . . . 78
6.4.3 Zobrazení alternativních nabídek anotací . . . 81
6.4.4 Sémantické filtrování . . . 83
6.5 Druhá série pokročilých experimentů . . . 83
6.5.1 Přínos sémantického filtrování . . . 84
6.5.2 Alternativní nabídky anotací . . . 85
6.6 Rozšiřování ontologie . . . 87

7 Dosažené výsledky a jejich analýza **89**
7.1 Protokol pro přenos anotací a formát anotace . . . 89
7.2 Vytvořený anotační systém . . . 90
7.3 Vyhodnocení srovnání anotačních nástrojů . . . 91
7.4 Určení optimálního množství zobrazovaných informací . . . 92
7.5 Přínos alternativních nabídek anotací . . . 93
7.6 Přínos sémantického filtrování . . . 94
7.7 Kolaborativní vytváření ontologie . . . 95
7.8 Souhrnné výsledky . . . 95

8 Závěr **97**

Literatura **100**

Přílohy **113**
Seznam příloh . . . 114

A Srovnání anotačních nástrojů **116**

B Srovnání pokročilých anotačních nástrojů **131**

C Specifikace formátu anotace 4A **133**

D Specifikace 4A protokolu verze 1.1 **137**
D.1 Správa sezení . . . 138
D.2 Uživatelé a skupiny uživatelů . . . 139
D.2.1 Řízení odběru anotací . . . 140
D.2.2 Synchronizace dokumentu . . . 141
D.2.3 Přenos typů anotací . . . 143
D.3 Přenos anotací . . . 146
D.4 Nabízení anotací . . . 146
D.5 Podpora kontrolovaného slovníku . . . 148
D.6 Přenos nastavení . . . 149
D.6.1 Chyby a varování . . . 150
D.7 Potvrzení bez zaslání dat . . . 155

E Zjednodušený příklad komunikace protokolem v. 1.1 **156**

F Specifikace 4A protokolu verze 2.0 **160**
F.1 Komunikace . . . 160
F.1.1 Synchronní kanál . . . 160
F.1.2 Asynchronní kanál . . . 162
F.2 Sezení a přihlášení uživatele . . . 162
F.2.1 Navázání spojení . . . 162
F.2.2 Ukončení spojení . . . 164
F.2.3 Přihlášení uživatele . . . 164
F.2.4 Odhlášení uživatele . . . 165
F.3 Uživatelé a uživatelské skupiny . . . 165
F.3.1 Seznamy uživatelů . . . 166
F.3.2 Seznamy skupin uživatelů . . . 168
F.3.3 Vstup uživatele do skupiny . . . 169
F.3.4 Odchod uživatele ze skupiny . . . 169
F.4 Odběry anotací . . . 169
F.4.1 Vytvoření odběru . . . 170
F.4.2 Rušení odběru . . . 171
F.4.3 Modifikace odběru . . . 171
F.4.4 Seznamy odběrů . . . 171
F.4.5 Přihlášení se k odběru . . . 172
F.4.6 Odhlášení se z odběru . . . 173
F.5 Synchronizace dokumentu . . . 173
F.5.1 Proces synchronizace . . . 173
F.5.2 Znovuposlání obsahu dokumentu . . . 176
F.5.3 Modifikace dokumentu . . . 177
F.6 Typy . . . 180
F.6.1 Jednoduché typy . . . 180
F.6.2 Formát typu anotace . . . 182
F.6.3 Manipulace s typy anotací . . . 184
F.6.4 Získání typů anotací od serveru . . . 185
F.6.5 Získání atributů z ontologie . . . 185
F.7 Anotace . . . 186
F.7.1 Formát anotace . . . 186
F.7.2 Manipulace s anotacemi klienta . . . 191
F.7.3 Vytvoření anotace klientem . . . 192
F.7.4 Modifikace a rušení anotace klientem . . . 193
F.7.5 Znovuzaslání anotací . . . 193
F.8 Návrhy anotací . . . 193
F.8.1 Manipulace s návrhy . . . 193
F.8.2 Získání návrhů . . . 194
F.8.3 Potvrzení návrhu . . . 195
F.8.4 Odmítnutí návrhu . . . 197
F.9 Kontrolovaný slovník . . . 197
F.9.1 Získání typů entit . . . 198
F.9.2 Získání entit . . . 198
F.10 Nastavení . . . 200

F.10.1 Přenos nastavení ze serveru 200
F.10.2 Změna parametrů 201
F.11 Chyby a varování 201
F.11.1 Chybová zpráva 201
F.11.2 Varovací zpráva 202
F.11.3 Seznam chybových kódů 202
F.11.4 Seznam kódů varování 213

G Integrace v systému z projektu Decipher **215**

Kapitola 1

Úvod

I přes velkou snahu o formalizování znalostí je stále většina textových informací dostupná pouze ve formě textu v přirozeném jazyce. Dosáhnout strojového porozumění takovému textu je velmi obtížné, což mimo jiné ztěžuje realizaci původní vize sémantického webu. Nejmodernější nástroje pro extrakci informací z textu jsou omezené pouze na úzké sady případů, pro které byly trénovány. V ostatních případech je stále potřebné manuální či poloautomatické vytváření metadat. V této práci se proto zaměřuji na jeden z přirozených způsobů formalizace znalostí – anotování textu.

Existuje celá řada anotačních nástrojů umožňujících vytváření jednoduchých textových poznámek, případně odkazů na referenční zdroje popisující zmíněné entity. Pokročilejší nástroje umožňují i strukturování anotací pomocí odkazů mezi nimi či pomocí atributů. Tato funkcionalita je však podporována pouze v omezené míře. Navíc jsou příslušné nástroje často úzce zaměřeny na konkrétní případy užití a pro obecnější úlohy je nelze použít vůbec, případně jen s mnohými omezeními.

Studií, které se zabývají srovnáním nástrojů pro manuální či poloautomatické anotování textu z uživatelského hlediska a možnostmi jejich využití, existuje překvapivě málo. Většina srovnání se totiž zaměřuje na nástroje pro automatické generování nabídek anotací v pozadí, které je však u řady moderních nástrojů možné jednoduše nahradit a pro určení možností samotných anotačních nástrojů je tak toto srovnání irelevantní. Existující studie zaměřené na uživatelské rozhraní a využitelnost nástrojů jsou navíc často zastaralé a neposkytují dostatek informací pro určení možností strukturování anotací a nasazení nástrojů při různých úlohách. V rámci svojí práce jsem proto připravil novou studii srovnávající více než 60 nástrojů. Vybrané pokročilé nástroje pak byly detailněji srovnány z hlediska možností nasazení pro komplexní anotační úlohy a 3 z nejlepších byly srovnány v praktických experimentech.

Současná podpora pro zjednoznačňování entit a vytváření strukturovaných anotací komplexních vztahů v textu byla shledána nedostatečnou. Většina nástrojů pro zjednoznačňování nabízí pohled prezentující pouze typ entity a adresu stránky v referenčním zdroji, která ji popisuje. Při vytváření vztahů je pak většina práce ponechána na uživateli, pro kterého se často jedná o netriviální úlohu. S tím souvisí vysoká náročnost anotování pro uživatele i potenciální vysoký počet chyb při delším anotování. Ve svojí práci se proto zabývám návrhem a ověřením nových konceptů uživatelských rozhraní anotačních nástrojů umožňujících snížení náročnosti anotování pro uživatele, a tím i zrychlení procesu anotování a zvýšení kvality vytvářených anotací. Mezi nejdůležitější z nich patří sémantické filtrování a pokročilý způsob prezentace alternativních nabídek anotací.

Vzhledem k potřebě prakticky ověřit nové koncepty bylo nutné navrhnout a implementovat nový anotační nástroj. Nástrojů pro ověření nových konceptů pro sémantické anotování vzniklo v minulosti již několik. Jejich specializované zaměření však obvykle vedlo k tomu, že se vývoj soustředil jen na konkrétní vybrané aspekty uživatelských rozhraní a nástroje nebyly šířeji využitelné. Já jsem si naopak vytkl cíl vytvoření univerzálního modulárního nástroje se širokými možnostmi konfigurace, umožňujícího nejen ověření navržených konceptů, ale i praktické nasazení v reálném prostředí a poskytnutí vhodné platformy pro budoucí výzkum v oblasti anotování textu. Vytvořený nástroj umožňuje snadné doplnění a úpravu funkcionality pro zkoumání vlivu různých aspektů uživatelského rozhraní na proces anotování i jednoduché vestavění a ověření zcela nových konceptů. Pro prokázání praktické využitelnosti byl nasazen i v evropském projektu DECIPHER.

S výslednou verzí vytvořeného anotačního systému pak byly provedeny experimenty, které demonstrují, že uživatelské rozhraní anotačních nástrojů významně ovlivňuje proces anotování. Bylo při nich prokázáno, že nově navržená funkcionalita sémantického filtrování a prezentace alternativ anotací umožňuje zvýšení rychlosti anotování a kvality vytvářených anotací. Srovnáním s obecným nástrojem Protégé je rovněž ukázán přínos vytvořeného systému pro kolaborativní tvorbu ontologií.

Zbytek této práce je organizován následovně: Kapitola 2 stanovuje cíle mojí práce. Jak bylo zmíněno výše, hlavními oblastmi, na které se ve svojí práci zaměřuji, jsou vliv uživatelského rozhraní na rychlost anotování a kvalitu vytvářených anotací. Současně se zabývám i prací s ontologiemi a jejich vytvářením v rámci vedlejšího efektu kolaborativního anotování.

V kapitole 3 je následně uveden aktuální stav poznání v oblasti tématu práce od definice samotného pojmu anotace, přes využití anotování pro spolupráci a strukturování znalostí až po různé způsoby anotování a popis existujících řešení.

Následuje kapitola 4, která popisuje zvolené metody řešení pro dosažení uvedených cílů. Ty zahrnují srovnání existujících anotačních nástrojů, návrh a realizaci nového anotačního systému a vyhodnocení aspektů ovlivňujících práci uživatele při poloautomatickém anotování. Z daných aspektů pak vyplynuly návrhy na zrychlení procesu anotování a zvýšení kvality vytvářených anotací při současném kolaborativním vytváření ontologie.

Dosažené výsledky začínají kapitolou 5, která popisuje realizovaný anotační systém a jeho pokročilé funkce. V kapitole 6 jsou potom popsány provedené experimenty a jejich výsledky. Současně jsou zde uvedeny publikace, ve kterých byly tyto výsledky prezentovány. Na popis experimentů navazuje kapitola 7, která analyzuje jejich výsledky a vyvozuje z nich závěry. Jsou zde zhodnoceny získané informace a formulovány přínosy této práce. Opomenout nelze ani otevřené otázky, které bude třeba zodpovědět v rámci dalšího výzkumu.

Závěr se stručným shrnutím výsledků práce, vědeckého přínosu a plánu dalšího výzkumu je uveden v kapitole 8.

Kapitola 2

Cíle práce

Primárním cílem mé práce je ukázat, že propojení strukturování znalostí s tagováním, podpořeným pokročilými anotačními nástroji, zjednodušuje proces získávání a strukturování znalostí a na základě experimentů vyhodnotit a kvantifikovat přínosy tohoto postupu. Pro dosažení tohoto cíle se zabývám různými aspekty anotování se zaměřením na práci uživatelů. Tyto aspekty se vzájemně ovlivňují a jejich vhodná kombinace může zvýšit produktivitu uživatelů i kvalitu vytvářených anotací. Kvantifikace těchto zlepšení by měla patřit k hlavním přínosům mé práce.

2.1 Srovnání uživatelských rozhraní anotačních nástrojů

Existuje celá řada anotačních nástrojů, avšak jen malé množství studií se zabývá jejich srovnáním. Např. Diana Maynard [57] srovnává 6 nástrojů z hlediska různých aspektů, jakými jsou např. použitelnost (instalace, dokumentace, vzhled, ...), přístupnost (funkcionalita uživatelského rozhraní) a interoperabilita (platformy a formáty). Článek [56] pak uvádí srovnání z hlediska uživatele provádějícího manuální anotování, uživatele výsledných anotací, tvůrce korpusů a vývojáře systémů, do kterých je anotační funkcionalitu potřeba zabudovat. Další srovnání, jako např. [109], se pak zabývají převážně technickými aspekty a perspektivu uživatele opomíjejí. V průběhu svojí práce jsem neobjevil žádnou studii, která by se detailně zabývala srovnáním nástrojů z hlediska uživatele a vyhodnocovala dopad jednotlivých částí uživatelského rozhraní na proces anotování.

Prvním dílčím cílem mojí práce tedy bylo srovnat jednotlivé nástroje z uživatelského hlediska a vyhodnotit, jak jednotlivé části uživatelského rozhraní ovlivňují činnost uživatele. Následně bylo cílem kvantifikovat, nakolik uživatelské rozhraní ovlivňuje rychlost anotování a kvalitu vytvářených anotací.

2.2 Vytváření strukturovaných anotací

Většina existujících nástrojů nabízí možnost anotování textovou poznámkou, případně tagy. Strukturované anotace jsou spíše výjimkou a ve většině nástrojů nejsou dostupné, nebo je jejich tvorba značně omezená či komplikovaná. Dokumenty, které se v průběhu anotování mění, jsou pro velké množství systémů rovněž nevyřešený problém.

Některé současné systémy dále umožňují kolaborativní práci s anotacemi, ale souběžný přístup obvykle není dobře ošetřen. Anotace jsou aktualizovány pouze v okamžiku načtení či obnovení dokumentu a následně může dojít k situaci, kdy někdo edituje neaktuální verzi

anotace. To může v konečném důsledku vést i k fatální ztrátě dat. Většina systémů navíc pracuje pouze se statickými dokumenty či snímky dokumentů z určitého časového okamžiku. Pokud se dokument mění, může být problém s aktualizací anotované verze. Další komplikaci pak představuje uživatel, který si dokument pozmění lokálně u sebe a pak provádí synchronizaci s centrálně uloženou verzí, a to buď jednorázově s připojením k systému, nebo postupně vkládáním větších bloků textu přes schránku. V této situaci nelze využít malé granularity změn a aktualizace pozic anotací může být netriviální.

Dalším cílem mojí práce bylo navrhnout systém, který umožní využití strukturovaných anotací spolu s klasickými poznámkami a tagováním. Mělo být podpořeno vytváření jak jednoduchých anotací, na které jsou uživatelé zvyklí z existujících nástrojů, tak i strukturovaných anotací s atributy různých datových typů, vnořenými anotacemi a vazbami na jiné anotace. Pomocí strukturování pak mělo být možné vyjádřit složitější relace mezi anotovanými pojmy. Součástí práce má být i řešení pro spolupráci v reálném čase.

Systém by měl být dostatečně obecný, aby jej bylo možné využít k různým účelům (včetně těch, které jsou uvedeny v návrhu konceptuálního frameworku pro digitální anotační systémy na webu od Niranatlamphonga, Choochaiwattana a Springa [65]) a dosáhnout tak i rozšíření možností využití anotací. Při tom by mělo být ukázáno, že strukturované anotace umožňují lepší popis sémantiky a snazší strojové zpracování anotací (např. podpora automatického odvozování informací), přičemž pokud tyto anotace využijeme při kolaborativním anotování v reálném čase, usnadní se tím spolupráce při tvorbě dokumentů, ontologií a dalších činnostech.

2.3 Anotování kdekoliv a kdykoliv

Dostupné nástroje umožňují anotovat dokumenty v určitém prostředí, kterým je obvykle webový prohlížeč nebo editor dokumentů. S tím je spojeno i omezení na anotované dokumenty, a to na takové formáty, které se v daném prostředí využívají. V žádném ze zkoumaných nástrojů není možné dokument anotovat ve více formátech současně (např. tvůrce webové stránky ji anotuje v textovém editoru ve formátu prostého textu či v syntaxi wiki, zatímco návštěvník této stránky ji anotuje ve webovém prohlížeči ve formátu HTML). Rovněž není možné anotovat v různých prostředích současně (např. ve webovém prohlížeči a v desktopovém textovém editoru).

Cílem práce bylo také vytvoření řešení, které umožní anotovat text v různých prostředích a formátech současně a převádět anotace mezi jednotlivými formáty dokumentu. Uživatel by měl mít možnost anotovat text v textovém editoru na svém pracovním počítači a pokračovat v jednoduchém editoru s webovým rozhraním na chytrém mobilním telefonu v průběhu konference. Anotovat tedy má být možné kdekoliv (v libovolném prostředí) a kdykoliv (kdykoliv má uživatel přístup k síti, sám nebo současně se spolupracovníky). Dosud nebyl k dispozici žádný otevřený protokol pro přenos anotací, který by splňoval náročné požadavky na kolaborativní anotování v heterogenním prostředí. Součástí práce tak musí být i návrh nového univerzálního protokolu pro přenos anotací a nového formátu anotace.

2.4 Zvýšení kvality vytvářených anotací

Při návrhu řešení je třeba dbát na to, aby byl výsledný systém co nejvíce konfigurovatelný a umožnil nastavovat množství zobrazovaných informací a další aspekty uživatelského rozhraní. Srovnáním různých nastavení takového systému je potom možné nalézt nejvhodnější

konfiguraci, která umožňuje uživateli např. dosáhnout za stejnou jednotku času vyšší kvality anotací.

Dalším dílčím cílem práce tedy bylo dosáhnout výše uvedené modularity vytvořeného systému. Součástí této části práce bylo také s využitím systému experimentálně určit, jaký vliv na proces anotování má množství informací, které jsou uživateli zobrazovány. Na základě výsledků je potom možné optimalizovat množství zobrazovaných informací tak, aby se zvýšila rychlost anotování a kvalita vytvářených anotací.

V neposlední řadě bylo cílem kvantifikovat přínosy pokročilých funkcí využitelných při poloautomatickém anotování – sémantického filtrování na základě šablon pro jednotlivé sémantické typy a pokročilé prezentace alternativních nabídek anotací. Praktickými experimenty by mělo být ukázáno, že tyto funkce zvyšují rychlost anotování a kvalitu vytvářených strukturovaných anotací.

2.5 Formalizace znalostí na základě anotování textu

Vytváření ontologií a dalších prostředků formalizace znalostí představuje velmi náročný proces [39]. Specializované nástroje pro tvorbu ontologií, jako např. WebProtege[1], často vyžadují netriviální znalost formálních aspektů reprezentace znalostí. Oboroví odborníci, pracující přímo s dokumenty v určité oblasti (např. pracovníci muzeí), však takové znalosti obvykle nemají. Proto je ontologie často vytvářena specialistou, který pomocí rozhovorů, různých dotazníků a poskytnutých materiálů získává informace od doménových odborníků. Přichází se tak o možnost vytváření ontologií přímo při práci s textem. Pevně dané ontologie v rychle se vyvíjejícím prostředí obvykle neobstojí. V reálném prostředí je třeba znalostní struktury rozšiřovat a aktualizovat. Pro správce ontologie však může být obtížné objevovat chybějící koncepty a jejich významy, pokud není tato činnost přímo propojena se zpracováním relevantních dokumentů. Bohužel jen málo existujících nástrojů umožňuje anotovat pomocí dříve vytvořených formalizovaných znalostních struktur a současně tyto struktury rozšiřovat.

Cílem práce je tedy umožnit jak kolaborativní vytváření ontologie při anotování, tak i snadné přidávání nových konceptů do existujících ontologií, při kterém bude mít jejich správce k dispozici řadu příkladů jejich využití, doložených v konkrétních textech. Následně je cílem ukázat, že kolaborativní rozšiřování ontologie s využitím reálných anotovaných dat usnadňuje proces tvorby a aktualizace formalizovaných znalostí.

[1] http://webprotege.stanford.edu

Kapitola 3

Stav poznání v oblasti tématu práce

V této kapitole se zabývám současným stavem poznání v oblasti anotování textu. Nejprve se zaměřuji na samotný pojem anotace a další důležité pojmy a koncepty a následně na vymezení zaměření práce a existující programová řešení z dané oblasti.

3.1 Definice pojmu anotace

Historicky je slovo anotace definováno jednak jako poznámka přidaná jako vysvětlení k nějaké literární práci a jednak jako akt samotného anotování. Více o historii slova anotace lze nalézt ve studii Maristelly Agosti a kol. [4].

Dále budu slovem anotace označovat doplňující informaci přidanou k textu bez omezení významu této informace či charakteru anotované části textu.

Anotace se nejčastěji vyskytují ve formě jednoduchých poznámek přidávaných do textu. Tyto byly připisovány a vkládány do papírových dokumentů už před stovkami let. V souvislosti s přechodem na digitální formu přišla i elektronická forma anotace, tzv. digitální anotace, což je dle definice objekt související s obsahem jednoho nebo více jiných objektů [14]. Pro elektronické dokumenty dnes existuje velké množství aplikací umožňujících vložení poznámky, která se zobrazí např. ve formě zvýraznění textu a bublinové nápovědy při najetí myší na daný text.

Anotace umožňují uživatelům přirozeně spojovat osobní obsah s informačními zdroji. Mají široké spektrum použití od obohacování informačních zdrojů o vysvětlující informace až po přenášení a sdílení myšlenek a znalostí o anotovaném subjektu [9]. Mohou tvořit obsahovou vrstvu věnovanou vyjasňování významu informačního zdroje na nižší vrstvě [5]. Nejedná se tedy pouze o komentáře. Anotování lze považovat i za hodnotnou autonomní intelektuální práci [4].

Pomocí anotací lze také také doplnit strukturované informace k nestrukturovanému textu. Díky tomu začaly být anotace využívány k mnoha dalším účelům. Mezi ty patří např. propojení dokumentů, doplnění sémantiky k vybrané části textu [6], [65] apod. Doplněná informace může být jak formálního, tak i neformálního charakteru [22]. Anotace lze potom rozdělit na [22]:

- textové – obsahují textový komentář,
- odkazové – odkazují na jiný dokument (informace je ve formě cíle odkazu),

- sémantické – přiřazuje elementy ze specifikovaného modelu s hodnotami z kontrolovaného slovníku.

V článku *On Information Organization in Annotation Systems* [22] je následně uvedeno i jemnější členění anotací dle různých kritérií, která často závisejí i na jejich konkrétním využití. Různé dimenze anotací jsou i náplní dalších prací (viz např. článek od Catherine C. Marshall [54]). Problematika anotací je rozsáhlá a pro popis všech jejich aspektů není v tomto textu prostor. Nyní tedy přes konkrétní příklad využití anotací přejdu k další důležité oblasti týkající se této práce.

Formální sémantické anotace se využívají v sémantických wiki, kde obsahují vztahy s formálními definicemi pojmů ve znalostních databázích (kontrolovaný slovník). Doplněním anotace k určitému termínu v textu tak můžeme tento termín jednoznačně formálně popsat.

Dalším důležitým pojmem je tzv. tag (značka či štítek), což je krátká, obvykle jednoslovná, textová anotace. Tagging (tagování či značkování) je tedy specifický druh anotování, který spočívá v doplňování krátkých anotací k textu a je považován za nejjednodušší způsob přidávání metadat k fragmentům textu. Více lze nalézt v článku *Conceptual Data Structures (CDS) – Towards an Ontology for Semi-Formal Articulation of Personal Knowledge* [99].

3.2 Kolaborativní strukturování znalostí

V této podkapitole jsou vysvětleny důležité pojmy, které se týkají kolaborativního strukturování znalostí s využitím anotací. Nejprve je popsáno kolaborativní tagování, jakožto nejjednodušší způsob využití anotací pro kategorizaci obsahu. Následně jsou popsána prostředí, ve kterých je tagování provozováno, a pojmy týkající se vznikajících znalostních struktur. Na to navazují různé způsoby vytváření anotací a jejich využitelnost ke tvorbě těchto struktur. Protože spolupráce v reálném čase, zejména paralelní anotování stejného dokumentu více uživateli, do celého procesu přináší určité problémy, je uveden i jejich popis a využívaná řešení.

Použití tagování pro spolupráci

Pokud je tagging provozován kolaborativně, jedná se o tzv. social tagging. Tagy přiřazují různí uživatelé podle svého osobního názoru. Tagy mohou označovat nejenom druh či obecné vlastnosti pojmů odkazovaných v textu, ale také osobní preference či názory tagujících uživatelů. Některé anotace potom spíše než informace o anotovaném obsahu nesou informace o uživateli, jeho zájmech a preferencích, osobních názorech a dojmech [80]. Tagy lze tedy chápat jako propojující elementy mezi zdroji a uživateli, resp. mezi zdroji a pojmy v mysli uživatele [93].

Uživatelé mohou označit stejnou věc vzájemně se vylučujícími termíny. Pokud je jeden termín označen více protichůdnými tagy, je třeba určit, který tag bude zachován. To může provádět specializovaný kurátor, přičemž mu může pomoci srovnání množství stejných tagů od více uživatelů.

Tagy se využívají pro kategorizaci obsahu, navigaci a vyhledávání. Jako jeden z prvních je začal využívat Flickr[1], ve kterém slouží ke kategorizaci fotografií. Dalším příkladem využití je Twitter[2], kde ve formě hashtags (klíčová slova vyznačená křížkem) slouží ke kategorizaci zpráv (Tweets). Tagy se využívají i v blozích a dalších systémech, kde je potřeba

[1] http://www.flickr.com/
[2] http://twitter.com/

kategorizovat obsah. Dle Zhou a kol. [112] jsou anotace jedním ze základních prvků sémantického webu.

Z množství tagů lze určit popularitu tagovaného pojmu. Ta je následně využitelná k různým účelům od nabízení při vytváření dalších tagů až po vyhledávání dokumentů, které se daným pojmem zabývají.

Vzhledem k tomu, že tag je krátká textová anotace, může být interpretován různými způsoby (slovo či fráze má více významů). Proto jsou obvykle sdíleny ve skupinách uživatelů se stejnými zájmy, přičemž každá skupina si vytváří určitý slovník tagů, jejichž významy jsou dány zvyklostmi dané skupiny.

Prostředí pro kolaborativní tagování

Jak známo, wiki je web, který umožňuje uživatelům přidávat a měnit obsah. Wiki stránky jsou vytvářeny kolektivně pomocí jednoduchého značkovacího jazyka [106].

Sémantická wiki je wiki rozšířená o technologie sémantického webu [98]. Kromě samotných wiki stránek tedy obsahuje i model znalostí, které jsou na nich popsány. Umožňuje tak formálně zachytit identitu dat a vztahy mezi nimi. Mezi nejznámější software pro sémantické wiki patří Semantic MediaWiki [86], což je rozšíření MediaWiki [58], které umožňuje ukládat strukturovaná data do wiki stránek (anotovat wiki stránky). Více lze nalézt v článku Sebastiana Schafferta [81], na Wikipedii [105] a v článku *Semantic Wikipedia* [98].

Pojem sémantická wiki má dvojí chápání. Lze jej chápat nejenom jako wiki rozšířenou o technologie sémantického webu, ale i jako software pro vytváření sémantických dat způsobem běžným pro wiki (angl. wiky way [26]) [83], [82].

Strukturování znalostí

Z termínů folks (lidé) a taxonomie (klasifikace) vznikl termín folksonomie [97]. Dle Zhou a kol. [112] je sociální anotace formou folksonomie, která označuje internetovou metodu pro kolaborativní generování nelimitovaných textových popisů, které kategorizují obsah webu.

Pokud lidé označí nějaký pojem tagem, lze tento pojem chápat jako entitu náležící do množiny entit označených daným tagem (obdoba klasifikace). Tagy lze potom chápat jako označení typů entit.

Prostředkem pro sdílení konceptualizace domény jsou ontologie. Jedná se o sémantické modely popisující tématickou část světa [84]. Ontologie definují vztahy mezi entitami v dané doméně.

Pokud pomocí tagů definujeme entity a pomocí ontologií vztahy mezi nimi, vytvoříme model domény. Pokud je některá entita označena novým (dosud neznámým) tagem a tento tag začnou lidé využívat, vytvoří novou množinu (typ) entit. Entity označené tímto tagem takto získají novou sémantiku, která je označována jako objevující se sémantika (emerging semantics [19]).

Způsoby vytváření anotací

Anotace lze vytvářet:

- manuálně,
- poloautomaticky,
- automaticky.

Manuální tvorba anotací je prováděna lidmi, přičemž jedním z přístupů k této činnosti je social tagging. Získané informace potom mohou být např. ve formě folksonomií či kolaborativně vytvořených ontologií a nově objevené sémantiky (emerging semantics).

S rostoucím množstvím různých tagů a komplexitou anotací však použitelnost manuální metody klesá. Jsou-li např. tagy jako datum, osoba, místo a aktivita, uživatelé intuitivně vyberou nejvhodnější tag. Máme-li tisíce tagů označujících konkrétní osoby, uživatel již bez doplňujících informací nemusí být schopen vybrat ten správný. Vyhledávání doplňujících informací pak může být zdlouhavé a je obtížné dosáhnout dostatečné motivace uživatelů, abychom získali dostatek metadat. Je-li navíc tato metadata potřeba doplnit i do samotné anotace, práce může být zdlouhavá a vyčerpávající a anotování většího objemu dat pak z ekonomického hlediska není reálné.

Při poloautomatickém vytváření systém uživateli nabízí anotace či jejich části a uživatel je pouze schvaluje a vylepšuje. Nabídky mohou být nejčastěji využívané tagy, ale také komplexnější anotace. K získání takových anotací existují různé systémy pro extrakci informací z nestrukturovaného textu. Výhodou poloautomatického anotování je velká úspora práce uživatele a udržení vyšší kvality vytvořených anotací.

Automatické vytváření anotací lze využít např. tam, kde je třeba automaticky doplňovat databázi znalostí vytvářenou lidmi. Anotace jsou potom vytvářeny stejně, jako u poloautomatického vytváření, ale nejsou ihned kontrolovány a potvrzovány lidmi. Výhodou je možnost rychlého zpracování velkých množství dat.

Pro jednoduché věty a názvy s jednoznačnými významy je dnes přesnost automatických nástrojů velmi dobrá, ale protože konstrukce přirozeného jazyka nabízejí velmi vysokou variabilitu vyjádření, trénovací data pro strojové učení nástrojů nemohou pokrýt všechny případy a u některých složitějších konstrukcí může být dosahovaná přesnost neuspokojivá. Např. v projektu DECIPHER bylo studováno vzájemné ovlivňování umělců a ukázalo se, že i přes veškerou snahu některé relace jako „vzdává hold“ stále nejsou dobře pokryty.

Struktura znalostí (šablony vyplňované automatickým nástrojem) navíc může být komplexní a současné nástroje pro zpracování přirozeného jazyka a strojové učení nemusí být schopné takovou úlohu zvládnout. V doméně kulturního dědictví mohou být příkladem znalostní schémata analyzující různé pohledy na dílo a postoje jeho zastánců a kritiků. Na toto téma může existovat velké množství literatury s různými názory a významy a pro automatický nástroj je pak velmi obtížné vytvořit obecný model, který informace uvedené v textu správně rozpozná.

Přestože jsou aplikovány různé přístupy pro zlepšení této situace s využitím velkých dat z webu [113], nejméně počáteční množina trénovacích případů pro strojové učení automatických nástrojů musí být vždy vytvořena manuálně. Manuální anotování je tedy využíváno především pro získání trénovacích dat pro zlepšení přesnosti a výkonnosti automatických nástrojů.

Vzhledem k tomu, že komplexní struktury jsou často složené z komponent, které mohou být rozpoznány automaticky, může být samo vytváření příkladů převedeno na poloautomatický proces. Nabízení anotací vede potom ke značnému zrychlení procesu přípravy dat.

Prostředí pro spolupráci v reálném čase

Protože se ve své práci budu zabývat i kolaborativním anotováním v reálném čase, které umožní rychlejší a efektivnější vytváření anotací, je třeba zde zmínit i specifika tohoto druhu spolupráce.

Spolupráce v reálném čase znamená, že spolupracující uživatelé pracují paralelně. Je třeba řešit souběžný přístup, při kterém jsou změny prováděné jedním uživatelem ihned promítnuty do stavu aplikace druhého uživatele, resp. všech ostatních spolupracujících uživatelů.

Pokud by se každá změna projevila až poté, co se rozšíří mezi všechny spolupracující uživatele, byla by aplikace uživatelsky nepřívětivá (např. zobrazení aktuálního stavu až po několika sekundách od vlastního provedení změny). Je tedy nutné zvolit určitý kompromis a některé operace provést u daného uživatele a poté zaslat informaci ostatním. V tomto případě musejí být ošetřeny konflikty, kdy dva uživatelé provádějí vzájemně se vylučující operace (např. jeden uživatel edituje text, zatímco druhý jej maže).

V prostředí webu přenos stavových informací v reálném čase vyžaduje využití technologií, které umožní asynchronní přenos informací mezi klientem a serverem. Pro asynchronní komunikaci je velmi populárním přístupem AJAX (Asynchronous JavaScript + XML, více lze nalézt v článku Jesse Jamese Garretta [31]), ten však umožňuje pouze asynchronní komunikaci ze strany klienta, nikoliv ze strany serveru. Server tedy vždy musí čekat na požadavek od klienta. Aby bylo možné ihned přenést aktuální informace ze serveru ke klientovi, je třeba využít Comet [76] – přístup podporující obousměrně asynchronní komunikaci webového prohlížeče se serverem.

Příklady systémů pro spolupráci v reálném čase jsou kancelářský balík Google Docs[3], Novell Vibe[4] a Microsoft Office Online[5].

3.3 Existující řešení pro anotování

Pro anotování v prostředí webu existuje celá řada nástrojů, přičemž první z nich vznikly před více než 15 lety (viz článek LaLiberte a Bravermana [50] nebo např. GrAnT [85]). Pro ukázku jsem zvolil několik nástrojů, u kterých jsou patrné obvyklé přístupy, možnosti anotování a sdílení anotací. Nástroje jsou často uzavřené a k anotacím nelze volně přistupovat a strojově je zpracovávat. Anotace nelze přenášet do jiných prostředí a dále je využívat. Kombinace obecného anotování textovou poznámkou a tagování, u kterého jsou přiřazovány krátké značky, rovněž není samozřejmostí, i když je podporována u celé řady nástrojů. Podpora pro strukturování anotací (možnost využití anotace v jiné anotaci) je u existujících nástrojů výjimkou, a pokud je dostupná, nástroje mají jiná výrazná omezení.

Existující nástroje lze rozdělit do tří kategorií [74]:

- serverové,
- proxy,
- rozšíření (na klientovi).

U serverových systémů je vše prováděno na serveru, ke kterému uživatel přistupuje přes webové uživatelské rozhraní. Není tedy třeba instalovat žádný anotační program, ale lze anotovat pouze dokumenty umístěné na specifickém serveru. Často se potom jedná spíše o vestavěnou anotační funkci nějakého systému, spíše než o samostatný anotační systém.

U proxy řešení je využit server, který načte obsah z jiného zdroje (obvykle jiného serveru) a ten následně umožní anotovat. Veškerá anotační funkcionalita je zde opět na serveru (není

[3] https://docs.google.com
[4] http://www.novell.cz/cs/novell/novell-vibe/
[5] https://www.office.com/

nutné nic instalovat). Nevýhodou však je, že server musí mít přístup k danému zdroji (klient přes něj např. musí zaslat přihlašovací údaje). Anotační funkcionalita je navíc zabudována přímo do načtené webové stránky, což může zavádět např. problémy při tisku dokumentu.

Pokud je využito rozšíření, uživatel si musí nainstalovat součást anotačního systému do nějakého svého programu (obvykle webového prohlížeče). Práce s anotacemi pak může probíhat odděleně od práce s dokumentem (anotační systém nemusí měnit obsah načítaného dokumentu) a lze anotovat cokoliv, k čemu má uživatel přístup. Nevýhodou je zde nutnost instalace.

Další dělení anotačních nástrojů je pak podle úlohy, kdy rozlišujeme 2 druhy vytvářených anotací:

- lingvistické,
- sémantické.

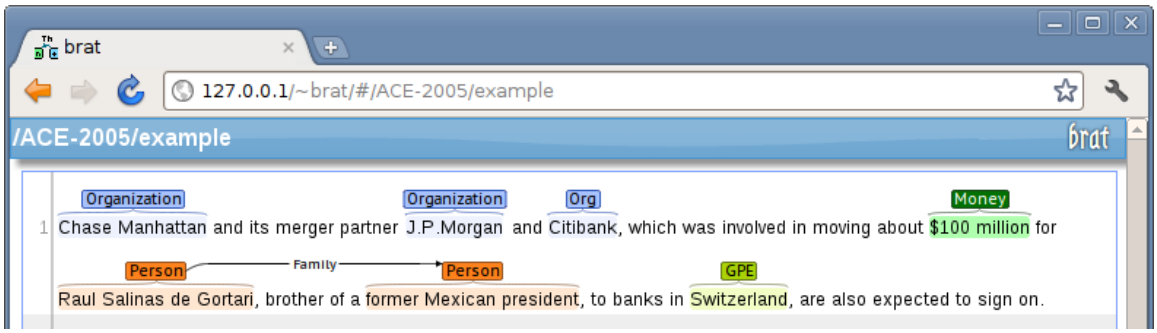
První přístup je reprezentován nástrojem BRAT [94] (viz níže), což je obecný lingvistický editor anotací. Například v případě anotování relací jsou zde zobrazeny pouze sémantické typy určitých slov či víceslovných výrazů, takže nelze jednoduše vyřešit víceznačnost a spojit jednotlivé entity s externími referenčními zdroji. Nástroj je vhodný pro úlohy, jakými jsou např. vyhledání koreferencí nebo extrakce vztahů mezi entitami s unikátními názvy.

Druhý přístup k poloautomatickému anotování reprezentují níže uvedené specializované sémantické editory anotací, zásuvné moduly pro existující anotační nástroje či rozšíření do webových prohlížečů [21], [35], [37], [38]. Tyto systémy jsou hůře využitelné pro úlohu obecné lingvistické anotace, ale excelují ve vytváření sémantických znalostních struktur (např. ve schématu RDF) a zjednoznačňování jmen. Nástroj, vytvořený v rámci této práce, rovněž patří do této kategorie.

Následují příklady existujících anotačních nástrojů. V závěru kapitoly se potom budu zabývat přenosem anotací mezi klientem a serverem.

BRAT

BRAT [94] je obecný lingvistický editor anotací, který již byl využit k přípravě celé řady datových sad. Klíčovými charakteristikami tohoto nástroje jsou plochá struktura základních anotací a jednoduchý způsob jejich prezentace (viz obrázek 3.1). Nástroj je vhodný pro vysoce specializované úlohy, jakými je např. vyhledání koreferencí nebo extrakce vztahů mezi biomedicinskými entitami s unikátními názvy. Odkazování do znalostní báze a hierarchické anotování komplexních vztahů, jakými jsou např. události, není podporováno.

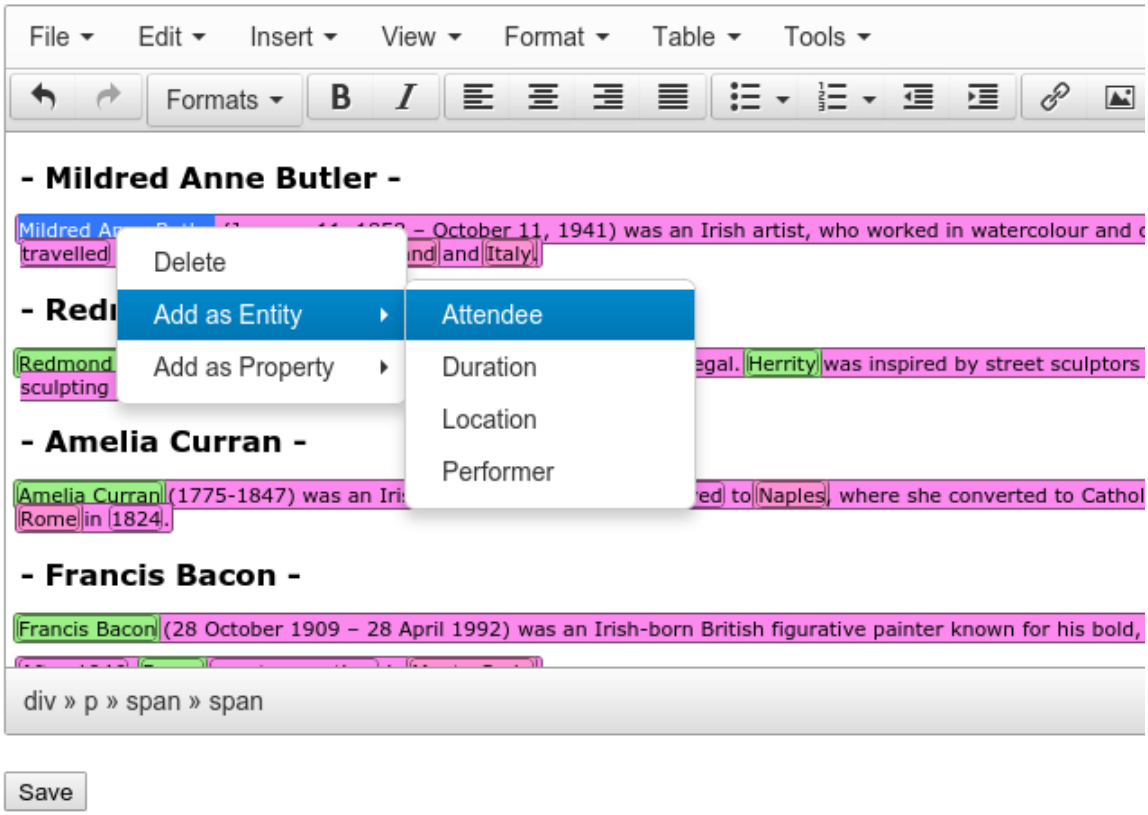


Obrázek 3.1: BRAT [15]

RDFaCE

RDFaCE [46], [45] je rozšíření WYSIWYG editoru TinyMCE [61] napsaného v jazyce JavaScript. Tento editor je využíván v řadě systémů pro správu obsahu (CMS) a anotační rozšíření tak lze využít v mnoha prostředích na webu. Anotace jsou ukládány přímo do dokumentu ve formátu RDFa (do atributů elementů HTML – více viz specifikace [1]). Vytvářejí se označením anotované části textu a vyplněním formuláře s atributy anotace. Zobrazení je pak v podobě okénka se stručnými informacemi, které se zobrazí při najetí myší, a pro detailní informace je třeba opět otevřít editační formulář.

Výhodou je jednoduchost řešení a snadné vestavění do libovolného systému využívajícího editor TinyMCE. Nevýhodami jsou nemožnost spolupráce v reálném čase, problémy s uchováním anotací při editaci textu a relativně pracné vyplňování formulářů při vytváření anotací. Rovněž je zde omezená možnost strukturování a nemožnost částečného překrytí anotací. Ukázka uživatelského rozhraní je na obrázku 3.2.



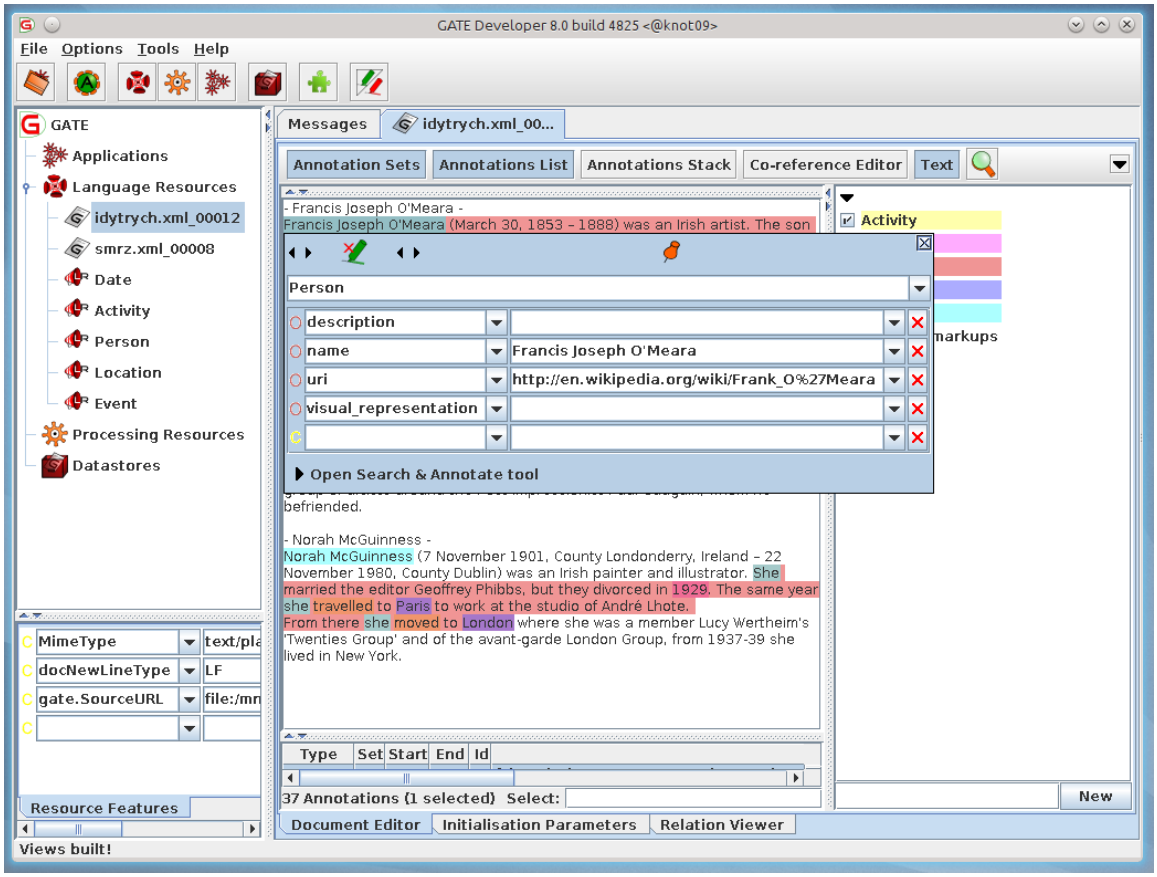
Obrázek 3.2: RDFaCE

GATE Developer

GATE Developer [32] je vývojářské prostředí, které poskytuje sadu nástrojů pro tvorbu programových komponent pro zpracování přirozeného jazyka. Jedním z integrovaných nástrojů je i vestavěný anotační nástroj. Anotace lze uložit do XML společně s anotovaným dokumentem, který je zde rozčleněn na identifikovatelné části (uzly), kterým se anotace přiřadí. Každá anotace může mít seznam atributů, které mají název a hodnotu. Typy anotací

i jejich atributy lze načíst ze souboru a libovolně přidávat v průběhu anotování. Zobrazení je ve formě tabulky anotací pod dokumentem a blikání fragmentu textu patřícího k vybrané anotaci. Anotace se edituje ve vyskakovacím okně (viz obrázek 3.3).

Hlavními výhodami jsou rychlost a jednoduchost anotování, možnost dynamického přidávání typů i atributů a možnost libovolného překrývání fragmentů. Hlavní nevýhodou je, že se jedná o rozsáhlou desktopovou aplikaci, kterou je před anotováním potřeba spustit, importovat do ní dokument ve formátu prostého textu a následně jednotlivé typy anotací. Pak teprve lze začít anotovat. Dalšími nevýhodami jsou nemožnost spolupráce v reálném čase a nemožnost tvorby odkazů mezi anotacemi.

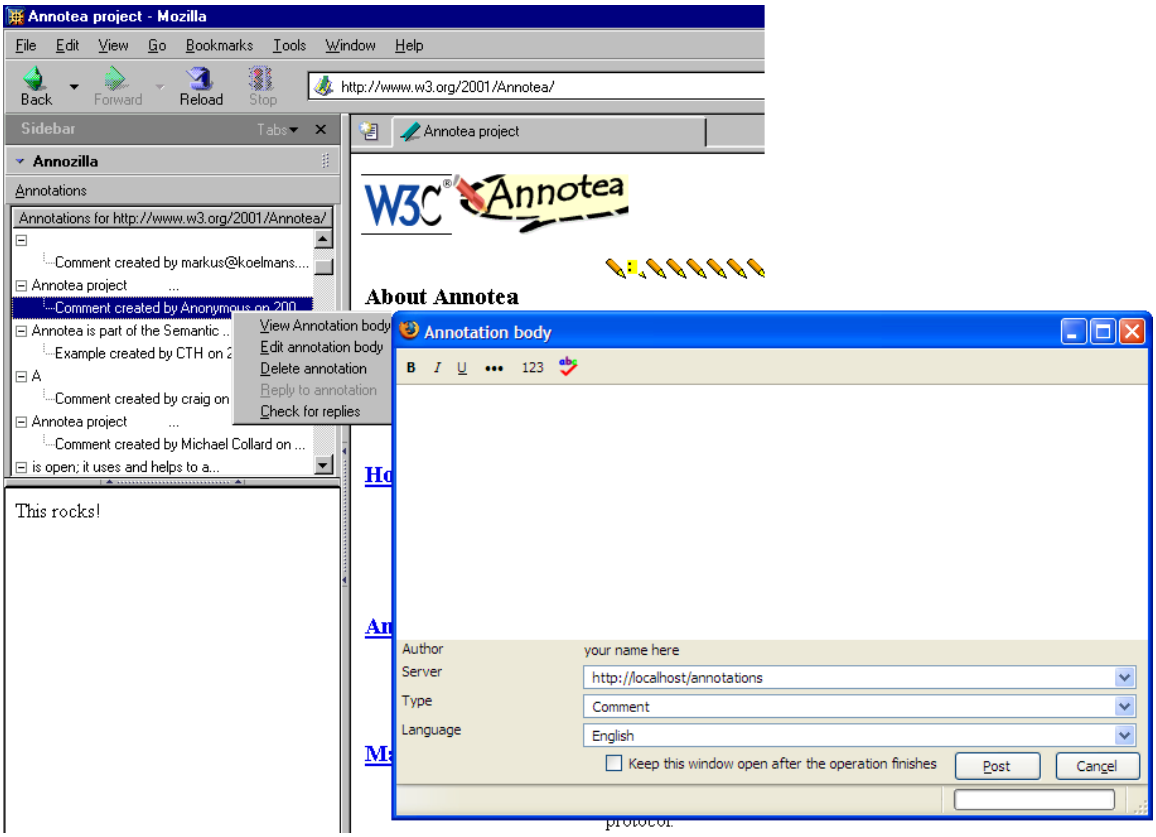


Obrázek 3.3: GATE Developer

Annozilla

Annozilla [62] je rozšíření webového prohlížeče Mozilla Firefox, které umožňuje anotovat obsah webových stránek. Anotace jsou ukládány na lokální či vzdálený server a pro identifikaci anotovaného místa v dokumentu je zde využit XPointer. Jedná se o jednoduché textové anotace, které mají definovaný typ a jazyk. Typ anotace je vybírán z několika předdefinovaných možností, mezi které patří např. komentář, vysvětlení, otázka, souhlas / nesouhlas apod. Přítomnost anotací v textu je zobrazována formou ikonky a samotné anotace jsou zobrazovány v postranním panelu. Pro přenos anotací na server se využívá protokol vyvinutý v rámci projektu Annotea (viz níže). Uživatelské rozhraní je na obrázku 3.4.

Výhodou je jednoduchost a uživatelská přívětivost řešení. Mezi nevýhody patří především nemožnost tvorby strukturovaných anotací, omezená množina typů, které nepřiřazují sémantiku anotovanému textu (nelze je využít jako tagy) a omezení na webové stránky (statický charakter dokumentu).



Obrázek 3.4: Annozilla [10]

Amaya

Editor webových stránek Amaya [72] od W3C má v sobě zabudovanou anotační funkcionalitu. Jedná se o součást projektu Annotea [44]. Anotace jsou externí (mimo dokumenty) a mohou být uloženy lokálně (v počítači uživatele) nebo na jednom či více anotačních serverech. Lokálně uložené anotace jsou privátní, vzdáleně uložené jsou veřejné.

Vzhledem k tomu, že editor je stále ve vývoji, je zde řada omezení a nevýhod. Filtr pro anotace určitého typu, od určitého uživatele nebo z určitého serveru je implementován v klientovi (načíst se musí vše). Vlastnosti anotací (včetně typů anotací) musejí být popsány v RDFS, což vylučuje dynamické strukturování anotací. Pracovat lze pouze s dobře strukturovanými dokumenty (XML či XHTML). Více informací lze nalézt v článcích Kahana, Koivunen a Swicka [44], [49].

Bundle Editor

Bundle Editor [111] nepatří mezi webové aplikace, ale jedná se o program pro demonstraci využití strukturovaných anotací při kooperativní tvorbě dokumentů. Umožňuje strukturo-

vání anotací jejich seskupováním do balíků. Anotace je možné filtrovat, řadit apod. Program je určen pro anotování textových dokumentů, do kterých jsou ukládány i anotace.

Program není veřejně dostupný a určený k širšímu využití. Využitá koncepce strukturování anotací se však ukázala jako efektivnější a uživatelsky přívětivější než využití prostých nestrukturovaných anotací [110].

ShiftSpace

ShiftSpace [88] je rozšíření do webového prohlížeče Mozilla Firefox, které umožňuje anotovat obsah webových stránek textovými poznámkami. Anotace jsou ukládány na server společně s kopiemi anotovaných stránek, přičemž kopie stránky může být modifikována. Modifikaci stránky lze chápat jako součást anotací na této stránce.

ShiftSpace jsem zde uvedl pro úplnost, protože byl dříve často uváděn jako jeden z nejznámějších nástrojů k anotování. Tento nástroj se však nejvíce vzdaluje od zaměření této práce, protože kromě anotování umožňuje i modifikace anotovaného textu. Modifikace textu, které jsou využity k doplnění informací v anotacích, potom narušují koncept anotování, kdy anotace není pouze přidaná informace, ale stává se obtížně odlišitelnou součástí originálního textu. Anotace potom nejsou využitelné mimo anotovanou kopii dokumentu, tedy ani v originálním textu.

Další editory

Existuje celá řada dalších řešení, která jsou určena převážně pro digitální knihovny (např. DiLAS [3]) nebo pro výzkumné účely. Příklady několika systémů lze nalézt v článcích Agosti, Ferra, da Rochy a kol. [3], [7], [74].

Za samostatnou zmínku stojí PREP Editor [63], který je jedním ze starších editorů s anotační funkcionalitou podporujících spolupráci v reálném čase. Je určen pro kolaborativní psaní dokumentů, z čehož vyplývá i omezení anotační funkcionality na potřeby specifické pro tuto činnost. Při spolupráci umožňuje manuální či automatické zasílání změn v dokumentu, přičemž u automatického je možné nastavení granularity výběrem ze tří možností: sloupec (column), kus textu (chunk) nebo stisk klávesy (keystroke). V jiných aspektech je však toto řešení z roku 1994 zastaralé.

Aktuální je nástroj Domeo[6] [21], který pracuje s novým formátem Open Annotation (viz níže). Má architekturu klient – server, kde klient je ve formě rozšíření do webového prohlížeče. Je excelentní v práci s obrázky (pravděpodobně nejlepší z dostupných nástrojů). Umožňuje vytváření jednoduchých anotací a odkazů na koncepty v ontologii. Přidávání atributů je však komplikované, neboť je za tímto účelem potřebné vytvářet zásuvné moduly. Anotace jsou navíc zobrazovány v bočním panelu a nikoliv přímo v textu, což snižuje přehlednost.

Různé systémy podporují různou inovativní funkcionalitu, přičemž některé funkce jsou velkým přínosem (lepší či rychlejší spolupráce, lépe strojově zpracovatelné anotace, uživatelská přívětivost apod.). Většina systémů je však vytvářena se zaměřením na řešení určitého problému, což vede nejenom k omezení využitelnosti takového systému, ale i k častému zanedbání některého z klíčových požadavků na anotační systém [67]. Jako konkrétní příklad takového systému lze uvést FAST (Flexible Annotation Service Tool) [8]. Jedná se o systém s univerzálním využitím nezávislý na platformě s mnoha výhodami [7]. Vzhledem k tomu,

[6] https://github.com/domeo

že jádro je zde zcela izolované od konkrétních systémů pro správu obsahu, aby bylo univerzálně využitelné, nemá žádné informace o konkrétním obsahu, ale pouze o anotacích. Není zde tedy žádná informace o obsahu anotované verze dokumentu, ale pouze odkaz na umístění v daném dokumentu. V případě potřeby je obsah dokumentu načten přes modul, který však potřebuje spolupráci se systémem pro správu obsahu, tedy s původním zdrojem informací. Pokud potom chceme anotovat webové stránky, musí mít anotační proxy přístup k danému webserveru, což je značné omezení.

Přenos anotací

Jak jsem uvedl výše, řada existujících nástrojů pro anotování je uzavřených a formáty a protokoly pro přenos anotací mezi klientem a serverem jsou nedostupné. Norma Web Annotation Data Model [79] (dříve Open Annotation Data Model [78]) od W3C pro ukládání anotací je aktuálně ve vývoji (těsně před dokončením této práce vyšla první verze, která by měla být stabilní) a využívá ji pouze malé množství projektů. Součástí této normy není žádný protokol pro přenos anotací. Dřívější projekt Annotea od stejné organizace [100]) se příliš neujal a i přes to, že model už je dostupný delší dobu, využívá se jen výjimečně (např. Annozilla). Mezi nevýhody modelu Annotea patří omezení na XML a XHTML dokumenty [44] a přenos anotací pomocí HTTP (komplikace pro přenos v reálném čase a server musí podporovat metody PUT, DELETE apod.).

Pro přenos anotací před určitou dobou firma IBM vyvinula API [30] a implementovala jej v projektu InsightLink. K implementaci však není dostupný dostatek informací a není zde popsán ani formát anotace (pouhé konstatování, že je ve formátu XML), ani protokol pro přenos. Pro využití je tedy nutné implementovat knihovnu, což v heterogenním distribuovaném systému zavádí nutnost implementace pro různé programovací jazyky a různá prostředí. Pro oblast obecného webu je tedy takové řešení nevhodné a je lepší využít jiný jednotný komunikační protokol.

Kapitola 4

Zvolené metody řešení

Pro dosažení výše uvedených cílů bylo třeba zvolit vhodné metody řešení. V této kapitole jsou tyto metody popsány. Nejprve je tedy uvedena zvolená metodika srovnání anotačních nástrojů se zaměřením na jednotlivá srovnávací kritéria. Protože pro dosažení většiny cílů práce bylo nutné navrhnout a realizovat nový anotační systém, následují nejdůležitější požadavky na tento systém a návrh jeho základní architektury. Zbytek kapitoly je pak zaměřen na zvolené metody pro zvýšení kvality vytvářených anotací a vylepšení procesu kolaborativní tvorby ontologií.

4.1 Metodika srovnávání anotačních nástrojů

Při srovnání anotačních nástrojů nejprve začnu srovnáním většího množství nástrojů z obecnějších hledisek, která jsem zvolil tak, aby bylo možné určit, do jaké míry je nástroj vhodný k úlohám popsaným v rámci této práce, tedy zejména k anotování složitějších struktur, jakými jsou např. události, a složených z entit s víceznačnými názvy. Současně je nutné určit, zda je s daným nástrojem možné provést potřebné experimenty. Zvolená hlediska jsou:

- typ aplikace (webová, desktopová, zásuvný modul, ...),
- granularita anotací (pouze celý dokument či jakýkoliv fragment textu),
- typ obsahu anotace (prostý text, formátovaný text, RDF, ...),
- tagy (volně zadávané, výběr ze seznamu či výběr z ontologie),
- atributy (jednoduché či strukturované),
- možnosti strukturování anotací,
- formát anotace,
- protokol pro přenos anotací,
- popis anotovaného fragmentu,
- dostupnost zdrojového kódu.

Prvním zvoleným hlediskem je typ aplikace. I když je zcela zásadní pro nasazení, pro účely srovnání zaměřeného na prvky uživatelského rozhraní je méně důležité. Dnešní webové aplikace totiž mohou mít stejné či velmi podobné rozhraní jako desktopové aplikace, což demonstruje např. aplikace ExTop[1].

Druhé kritérium zkoumá nejnižší jednotku textu, kterou je možné anotovat. Je-li podporovaná granularita příliš hrubá (lze anotovat pouze celý dokument), nástroj je pro sémantické anotování textu nevhodný. Takové nástroje budou ze srovnání vyřazeny. Pokud je z dokumentu před anotováním pořízen snímek a dokument je anotován ve formě obrázku, nástroj je rovněž nevhodný, protože vytvořené anotace nejsou použitelné pro strojové zpracování (označený text, a v horším případě i anotace, je třeba číst pomocí optického rozpoznávání znaků). Chceme-li anotovat jména osob apod., nejsou vhodné ani anotace zakotvené ke konkrétnímu bodu či oblasti na stránce.

Obsah anotace je chápán především jako textové pole pro poznámku či prostor pro jiný typ informace, který lze uložit přímo do anotace. Je zde především z historického hlediska, kdy poznámka je základním typem anotace a velké množství uživatelů očekává možnost jejího vložení jako minimální funkcionalitu nástroje. Toto kritérium jsem zvolil, protože zobrazení toho, co je uživateli notoricky známé, může mít vliv na jeho první dojem z daného nástroje.

Tagy poskytují možnost označení typu entity. Jak bylo uvedeno výše, je to jednoduchý nástroj pro klasifikaci, který je díky sociálním sítím velmi populární. Aby tagy bylo možné využít při strojovém zpracování sémantických anotací, je potřebné, aby měly jednoznačný význam a aby pro stejný koncept pokud možno nebyly využity dva různé tagy s duplicitním významem. Proto je vhodnější, když uživatel tag vybírá z databáze existujících konceptů dříve, než se rozhodne pro vytvoření nového. Je-li databáze rozsáhlá, může být tento výběr uživateli zjednodušen pomocí vhodného strukturování. Spíše než prostý lineární seznam některé nástroje prezentují výběr formou stromu.

Vzhledem k tomu, že se práce soustředí na vytváření strukturovaných anotací, je třeba vyhodnotit kritéria, která tento typ anotování umožňují. Dvěma nejčastějšími způsoby strukturování anotací jsou:

- přidávání atributů anotace,
- vytváření vazeb mezi anotacemi.

Atributy obvykle reprezentují vlastnosti anotované entity (např. jméno a datum narození osoby), ale mohou reprezentovat i vlastnosti samotné anotace (čas vytvoření, autora apod.). Mohou být jednoduchých datových typů (řetězec, číslo, datum, ...) nebo strukturovaných (odkaz na anotaci, vnořená anotace apod.). Méně vhodnou variantou je zadávání ve formě dvojic řetězců reprezentujících název a hodnotu, neboť číselné hodnoty a hodnoty dalších, často využívaných základních typů pak musí být při strojovém zpracování identifikovány a převedeny na odpovídající typ. Rovněž zde může být problém s víceznačností názvu atributu. Podstatné je i to, zda je seznam atributů fixní, nebo je lze přidávat. U některých nástrojů jsou totiž atributy definované v kódu nástroje a nemůže je změnit ani administrátor.

Dalším kritériem pro srovnání je využitý protokol pro přenos anotací mezi klientem a serverem a formát anotace. Jak bylo uvedeno výše, pro přenos anotací jsou často využívány uzavřené proprietární protokoly. Kvůli nedostupnosti univerzálního standardizovaného

[1] http://examples.sencha.com/extjs/6.0.2/examples/classic/desktop/index.html

protokolu řada implementací volí vlastní nezdokumentované řešení, kdy jsou zprávy zasílány pomocí HTTP (typicky se využívá AJAX). V případě využití samotného HTTP je pak komplikace s podporou metod PUT a DELETE, které jsou často z bezpečnostních důvodů zakázány, a absencí metod pro pokročilejší funkcionalitu.

Formát anotace pak může být omezujícím faktorem při strukturování anotací a je zcela zásadní pro interoperabilitu s jinými nástroji. S formátem anotace souvisí i popis anotovaného fragmentu. Je-li fragment popsán pouze pomocí offsetů, je po jakékoli změně textu obtížné rozpoznat, k čemu anotace patří, a provést aktualizaci. Při anotování webových stránek s dynamickým obsahem, jakým jsou např. reklamy, jsou pouhé offsety zcela nedostačující.

Dostupnost zdrojového kódu a obecně celé aplikace je pro naše účely důležitá. U licencovaných aplikací by bylo ekonomicky nevýhodné platit roční využívání pro relativně krátký srovnávací anotační experiment a současně by mohlo být komplikovanější vyhodnocení vytvořených anotací, neboť řada z těchto aplikací nemá možnost exportu ve využitelném formátu. Z nedostupnosti zdrojového kódu pak často plyne i uzavřenost formátu a protokolu, což bude patrné i z vytvořené srovnávací tabulky.

Jednotlivá políčka tabulky pak budou obarvena dle toho, zda je daný nástroj z tohoto hlediska vhodný, s určitými omezeními použitelný, či zcela nevhodný.

Výslednou tabulku se srovnáním nástrojů následně seřadím dle celkové vhodnosti jednotlivých nástrojů, přičemž bude brána v úvahu i důležitost jednotlivých srovnávacích kritérií pro následný postup. Několik vybraných nástrojů pak srovnám detailněji z hlediska pokročilé funkcionality (překrývání anotovaných fragmentů textu, typy atributů, vnořování anotací apod.). Dva nástroje, které ze srovnání vyjdou nejlépe, budou následně srovnány s nástrojem vytvořeným v rámci této práce při praktickém experimentu s vybranou skupinou uživatelů.

Při tomto srovnávacím anotačním experimentu se vyhodnotí vhodnost jednotlivých nástrojů na komplexnější anotační úlohu a to, které aspekty uživatelského rozhraní jednotlivých nástrojů nejvíce ovlivňují rychlost anotování a kvalitu výsledků. Podstatné bude, do jaké míry a jakým způsobem je uživatel ovlivněn (zda např. dochází k nezanedbatelnému nárůstu či poklesu jeho pozornosti apod.).

4.2 Anotační systém

Pro dosažení výše uvedených cílů bude třeba navrhnout nový anotační systém, se kterým bude možné provést experimenty. Návrh a implementace systému budou provedeny ve dvou iteracích. V první iteraci určím, které z prvků uživatelského rozhraní využívaných v existujících nástrojích by nový nástroj měl mít a navrhnu nové prvky, které dle mého zjištění současným nástrojům chybí.

Pro určení potřebné funkcionality, kterou bude nový nástroj vybaven, bude nutné stanovit nejen obecné požadavky, ale i konkrétní praktické případy použití daného nástroje. Nástroj proto bude následně vyvíjen v rámci evropského projektu Decipher.

Po skončení projektu Decipher a vyhodnocení informací od uživatelů bude návrh upraven a některé části implementace zcela přepracovány pro dosažení odpovídajících parametrů výsledného řešení.

Vývoj v oblasti anotací se dle Wenga a Gennariho [104] dělí na dvě hlavní kategorie. Jednou je vývoj datových schémat a druhou vývoj anotačních systémů. Vzhledem k výše uvedenému bude moje řešení v obou těchto oblastech, přičemž návrh formátu anotace spadá do první oblasti a návrh protokolu pro přenos anotací, klienta a serveru do druhé. Popsaný

protokol pro přenos anotací jsem proto navrhl tak, aby jej bylo možné využít i s jiným formátem anotace.

Nový anotační systém bude tvořen serverem a klienty do různých prostředí. Při návrhu tohoto systému zvážím nejenom požadavky dle cílů práce, ale i další aspekty uvedené v literatuře, jako např. základní operace a procedury, které by měly anotační systémy poskytovat dle [14], a požadavky na anotace a anotační systémy dle [67].
Anotace musí být [67]:

- zobrazeny přímo v dokumentu,
- vybaveny silnými vyjadřovacími schopnostmi,
- nezávislé na formátu,
- rozšiřitelné i sestavovatelné (nové typy musí navazovat na předchozí),
- distribuované a otevřené (kdokoliv může anotovat kterýkoliv dokument, k němuž má přístup),
- platformně nezávislé,
- robustní (určení pozice odolné vůči změnám v dokumentu).

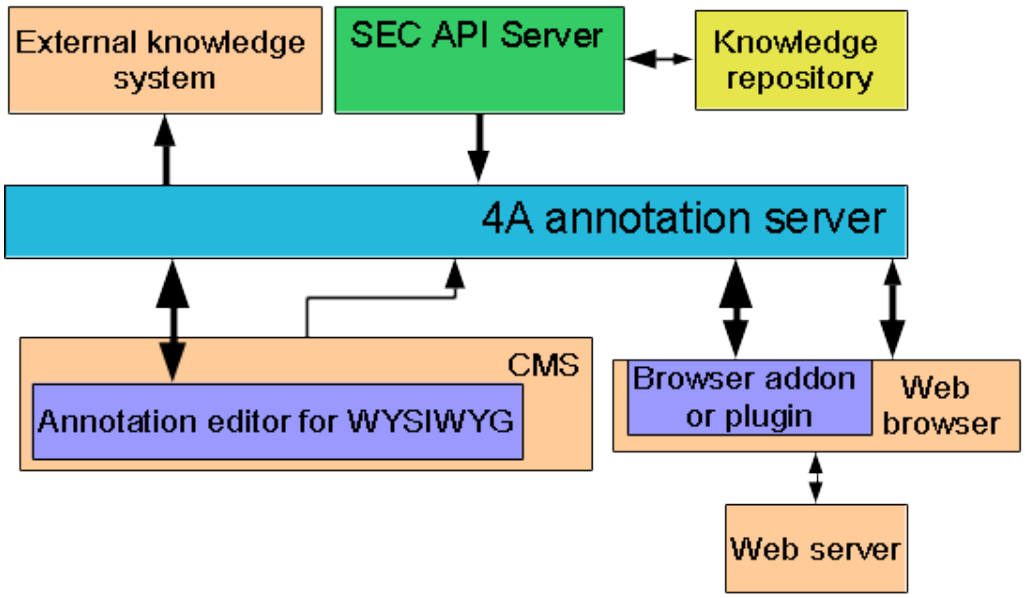
Pro návrh uživatelského rozhraní potom budou hodnotným zdrojem i požadavky na uživatelské rozhraní anotačního systému uvedené v článku [107]:

- pro vytvoření anotace by měl být potřeba minimální pohyb myší,
- anotovaný text by mělo být snadné odlišit od textu anotace,
- anotace by při čtení anotovaného textu měly být na 1. pohled viditelné,
- vztahy mezi anotovaným textem a anotacemi by měly být jasně viditelné,
- mělo by být možné snadno odlišit jednotlivé anotátory.

Takto docílím vytvoření systému, který překoná známá omezení a nevýhody existujících systémů.

4.2.1 Architektura

Architektura systému bude navržena s ohledem na interoperabilitu a nahraditelnost jeho jednotlivých součástí. Uživatelské rozhraní bude poskytnuto klienty, kteří budou komunikovat s anotačním serverem. Anotační server zajistí autentizaci uživatelů, kontrolu, ukládání a aktualizaci dat. Pro automatické vytváření nabídek anotací a přístup ke kontrolovanému slovníku (externí znalostní bázi) bude vytvořen server pro sémantické obohacení textu, na který bude anotační server zasílat požadavky. Anotační server poskytne rovněž možnost externí autentizace prováděné systémy pro správu obsahu (CMS) a rozhraní pro export dat do různých znalostních systémů. Zjednodušené schéma systému je na obrázku 4.1.



Obrázek 4.1: Základní architektura systému

4.2.2 Formát anotace

Protože většina existujících formátů anotací v době zahájení práce neposkytovala dostatečnou flexibilitu a nebylo do nich možné jednoduše uložit všechny potřebné informace pro strukturování anotací, nebylo při návrhu nového formátu vhodné vycházet z žádného z nich. Tento formát tedy musí být navržen od úplného základu a musí umožnit jak uložení jednoduchých anotací, které jsou uživatelé zvyklí vytvářet v jiných systémech, tak i ukládání komplexních strukturovaných anotací.

Vzhledem k požadavku na univerzálnost je nutné, aby bylo možné anotace využívat i jako tagy a ve vytvořeném systému tagovat. Navržený formát tedy musí umožnit i tuto alternativu.

Anotace budou strukturovány pomocí atributů (obdobně jako v návrhu Bremera a Gertze [16]). Oproti řešení v Bundle Editoru [111] však bude možné atributy využít nejenom k vytváření vazeb mezi anotacemi, ale také k uložení hodnot jednoduchých datových typů, což umožní vytvořit i vnitřní strukturu anotace.

U některých systémů je při přidání nového atributu anotace nutné upravit klienty (např. Annotea [49]). Této situaci je nutné se při návrhu vyhnout a umožnit jednoduché přidání libovolných atributů, což ovlivní nejenom návrh formátu anotace, ale následně i návrh klientů.

Vzhledem k tomu, že dle Cadize, Gupty a Grudina [20] je prokázáno, že největším problémem anotačního systému je osiřování anotací (situace, kdy po změně dokumentu již nelze najít původní umístění anotace), formát anotace musí být navržen tak, aby obsahoval dostatečně robustní popis pozice anotovaného fragmentu.

4.2.3 Protokol pro přenos anotací

Anotace je možné ukládat zcela nezávisle na anotovaných dokumentech, nebo společně s kopiemi dokumentů vytvořenými v době anotování. Pokud se při využití prvního přístupu dokument mění, anotace rychle ztrácejí platnost, protože po mnoha změnách již nemusí být možné určit původní pozici anotace. Při využití druhého přístupu může dojít k situaci, kdy se anotovaná kopie vyvíjí nezávisle na původním dokumentu. Tato situace je dle Phelpse a Wilenskyho [67] špatná, protože anotace by měla vždy anotovat dokument samotný, a ne něco jiného. Tento problém vyřeším tak, že společně s anotacemi bude uložena i kopie dokumentu, která bude průběžně aktualizována, včetně všech obsažených anotací. Při menších změnách je vyšší pravděpodobnost správné aktualizace anotace, a proto bude vhodné při editaci dokumentu ihned přenášet jeho změny. Dostupné protokoly pro přenos anotací však neumožňují přenos změn anotovaného textu a dalších potřebných informací, navíc nejsou vhodné pro navržený způsob anotování. Nově navržený protokol musí tyto nedostatky odstranit.

Navržený protokol musí umožnit nejenom přenos samotných anotací oběma směry, ale také přenos typů anotací. Obdobně jako popisují LaLiberte a Braverman [50] musí být možné jej bez další vrstvy přenášet v HTTP, což přináší určité problémy, protože protokol HTTP nebyl navržen pro obousměrnou asynchronní komunikaci.

Před zahájením anotování bude nutné provést synchronizaci dokumentu a v průběhu anotování přenášet změny v tomto dokumentu. Mezi přenášené informace pak bude patřit i nastavení klienta a serveru, což umožní rychlé dynamické rekonfigurace potřebné pro provádění experimentů s různými variantami nastavení.

Aby při větším množství uživatelů a vytvářených anotací zobrazení nebylo nepřehledné, uživatel může požadovat pouze anotace z určitého zdroje, kterým může být jiný uživatel, skupina uživatelů či jiný zdroj. Protokol tedy musí umožnit přihlašování a odhlašování od zdrojů anotací.

Anotace mohou být uživateli nabídnuty serverem, který je získá pomocí automatické extrakce informací z textu. Uživatel je potom může pouze schvalovat. Nabízené anotace musejí být přenášeny odděleně od uložených anotací, aby nedocházelo k jejich mísení.

Protokol musí umožnit i jednoduchou správu sezení a možnost budoucího rozšíření o správu přístupu k anotacím, která v první verzi protokolu není potřebná, protože se bude anotovat způsobem typickým pro Wiki (viz výše).

Uživatel často spolupracuje s určitou skupinou dalších uživatelů se stejným cílem. Tedy např. kurátor výstavy se svými asistenty, kteří mu vyhledávají potřebné informace. Odděleně pak může pracovat s jinou skupinou, jakou jsou například návštěvníci výstavy. Každá skupina může ke spolupráci potřebovat jiné typy anotací (asistent může doplnit anotace pro vyjádření souvislostí mezi fakty uvedenými v textu, zatímco návštěvník může vyjadřovat svoje názory na výstavu, doplňovat chybějící informace z jiných zdrojů apod.). Proto bude vhodné uživatele rozčlenit do skupin a typy anotací přiřadit konkrétním skupinám.

Při sémantickém anotování je potřeba, aby uživatelé označovali každou zmínku o stejné entitě stejným způsobem. Odpadá tak nutnost vytváření následného mapování různých označení se stejným významem. Protože však každý uživatel může intuitivně zvolit jiný způsob označení, je vhodné všem nabídnout společnou množinu předpřipravených entit, z níž mohou vybírat dříve, než se rozhodnou pro přidání vlastního popisu entity. Touto množinou je tzv. kontrolovaný slovník a manipulace s ním musí být v protokolu rovněž podpořena.

Protokol také musí být snadno rozšiřitelný o přenos dalších informací, protože je pravděpodobné, že s rostoucím množstvím různých využití systému budou přibývat i specifické požadavky na další funkcionalitu.

4.2.4 Server pro práci s anotacemi

Nejprve navrhnu jednoduchý modulární server pro práci s anotacemi, který bude poskytovat pouze minimální nutnou funkcionalitu. Tato funkcionalita zahrnuje:

- ukládání a aktualizace (synchronizace a zaznamenávání změn) kopií dokumentů,
- základní práci s anotacemi (ukládání, aktualizace a mazání),
- správu typů anotací,
- správu uživatelských účtů s webovým rozhraním,
- správu uživatelských nastavení,
- základní komunikaci s klientem.

Základní práce s anotacemi zahrnuje základní kontrolu syntaxe anotací, distribuci změn mezi klienty přihlášenými k odběru apod. Nejsou zde obsaženy žádné pokročilé kontroly správnosti, manipulace s nabídkami anotací, ani jiné pokročilé funkce.

Pokročilé funkce serveru pak budou navrženy a implementovány v rámci zásuvných modulů, aby mohly být jednoduše vylepšeny a nahrazeny.

Aby do budoucna bylo možné změnit protokol, kterým bude server komunikovat s klienty, a formát anotace, bude navržena vhodná vnitřní reprezentace anotace a požadavků klientů. Následně bude vytvořena abstraktní vrstva, která oddělí vstupní modul serveru od jeho modulárního jádra.

4.2.5 Klienti

Klienti budou editory anotací, které poskytnou (grafické) uživatelské rozhraní. U klienta navrhnu především koncept, základní podobu uživatelského rozhraní a základní a volitelnou funkcionalitu, protože do budoucna předpokládám implementace od různých autorů, kteří vytvoří doplňky do celé řady existujících aplikací (svých či od třetích stran). Jedním z prvních klientů bude doplněk do textového editoru TinyMCE [61], který bude vyvíjen v rámci diplomových prací na FIT VUT v Brně.

Klienty lze dle aplikace rozdělit na 3 druhy:

- doplňky do textových editorů,
- doplňky do prohlížečů dokumentů,
- další klienti.

Doplněk do textového editoru je specifický tím, že pracuje s měnícím se textem. Uživatel může kdykoliv zasáhnout do kterékoliv části textu a jednotlivé fragmenty přidávat, upravovat či mazat. Při práci s anotacemi je potom třeba na tyto akce odpovídajícím způsobem reagovat. Pokud je anotovaný fragment modifikován, je třeba rozhodnout, zda bude anotace upravena nebo vymazána či jinak zneplatněna. Zachování původní anotace není

možné, protože anotovaný text je její součástí. Vždy je tedy třeba upravit alespoň anotovaný fragment.

Textový editor může pracovat jak se strukturovaným, tak i s nestrukturovaným textem. Každý strukturovaný text může být převeden do formátu XML a pozice fragmentu v textu vyjádřena pomocí XPath, offsetu a délky. Server bude jako výchozí formát využívat (X)HTML a každý klient, který pracuje s jiným strukturovaným formátem, tedy musí pro účely synchronizace dokument převést do tohoto formátu a následně přepočítávat pozice anotací. Strukturovaný text lze převést i na nestrukturovaný (linearizovat) a pozici fragmentu v textu přepočítat pouze na offset a délku. Při převodu je však důležité využít jednotný algoritmus, aby při zpětném převodu pozice nebyla určena chybně. Server tedy nejprve při synchronizaci porovná, zda se linearizace od klienta shoduje s jeho linearizací, a následně sám vypočítává pozice anotovaných fragmentů ve strukturovaném textu. Takto může pracovat s daty od různých doplňků nezávisle na tom, v jaké formě uživatel s textem pracuje. Stejný text je tedy možné anotovat v jednoduchém editoru i ve webové stránce.

Doplněk do prohlížeče dokumentů pracuje se statickým textem. Typickým příkladem je doplněk do webového prohlížeče, který umožňuje anotovat zobrazenou webovou stránku. U webové stránky je třeba pracovat s textem ve strukturované formě, protože kolem anotovaného textu mohou být ve stránce i měnící se části, které k anotovanému textu nepatří (např. reklamy, počítadla apod.). Pokud se jedná o prohlížeč jiných dokumentů (např. PDF), doplněk může dokument převést do (X)HTML a přepočítávat pozice anotací, nebo dokument linearizovat a pracovat s ním v této podobě.

Mohou existovat i další varianty klientů do různých aplikací. Vždy je však třeba, aby byl anotovaný dokument jednoznačně identifikovatelný a bylo možné jej převést do XML či prostého textu a nalézt pozici anotovaného fragmentu.

Pokud se bude pracovat s jiným strukturovaným formátem než (X)HTML, bude na serveru uložena pouze linearizovaná podoba dokumentu, nebo dokument převedený do formátu (X)HTML. Vzhledem k tomu, že vizualizace je prováděna s originálním dokumentem, na formátu uložení na serveru nezáleží. Je však důležité, aby byl převod formátů prováděn vždy stejným způsobem, na kterém se autoři doplňků pracujících s tímto formátem musejí shodnout. Pokud budou různé doplňky využívat různý převod formátů, nebude možné spolupracovat mezi různými doplňky (bude docházet k chybám synchronizace a zneplatňování anotací). Vzhledem k velkému množství různých formátů dokumentů a jejich vývoji není možné definovat převody pro všechny možné formáty, ani jeden obecný převod. Pokud se však bude pracovat s linearizovanou verzí dokumentu, která bude vždy obsahovat všechen anotovatelný text, problémy by měly být minimální.

Uživatelské rozhraní klientů

Uživatelské rozhraní je velice důležité, protože tvorba digitální anotace je pracnější než u papírové [53] a čím je složitější, tím méně uživatelé anotují. Pokud jsou navíc anotace využity ke spolupráci, uživatelské rozhraní ovlivňuje nejenom počet, ale i typy problémů, které spolupracovníci pomocí anotací řeší [107]. Pokud tedy chceme dosáhnout toho, aby uživatelé vytvářeli velké množství anotací, které bude možné dále využívat, musí být anotování jednoduché. Současně je však nutné dbát na to, aby uživatel ve své práci nebyl příliš omezen. Zatímco v papírové podobě může uživatel vytvořit libovolnou anotaci, v elektronické formě je mu vždy vnucována určitá forma anotace a způsob anotování [53]. Strukturované anotace jsou jednou ze složitějších forem anotace, ale mají velké vyjadřovací schopnosti (minimální

omezení). Pokud bude jejich vytváření pomocí navrženého uživatelského rozhraní rychlé a jednoduché, můžeme z anotací získat velké množství hodnotných informací.

4.2.6 Nabízení anotací

Nabídky anotací budou generovány v serveru pro sémantické obohacení textu. Tento server bude bezstavový a poskytne jednotné rozhraní pro různé nástroje pro rozpoznáváni pojmenovaných entit. V anotačním serveru bude vytvořen zásuvný modul, který bude spolupracovat s tímto serverem a poskytovat filtraci nabídek a jejich zpřístupnění klientovi. Zajistí rovněž zpracování zpětné vazby od uživatele.

4.3 Zvýšení kvality vytvářených anotací

Pro zvýšení kvality vytvářených anotací je potřebné zjistit, jak uživatelé při anotování postupují, na co se zaměřují a co naopak snadno přehlédnou. Je třeba vyhodnotit typy a příčiny chyb, které dělají, a navrhnout řešení, které umožní tyto chyby minimalizovat.

Za tímto účelem bude provedeno několik sad experimentů s uživateli. Při prvním experimentu budou provedeny testy s menším množstvím uživatelů, přičemž budu sledovat a vyhodnocovat jejich postupy. Po dokončení každého úkolu identifikuji chyby a s daným uživatelem prodiskutuji příčiny chyb a opatření, kterými by těmto chybám bylo možné předejít.

Následně budou připraveny experimenty se zaměřením na množství a typy informací, které jsou uživateli zobrazeny, na způsoby jejich prezentace a na to, které informace zobrazit ihned a které až na vyžádání uživatelem.

Tyto experimenty budou prováděny s texty z domény historie a umění. Tato doména je vhodná z více důvodů. Prvním důvodem je, že část práce proběhne v rámci projektu Decipher, který se na tuto doménu zaměřuje. Druhým a rovněž velmi podstatným důvodem je, že jsou tyto texty srozumitelné pro široké spektrum uživatelů, protože historie (dějepis) se učí i na základních školách. Současně však běžní uživatelé často nemají takové hluboké znalosti, aby při zjednoznačňování měli z předloženého textu bez přidaných informací okamžitě jasno v tom, o jakou entitu se jedná, a úloha se redukovala na pouhé korektní vyhledávání v kontrolovaném slovníku. Třetím důvodem je pak snadná dostupnost textů s velkým množstvím víceznačných entit (názvy geografických umístění a jména osob).

Testovací data budou získána s využitím obdoby projektu Wikilinks[2], vytvořené ve Výzkumné skupině znalostních technologií na FIT VUT v Brně (Knowledge Technology Group, dále KnoT). V rámci tohoto projektu Ing. Otrusina a M. Mužila vytvořili skripty pro extrakci seznamu odkazů na Wikipedii z korpusu CommonCrawl[3] a pro následné nalezení odkazů se stejným textem vedoucím na různé stránky. Tento seznam odkazů bude bude využit jako základ. Položky seznamu budou filtrovány a s jejich využitím budou získány odpovídající texty. Následně bude nutné vyhodnotit správnost odkazů, pokrytí znalostní bází využitých nástrojů pro extrakci informací z textu a další informace, které budou nezbytné nejen pro přípravu, ale i pro následné vyhodnocení prováděných experimentů.

Při anotování vždy dochází k určitému kompromisu mezi rychlostí anotování a kvalitou vytvářených anotací. Je tedy třeba se zabývat i usnadněním práce pro zvýšení rychlosti pomocí poloautomatického anotování a sémantického filtrování.

[2] http://www.iesl.cs.umass.edu/data/wiki-links
[3] http://commoncrawl.org/

Nabízení anotací

Poloautomatické anotování je dnes běžné v celé řadě nástrojů. Lze identifikovat dva hlavní přístupy:

- předanotování,
- nabízení anotací.

Při předanotování je text anotován automatickým nástrojem a uživatel následně opraví chyby a doplní další anotace. Výhodou je jednoduchost tohoto řešení a fakt, že není potřebné explicitně potvrzovat každou anotaci. Tento způsob využívají nástroje jako RDFaCE či Gate Developer (viz výše). Nevýhodou je fakt, že hrozí vyšší riziko přehlédnutí chyby, protože anotaci, kterou uživatel zapomněl překontrolovat, nerozpoznáme od anotace, kterou zkontroloval či dokonce explicitně vytvořil.

Při nabízení anotací jsou anotace vygenerované automatickým nástrojem zobrazeny odlišným způsobem jako tzv. nabídky anotací. Ty pak uživatel musí explicitně potvrdit, jinak jsou zahozeny. Výhodou je menší pravděpodobnost chyby, nevýhodou je složitější řešení a fakt, že uživatel musí provést více explicitních akcí (je-li kvalita vygenerovaných anotací dostatečná, aby byl počet potvrzovaných anotací vyšší než 50 %, což dnes většinou můžeme předpokládat). Příkladem nástroje s touto funkcionalitou je Domeo[4].

Vzhledem k tomu, že úspěšnost automatických nástrojů stále není dostačující, předpokládám, že lepší způsob pro dosažení kvalitního výsledku bude nabízení anotací. Tuto hypotézu ověřím experimentem se dvěma nástroji, implementujícími opačné přístupy.

Sémantické filtrování

Při vytváření strukturovaných anotací sestavujeme složitější anotace propojováním anotací entit a dalších identifikovatelných prvků v textu (např. datum, interval, aktivita apod.). Anotace entit mohou být vytvořeny automaticky, poloautomaticky či manuálně. Při vytváření strukturované anotace pak uživatel vyplňuje jednotlivé atributy struktury. Např. u anotace události v projektu Decipher[5] musí vyplnit osobu (actor), aktivitu, místo, datum zahájení, datum ukončení a další atributy. Některé nástroje umožňují provázání anotací pomocí trojic (např. Pundit[6]) a některé pomocí atributů (např. RDFaCE). Volba anotace, která do vztahu vstupuje, je však vždy ponechána pouze na uživateli.

V rámci svojí práce navrhnu koncept sémantického filtrování, usnadňující uživateli volbu anotace, která je (v rámci atributu) ve vztahu s právě vytvářenou strukturovanou anotací. Tento koncept následně experimentálně vyhodnotím a určím, jaký má vliv na kvalitu vytvářených anotací a rychlost anotování.

4.4 Kolaborativní tvorba ontologií

Při kolaborativním vytváření ontologií bude důležité navrhnout vhodné režimy interakce, umožňující takovou spolupráci znalce domény s odborníkem na vytváření ontologií, aby bylo dosaženo optimálního výsledku s minimálním úsilím.

Tvorba ontologie je časově náročná a nelze očekávat, že odborník bude schopen v reálném čase spolupracovat se znalcem domény a přímo vytvářet ontologii a doplňovat všechna

[4] http://www.annotationframework.org/
[5] http://decipher-research.eu/
[6] http://thepund.it/

potřebná omezení, umožňující správné strojové odvozování informací. Je nepřípustné, aby znalec po vysvětlení konceptu strávil několik minut čekáním, až bude vše správně formalizováno. Systém tedy musí být navržen tak, aby při anotování textu znalcem vznikal dostatek metadat umožňujících doplnění ontologie. Spolu s ukotvením anotací v konkrétním textu by pak měl vzniknout dostatek podkladů, umožňujících porozumění konceptům a správné doplnění ontologie jejím tvůrcem.

Typický způsob tvorby ontologie je iterativní. V první fázi iterace tvůrce ontologie vytvoří základ, který dá k dispozici uživateli (znalci domény). Ten následně ve druhé fázi ontologii využívá (např. anotuje text) a identifikuje chybějící koncepty. Ve finální fázi tvůrce ontologie vyhodnotí dodané informace a doplní ontologii. Tradičně jsou informace od uživatele v nestrukturované formě (mohou být i na papíře) a často neposkytují dostatek informací k pochopení konceptu. Pro jejich porozumění a správnou formalizaci je tedy potřeba sezení, kde oba zúčastnění vše proberou, aby tvůrce ontologie s využitím získaných poznatků vytvořil další verzi, aniž by do ní zavedl chyby.

Ve své práci se zaměřím na všechny tři fáze tohoto procesu. V první fázi vyhodnotím, do jaké míry mohou anotace v textu pomoci s tvorbou (počáteční verze) ontologie. Ve druhé fázi pak budu zkoumat, zda navržené řešení urychluje tvorbu informací pro vytvoření další verze ontologie. V závěrečné fázi pak vyhodnotím, zda kolaborativní anotační nástroj umožní urychlení a zefektivnění interview tvůrce ontologie se znalcem domény.

Pro experimenty bude zvolena datová sada z aspektově orientované analýzy sentimentu. Hledání příslušných aspektů v textu a určení jejich významu je totiž jedním z ne zcela přímočarých úkolů a pro tvůrce ontologie může být velmi obtížné. Např. v oblasti obchodu se zbožím může být obtížnější určit, jaké aspekty zákazníky skutečně zajímají, zda je popis v textu ironický či nikoliv apod.

Kapitola 5

Realizovaný anotační systém

V první fázi práce byl navržen a implementován anotační systém, který byl nasazen v projektu Decipher. Po skončení projektu proběhla druhá iterace vývoje a s výsledným systémem byly prováděny další experimenty.

Anotační systém byl navržen a implementován s ohledem na výše uvedené požadavky. Pro implementaci anotačního serveru jsem zvolil jazyk Java, aby mohl být nasazen na různých platformách. Server pro sémantické obohacení textu byl vytvořen v jazycích Python a C, aby bylo možné snadno a efektivně obalit dostupný nástroj pro rozpoznáváni pojmenovaných entit vyvíjený ve skupině KnoT a aby bylo dosaženo minimálních paměťových nároků, které jsou při práci se znalostní bází kritické.

V rámci popisu anotačního systému zmiňuji i nedostatky, které byly odhaleny v rámci nasazení systému v projektu Decipher, a úpravy, které byly provedeny ve druhé iteraci vývoje.

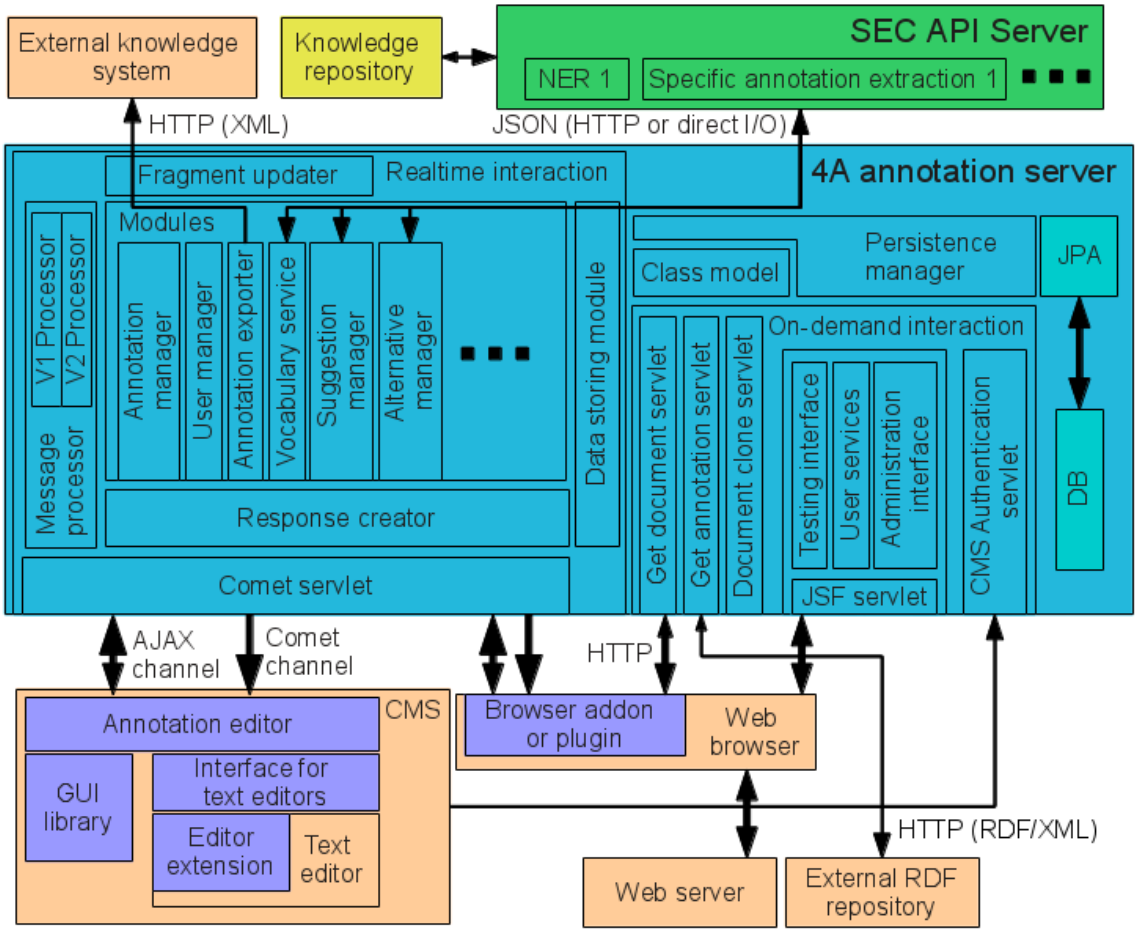
5.1 Architektura

Na obrázku 5.1 je detailní architektura celého systému. Pro komunikaci mezi anotačním serverem a klienty se využívá navržený protokol (viz níže), který je na nižší vrstvě přenášen protokolem HTTP dvěma kanály. Pro jeden kanál se využívá AJAX[1], pro druhý Comet[2]. Koncové body obou kanálů jsou stejné. Typ požadavku se rozliší pouze podle jeho obsahu a budoucí záměna HTTP za jiný protokol by tak nebyla komplikovaná.

Pro komunikaci mezi anotačním serverem a bezstavovým serverem pro sémantické obohacení (SEC – Semantic Enrichment Component) se využívá jednoduchý protokol založený na formátu JSON. Tento protokol byl vyvinut ve skupině KnoT pro vytvoření jednotného rozhraní k různým nástrojům pro rozpoznávání pojmenovaných entit a ke kontrolovanému slovníku. V rámci své práce jsem se na jeho vývoji aktivně podílel a navrhl do něj řadu vylepšení.

Anotační server umožňuje export anotací do externích znalostních systémů, a to jak v XML, tak i v RDF. Jedním z takových systémů byl StoryScope vyvíjený v rámci projektu Decipher (viz níže).

[1] Asynchronní komunikace z klienta na server [31]
[2] Asynchronní komunikace ze serveru na klienta [76]



Obrázek 5.1: Architektura anotačního systému

Systém je možné nasadit více způsoby. SEC může běžet jako samostatná služba a obsluhovat více anotačních serverů, nebo jej lze spustit přímo z anotačního serveru. Služby pro sémantické obohacení lze v anotačním serveru přepínat i za běhu.

5.2 Server pro práci s anotacemi

Vnitřní struktura anotačního serveru je také patrná z obrázku 5.1. Server lze rozdělit na část, která provádí zpracování v reálném čase (Realtime interaction) a na část, která pouze odpovídá na vnější požadavky (On-demand interaction). Dále je zde abstrakce nad standardním Java Persistence API pro přístup k databázi (Persistence manager) a datový model (Class model).

Požadavky od klientů jsou zpracovávány v reálném čase následujícím způsobem: Přijde-li na server požadavek od klienta, tento požadavek je nejprve zpracován ve vstupním modulu (Message processor), kde je převeden do vnitřní struktury pro reprezentaci požadavku. Vstupní modul umožňuje integraci komponent pro zpracování různých protokolů. Aktuálně je podporován protokol 4A verze 1 využitý v Decipher a protokol 4A verze 2, který bude popsán níže.

Následně je požadavek předán do modulární části serveru, kde jej komponent pro sestavení odpovědi (Response creator) postupně posílá jednotlivým modulům. Každý modul provede určité operace, jejichž výsledky uloží do struktury s požadavkem. Po volání posledního modulu tato struktura vstupuje do modulu pro ukládání dat (Data storing module), kde jsou veškerá data z požadavku uložena do databáze. Tím je zajištěno, že moduly mohou postupně provádět různé kontroly a o úspěšnosti či neúspěšnosti celého požadavku lze rozhodnout v kterémkoliv z nich. Není tedy problém doplnit správu oprávnění k dostupným zdrojům. Následně jsou znovu volány všechny moduly a mohou doplnit další data pro klienta, např. identifikátory přidělené databázovým serverem. Po vygenerování odpovědi na daný požadavek jsou všechny moduly volány znovu a mohou vytvořit data, která budou distribuována všem klientům, kteří mají synchronizovaný stejný dokument, pracují ve stejných skupinách uživatelů apod.

Moduly

Základním modulem serveru je `Annotation manager`, který zajišťuje správu dokumentů a anotací. Při synchronizaci či změně dokumentu vyhodnocuje změny a aktualizuje anotace. Zajišťuje zpřístupnění typů anotací i samotných anotací klientům a kontroluje nové či modifikované anotace.

Modul `User manager` zajišťuje funkcionalitu spojenou se členstvím uživatelů ve skupinách.

Modul `Annotation exporter` slouží pro export anotací do externích systémů v reálném čase. V konfiguraci modulu je zadána adresa systému, do kterého se data exportují. Nepodaří-li se vzdálený systém kontaktovat, jsou data ukládána do vyrovnávací paměti a odeslána při nejbližší příležitosti. Např. v projektu Decipher byl tento modul využíván pro vytváření stránek událostí v databázi StoryScope. Vždy, když uživatel anotoval událost, byla automaticky exportována a přiřazena k daným dokumentům. Uživatelé tak místo pracného vyplňování formulářů pouze vyhledávali události přímo v textu. Export je ve formátu XML, přičemž se do dané anotace rekurzivně doplní veškeré závislosti, což zjednodušuje zpracování dat na přijímající straně a urychluje interakci. Vyhodnocení v projektu Decipher ukázalo, že objem duplicitně zasílaných dat je vzhledem ke značné redukci počtu zpráv mezi systémy zanedbatelný.

Modul `Vocabulary service` zpřístupňuje kontrolovaný slovník klientům. V první verzi systému byly entity ze slovníku reprezentovány pouze URI a uživatelům se zobrazovaly s typy, názvy, případně obrázky entit. Při nasazení systému v projektu Decipher jsme následně narazili na problémy při aktualizaci slovníků, kdy byly anotace externí událostí znehodnoceny se zánikem dané URI. Proto byl protokol i tento modul vylepšen a společně s URI entity se do anotací nově ukládají i veškeré volitelné atributy entity. Po zániku URI pak anotace stále obsahuje kompletní data. Je-li slovník aktualizován, mohou být všechny entity, které jsou obsažené v nové verzi slovníku, aktualizovány.

Nabízení anotací zajišťuje modul `Suggestion manager`. Po synchronizaci dokumentu ihned zažádá o vygenerování nabídek anotací. Ty jsou následně na požádání zpřístupněny klientům. Dojde-li ke změně dokumentu, jsou nabídky v rámci možností aktualizovány pouze na základě vyhledávání fragmentů. Následně je zaslán nový požadavek na server pro sémantické obohacení a po obdržení výsledků se vyhodnotí rozdíly a aktualizuje databáze. Tímto postupem je zajištěno, že uživatel může plynule pokračovat v práci i přes to, že sémantické obohacení textu trvá i desítky sekund. Rozdíly se pak uživateli zobrazují postupně. `Suggestion manager` zpracovává i zpětnou vazbu od uživatele. Je-li nabídka anotace od-

mítnuta, míra důvěry v její správnost je automaticky snížena a současně již není nabízena v dané skupině uživatelů, neboť lze předpokládat, že v rámci činnosti této skupiny je daná nabídka nerelevantní. Je-li nabídka potvrzena, je míra důvěry v její správnost zvýšena, což usnadňuje rozhodování v jiných skupinách uživatelů, pro které však daný fragment textu může mít jiný význam. Každá anotace, kterou uživatel vytvoří, je rovněž transformována do nabídky pro jiné skupiny uživatelů. Potvrzení takové nabídky pak potvrzuje správnost dané anotace.

Modul `Alternative manager` zajišťuje správu alternativních nabídek anotací. Alternativní nabídky jsou ze serveru pro sémantické obohacení získávány separátně a jsou přiřazeny k příslušným nabídkám, které byly při sémantickém obohacení zvoleny jako nejpravděpodobnější. Je-li nabídka odmítnuta, nastavený počet alternativ je transformován na nabídky, které jsou prezentovány uživateli. O zobrazení dalších alternativ může uživatel zažádat i manuálně, nebo si může nastavit, aby mu byly zobrazovány ihned jak jsou k dispozici. Vyhodnocením dopadu různých nastavení na činnost uživatele se budu zabývat níže.

V části serveru, která odpovídá na další požadavky (On-demand interaction) se nachází především administrační rozhraní. Jedná se o webové uživatelské rozhraní umožňující import ontologie do grafu typů anotací, správu uživatelských účtů a nastavení různých komponent serveru.

`CMS Authentication servlet` umožňuje využít externí autentizaci uživatele prováděnou systémy pro správu obsahu, ve kterých je integrován klient. `Get document servlet` zpřístupňuje anotované kopie dokumentů a `Get annotation servlet` vytvořené anotace. Na rozdíl od exportu skrze `Annotation exporter` jsou zde anotace exportovány na vyžádání a nikoliv v okamžiku jejich vzniku.

Při běžné práci uživatelů často vzniká kopie dokumentu, která je následně spravována odděleně od originálu. Protože v tomto případě bude mít každá kopie dokumentu vlastní umístění a jejich obsahy se mohou vyvíjet zcela nezávisle, nelze spravovat anotace pro obě současně. Stávající systémy, které udržují anotace odděleně od dokumentu, tento problém obvykle nemají vyřešen a nově vytvořená kopie neobsahuje žádné anotace – ty je do ní nutné zkopírovat pomocí exportu a importu manuálně. Komponent `Document clone servlet` tento problém řeší a na požadavek naklonuje anotace z určitého dokumentu do jiného dokumentu. V projektu Decipher byl využíván v případě, kdy byl dokument z jiného muzea nalezen přes vyhledávání v centrálním úložišti a následně duplikován do lokálního StoryScope daného muzea, kde mohl být využit jako základ pro vyprávění o určitém předmětu kulturního dědictví (Narrative).

Poslední součástí serveru, kterou jsem dosud nezmínil, je `Fragment updater`. Jedná se o knihovnu pro aktualizaci pozic anotovaných fragmentů vyvinutou pod mým vedením v rámci výzkumu ve skupině KnoT. Každý fragment je nejprve vyhledáván na přesnou shodu a následně v obou směrech od původní pozice či od počátku dokumentu (pokud na původní pozici neexistuje žádný element). Po vyhledání jsou nalezené fragmenty porovnány a je zvolen ten, který se nejvíce shoduje s původním fragmentem. Překročí-li rozdíl mezi původním a nově nalezeným fragmentem nastavený práh, fragment je označen jako nenalezený a dojde k osiření anotace. Vyhodnocení je prováděno nejen na základě Levenshteinovy vzdálenosti, ale i na základě dalších kritérií, jakými jsou shoda prvního a posledního písmene, prvního písmene a posledního slova apod. Lze tak snadno detekovat změny jako zkrácení křestního jména osoby, vypuštění 2. jména a další případy, které se při testech v rámci projektu Decipher ukázaly jako problematické a nebyly pokryté dostupnými knihovnami. Knihovna je do značné míry konfigurovatelná a snadno rozšiřitelná. V rámci projektu byla vyhodnocena její úspěšnost z pohledu uživatelů a ti byli převážně spokojeni. Nejčastěji zmi-

ňovaným problémem výsledné verze je, že se anotace posouvá na předchozí či následující výskyt stejného jména osoby. Tento problém by do budoucna bylo možné částečně vyřešit kontrolou výskytu dalších anotací stejného typu na aktualizovaném fragmentu, ale zavedlo by to nutnost pracovat s databází anotací uvnitř knihovny, čímž by došlo ke znatelnému zpomalení procesu.

5.3 Klienti

Jak bylo uvedeno výše, klienty je výhodné realizovat jako doplňky do různých prostředí. Nejčastěji využívanými jsou:

- doplňky do webových prohlížečů,
- bookmarklety,
- doplňky do textových editorů.

Doplňky do webových prohlížečů

V rámci projektu Decipher byly vyvíjeny doplňky do prohlížečů Mozilla Firefox, Opera a Internet Explorer. Dokončen byl pouze doplněk pro Mozilla Firefox, protože u prohlížeče Opera v průběhu vývoje došlo ke změně jádra a v nové verzi byl již hotový kód zcela nepoužitelný. Obdobně skončil i vývoj doplňku pro Internet Explorer, kde s dalšími verzemi docházelo k většímu množství změn a už v polovině vývoje bylo věnováno větší úsilí ladění pro nové verze než doplňování chybějící funkcionality.

Ukázalo se, že uživatelé používají různé prohlížeče v různých verzích a doplněk pro Mozilla Firefox tak ve výsledném produktu nebylo možné nasadit. S rychlým vydáváním nových verzí prohlížeče navíc doplněk rychle zastarával a jeho údržba byla náročná. Z tohoto důvodu byl vývoj doplňku pro webový prohlížeč ve druhé verzi systému odložen.

V moderních prohlížečích dostupných v době dokončování této práce je patrná snaha o sjednocení a to jak v oblasti zobrazení webových stránek, tak i v oblasti rozšiřování prohlížečů doplňky a zásuvnými moduly. Aktuálně však stále není možné vytvořit doplněk do webového prohlížeče, který by bylo možné využít ve většině nejpoužívanějších prohlížečů[3] a současně nejpoužívanější prohlížeč stále nedosáhl 75 % využití[4]. Vývoj doplňku do prohlížeče tedy bude perspektivní do budoucnosti.

Bookmarklety

Bookmarklety jsou skripty vkládané do záložek ve webovém prohlížeči. Skript načte adresu aktuální stránky a ta je následně zobrazena ve stránce anotační aplikace. Stránka je tedy načtena externím serverem, je do ní injektován kód anotačního nástroje a následně je zobrazena uživateli. To však znamená, že anotační server musí mít přístup k anotované stránce, případně autentizační informace uživatele, což je bezpečnostní riziko. Nasazení v prostředí, kde nejsou všechny informace veřejné, je proto problematické. Vývojem Bookmarkletu jsem se tedy po analýze možností realizace dále nezabýval.

[3] http://www.w3schools.com/browsers/browsers_stats.asp
[4] http://gs.statcounter.com/#browser-ww-monthly-201412-201601

Doplňky do textových editorů

Doplňky do textových editorů neumožňují ihned anotovat jakoukoliv zobrazenou webovou stránku jako doplňky do webového prohlížeče. Ukázalo se však, že toto omezení pro účely projektu Decipher není podstatné, protože uživatelé vybrané materiály před anotováním ukládají do svojí kolekce dokumentů. Zajistí si tak jejich dostupnost i v případě, že web již nebude dostupný a případnou aktualizaci lze provést jak manuálně, tak i automaticky dle URL zdrojového dokumentu. Tento způsob využití lze aplikovat v mnohem širším měřítku než v rámci daného projektu a vzhledem k velkému množství stránek, které jsou každý den z webu odstraněny (dle statistik pouze na Wikipedii zanikne více než 1000 stránek denně[5]), je tento přístup výhodný. I z tohoto důvodu jej využívá celá řada systémů pro správu korpusů. Jako příklad lze uvést GATE Teamware[6] kde jsou všechny dokumenty nejprve uloženy v systému a následně anotovány.

S vývojem HTML5 přišla nová generace textových editorů, které splývají ze stránkou a v případě potřeby umožňují editovat její libovolnou část. Je tedy pravděpodobné, že tyto editory bude v budoucnu možné nasadit na libovolný obsah např. jednoduchým vložením několika řádků kódu triviálním doplňkem webového prohlížeče a jejich anotační doplněk tak bude mít téměř stejné výhody jako složitý doplněk do webového prohlížeče.

Velkou výhodou doplňků do textových editorů je možnost anotovat text už v okamžiku jeho vytváření. Autor textu tak může ihned jednoznačně identifikovat uvedené entity a vyznačit důležité vztahy mezi nimi.

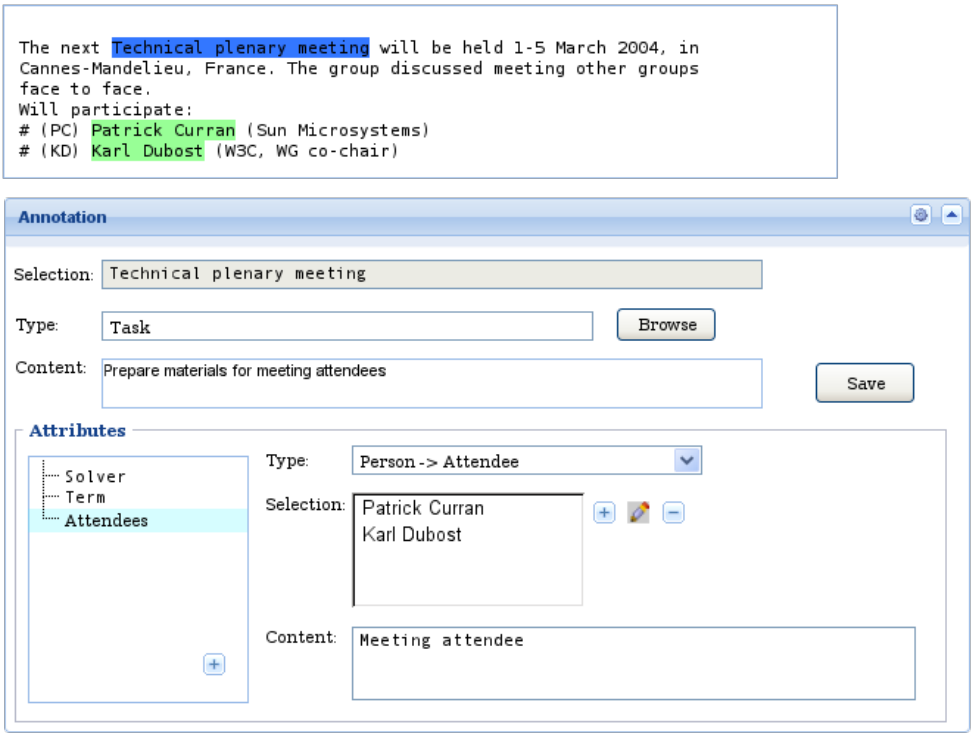
Z výše uvedených důvodů byl doplněk do textového editoru v projektu Decipher zvolen jako nejvhodnější řešení, jeho vývoj byl dokončen a byl integrován do výsledného produktu.

Aby využití doplňku nebylo omezené pouze na systémy pro správu obsahu využívající konkrétní editor, rozhodl jsem se pro řešení, kdy bude vytvořena vrstva abstrakce nad daným editorem a samotný klient anotačního systému bude implementován nad touto vrstvou. Vzhledem k tomu, že je od rozhraní textového editoru vyžadováno malé množství operací, bude snadné nástroj portovat na různé editory a zajistit tak jeho nejširší možné nasazení. Důsledkem tohoto rozhodnutí je potřeba oddělení uživatelského rozhraní anotačního nástroje od uživatelského rozhraní textového editoru, které se pro jednotlivé editory značně liší a bylo by obtížné generalizovat jeho modifikace.

Uživatelské rozhraní klienta

Jak bylo uvedeno výše, nejprve jsem provedl návrh uživatelského rozhraní. Ukázka mockupu z 1. verze návrhu klienta je na obrázku 5.2. Následně byly dle návrhu implementovány 2 verze klienta – editor anotací pro WYSIWYG editory v jazyce JavaScript a doplněk do webového prohlížeče Mozilla Firefox. Dále budu popisovat především 2. verzi klienta pro WYSIWYG editory do webových systémů pro správu obsahu. Oproti první verzi je zobrazení stručnější a horní část původního okna se zobrazuje při kliknutí na kořen stromu atributů (viz níže). Po prvotním testování se ukázalo, že je toto vhodný kompromis pro úsporu prostoru na obrazovce uživatele, neboť prvotní testy ukázaly, že okno implementované dle původního návrhu často překrylo příliš velkou část anotovaného textu a uživatel s ním musel často manipulovat.

[5] http://tools.wmflabs.org/wmcharts/wmchart0004.php
[6] https://gate.ac.uk/teamware/

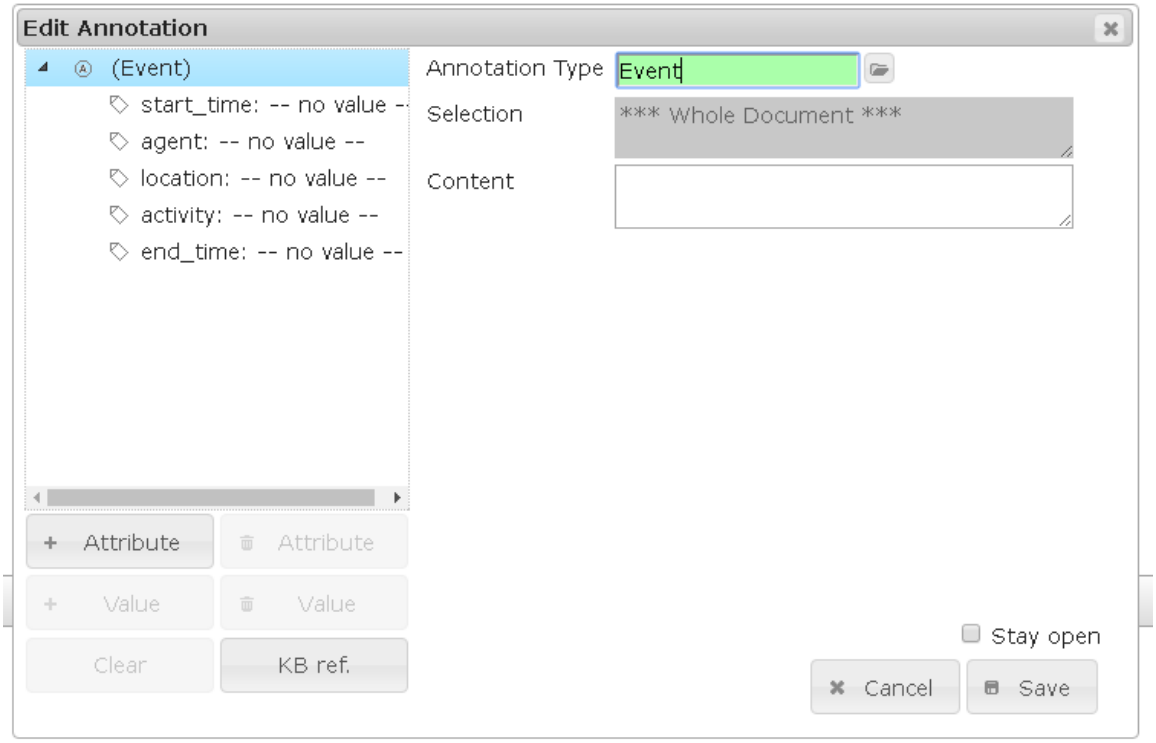


Obrázek 5.2: Návrh 1. verze klienta

Nejdůležitější částí editoru anotací je okno pro vytváření či editaci anotace, které je na obrázku 5.3. V levé části se nachází strom atributů. V kořeni tohoto stromu je samotná vytvářená anotace. Je-li zvolen kořen stromu, v pravé části okna jsou zobrazeny vlastnosti anotace jako celku. Nejprve je třeba zvolit typ anotace, k čemuž lze využít pole s automatickým doplňováním nebo dialog pro výběr ze stromu typů. I přes to, že typy anotací jsou organizovány do acyklického grafu, uživateli se vždy zobrazuje strom a některé typy jsou tak ve více větvích. Z hlediska uživatele je toto zobrazení přehlednější a k požadovanému typu může dojít více cestami. Např. umělecká díla mohou být členěna dle autorů a současně dle média (tužka, olejové barvy, pastely, ...). V poli s automatickým doplňováním jsou jednotlivé úrovně stromu odděleny šipkami. Uživatel také může přímo zadat název nového typu. Pracuje-li se s importovanou ontologií, její správce následně vidí tyto typy jako návrhy pro rozšíření ontologie o nové koncepty a současně se může přímo podívat na příklad jejich využití. Následně zajistí jejich správné začlenění do grafu kompletní ontologie.

Pokud tak neučinil již před otevřením anotačního okna, může uživatel po volbě typu vybrat část textu, která bude anotována. Fragmenty anotací lze libovolně vnořovat a překrývat, a to i částečně. Anotovanou část textu lze vybrat i v poslední fázi vytváření anotace, kdy je již jasné, co bude v anotaci zahrnuto (např. při anotaci události může být zahrnuta i předchozí či následující věta obsahující specifikaci data a času).

Textový obsah anotace (či textová poznámka) poskytuje prostor, na který jsou uživatelé zvyklí z jiných systémů. Pro sémantické anotace je tato poznámka nejméně hodnotná, protože složitost jejího strojového zpracování může být stejná jako složitost zpracování originálního anotovaného textu (je zde opět textový popis). Nicméně uživatelé v projektu Decipher se vyjádřili, že je vhodné mít tradiční prostor pro poznámku, kam mohou poznačit



Obrázek 5.3: Okno pro vytváření anotace

nějakou doplňující informaci pro člověka, který bude anotace využívat. Proto byl tento prostor zachován.

Pod stromem atributů se nacházejí tlačítka pro přidání a odstranění atributu či hodnoty. Každý atribut může mít více hodnot, přičemž pokud má jednu hodnotu, je zobrazena přímo u atributu. Pokud má více hodnot, je pod daným atributem další úroveň stromu, kde jsou jednotlivé hodnoty uvedeny.

Každý atribut může být jednoduchého či strukturovaného typu. Mezi jednoduché typy patří:

- textový řetězec,
- URI,
- URI obrázku,
- datum a čas,
- délka trvání
- celé číslo,
- desetinné číslo,
- pravdivostní hodnota,
- delší text,

- geografické umístění,
- entita z kontrolovaného slovníku,
- ...

Základní typy jako textový řetězec, číslo a datum poskytuje celá řada nástrojů. Uživatelé však často potřebují i další typy, jakými jsou např. geografické umístění. Pro kurátora výstavy může být důležité, v jaké vzdálenosti od jeho muzea se dílo nachází, aby určil, která díla zapůjčit. Zobrazení místa na mapě je pak velkou výhodou a pro vyhledání trasy k danému muzeu následně stačí asi 3 kliknutí myší. Obdobně může být důležitá i délka trvání (např. na jak dlouho lze dané dílo zapůjčit), pravdivostní hodnota (je dílo momentálně vystaveno) apod.

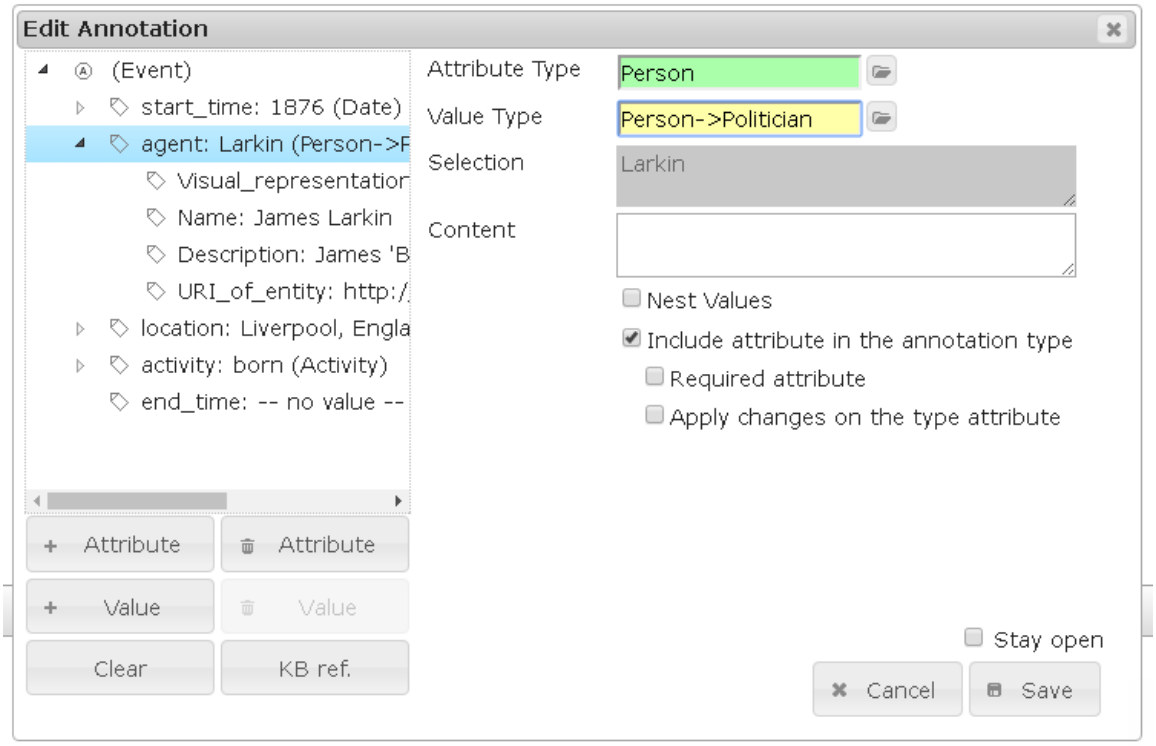
Sturkturovaný atribut může obsahovat odkaz na anotaci či vnořenou anotaci. Vnořená anotace je přímo závislá na nadřazené anotaci a pokud by nadřazená anotace zanikla, ztratila by význam. Vnořování lze využít pro víceúrovňové strukturování událostí apod.

Při importu ontologie do systému jsou jednotlivé třídy reprezentovány pomocí typů anotací a vztahy mezi nimi pomocí atributů. Individuálové jsou rovněž reprezentováni typy anotací, protože pro uživatele tato informace obvykle není podstatná a při explicitním uvedení by ho pouze zbytečně zatěžovala. Když totiž anotuje konkrétního člověka, uvažuje o tom, který je to člověk, a ne o tom, jestli bude anotovat obecně třídou reprezentující osoby nebo konkrétním individuálem. Uživatel typicky intuitivně zvolí individuála a nenalezne-li jej, zvolí třídu a individuála vytvoří instancí anotace. Toto je jedním z prvků odstínění uživatele od ontologie, které bude popsáno níže.

V dialogu pro přidání atributu uživatel zadá jeho název a zvolí typ. Následně se rozhodne, zda se atribut má stát součástí šablony daného typu anotace či nikoliv a zda má být vyžadován (např. lze nastavit, aby nebylo možné anotovat osobu, aniž bychom uvedli její jméno).

Byl-li strom typů importován z ontologie, mohly z ní být importovány i atributy, které nepatří k žádnému typu (v ontologii vztahy bez subjektu). Příklady takových atributů jsou „patří k“, „je součástí“ apod. – tedy typicky generické atributy, které lze obvykle aplikovat velmi obecně a nejsou proto definovány pro téměř každou třídu, ale globálně bez uvedení subjektu. Aby uživatel tyto atributy nevytvářel duplicitně s jiným názvem, může se při přidávání nového atributu nejprve podívat, zda právě přidávaný atribut nelze zvolit z těchto generických atributů. Tím může zvolit název a v některých případech současně i typ atributu, tedy typ entity na druhém konci vztahu.

Po přidání samotného atributu je v pravé části anotačního okna zobrazen formulář pro vyplnění jeho hodnoty. V tomto formuláři lze nejen dodatečně změnit typ atributu, ale také nastavit typ hodnoty, který může být nejen typem atributu, ale i jeho podtypem. U jednoduchých typů je vyplnění triviální. U strukturovaných se vlastně jedná o celou další anotaci a zobrazený formulář tedy odpovídá formuláři pro anotaci na vyšší úrovni v kořeni stromu atributů – viz obrázek 5.4. Uživatel se může rozhodnout, zda bude daná anotace odkazovaná nebo vnořená. Bude-li odkazovaná, jsou ve chvíli uložení vytvořený dvě anotace, z nichž ta na vyšší úrovni je závislá na té na nižší úrovni, ale ne naopak. Při editaci je pak potřeba je upravovat samostatně. Společná editace by sice byla možná, ale je nežádoucí, protože uživatel by si nemusel uvědomit, že nemodifikuje jednu anotaci ale dvě, a pokud by struktura anotací byla složitější, mohl by jako vedlejší efekt provést i neočekávané změny.



Obrázek 5.4: Vytváření anotace s odkazovanou anotací

Odkazovanou anotaci lze vytvořit stejně jako anotaci na vyšší úrovni (zvolit text, přidat atributy, ...), nebo ji vybrat z anotací v textu. Tento výběr bude detailněji popsán níže v rámci sémantického filtrování.

Protože ontologie typicky obsahuje třídy (člověk, místo, datum, ...) a konkrétní instance zde nejsou uvedeny, bylo by při anotování s využitím pouhé importované ontologie nutné manuálně vyplňovat velké množství informací. Z tohoto důvodu se využívá i kontrolovaný slovník. Vytvoří se anotace, která obsahuje odkaz na entitu v kontrolovaném slovníku a případné další atributy, které v tomto slovníku chybí. Uživatel tak např. u osoby nemusí vyplňovat jméno, příjmení, datum narození, hledat obrázek apod., ale tyto informace jsou automaticky doplněny po vyhledání dané osoby ve slovníku. Za tímto účelem je pod stromem atributů tlačítko „KB Ref.“, které nahradí základní atributy jako jméno, příjmení apod. jediným jednotným atributem (jednoduchého typu) s odkazem do kontrolovaného slovníku. Vyhledávací pole s automatickým doplňováním je předvyplněno anotovaným textem (často např. příjmení osoby) a uživatel jej případně upraví, aby bylo správně vyhledáno (např. doplní křestní jméno). Následně vybere z nabízených možností a může přidat případné další atributy, které ve slovníku chybí. Samozřejmostí je možnost omezit vyhledávání pouze na některé typy entit ve slovníku (v projektu Decipher bylo využíváno cca 10 typů dle aktivních slovníků).

Možnost přímého vyplnění atributů, kterou lze okamžitě přepnout na výběr z kontrolovaného slovníku, je v tomto nástroji unikátní. Jiné nástroje obvykle podporují pouze kompletně manuální vyplnění nebo výběr ze slovníku. Případně jsou dle slovníku předvyplněny hodnoty pro manuální vyplňování, což přináší možnost vytváření nekonzistence, která je obecně špatná. Pak je např. těžké určit, zda platí datum narození ve slovníku či v anotaci,

když uživatel tvrdí, že se jedná o entitu ze slovníku, ale upravil její údaje. V systému 4A je intuitivně veden k tomu, aby zvolil entitu ze slovníku a následně přidal atribut, kterým naznačí, že by datum ve slovníku mělo být aktualizováno. Případně lze pracovat ve speciálním režimu, který bude popsán níže, a přímo opravovat slovník.

Vizualizace anotací

Vytvořené anotace jsou vizualizovány v okénkách, která se zobrazují při najetí myší nad anotovaný fragment (viz obrázek 5.5).

Obrázek 5.5: Okénko anotace

U každé anotace je zobrazen anotovaný text, typ a další informace jako celé jméno či název, obrázek, popis, URI apod. Uživatel si může nastavit, které informace mu budou zobrazovány a které ne. Je-li v atributu vnořená či odkazovaná anotace, je zobrazena přímo v tomto atributu, aby bylo vše přehledně na jednom místě, ale je zde možné kliknout i na ikonku, která vnořenou či odkazovanou anotaci zobrazí přímo u jejího anotovaného fragmentu. Uživatel může jednotlivé úrovně sbalovat či rozbalovat a může si zvolit, do kolikáté úrovně budou anotace ve výchozím stavu rozbaleny.

Z okénka dané anotace lze přímo přejít do její editace nebo ji vymazat. Jedná-li se o nabídku anotace, lze ji potvrdit i bez editace nebo si k ní lze vyžádat alternativy, nejsou-li již zobrazeny.

Další funkce

Protože lze anotovat nejen fragment, ale i celý dokument, v klientovi je k dispozici i okno pro zobrazení anotací k celému dokumentu. Obdobně jsou řešeny i nabídky anotací k celému dokumentu, které jsou pro přehlednost uvedeny v samostatném okně.

Uživatel se může přihlašovat k odběru anotací, aby se mu zobrazovalo pouze to, co je pro něj podstatné. Anotace lze vybírat podle typu a zdroje (autora či skupiny uživatelů). V první verzi systému byly filtry zadávány přímo, ale při praktickém použití v projektu Decipher se toto ukázalo jako nepoužitelné. Uživatelé totiž mohou potřebovat pracovat ve více oknech současně a v každém mohou potřebovat jiné zobrazení. Odběry pak nelze nastavit centrálně pro uživatele, ale musí být nastaveny pro konkrétní instanci editoru anotací. Nastavení filtrů však může být relativně pracné a uživateli se nevyplatí jej opakovat pro každou instanci, kterou potřebuje mít odlišnou – v důsledku si raději nechá zobrazené vše v obou instancích, čímž mu však možnost nastavení odběrů nic nepřináší.

Ve druhé verzi systému proto byla zavedena další úroveň, kdy si uživatel nejprve vytvoří tzv. „odběry“ se seznamy filtrů a k těm se následně přihlašuje jedním kliknutím. Odběry lze také libovolně kombinovat, přičemž jsou pravidla seřazena dle specifičnosti a postupně aplikována. Uživatel si tedy může vytvořit jeden odběr, kterým zobrazí vše z domény umění, a následně druhý, kterým skryje všechny umělce, kteří nejsou malíři. Pro urychlení práce lze využít i sdílené odběry od ostatních uživatelů ve skupině.

Klient umožňuje i další nastavení jako omezení nabídek anotací na určitý typ anotace či oblast dokumentu, volbu skupin uživatelů, ve kterých se uživatel momentálně nachází, nastavení zvýraznění fragmentů (barva písma, pozadí, tučné písmo, . . .), přizpůsobení vzhledu uživatelského rozhraní apod.

5.4 Formát anotace

Při zahájení této práce v roce 2009 byly pro vytváření pokročilých anotací k dispozici pouze formáty Annotea [100] a Open Annotation [78], jehož vývoj byl právě zahájen (fáze 1 začala v květnu 2009[7]). Jiné formáty využité v různých nástrojích nebyly navrženy univerzálně a nebyly ani dostatečně zdokumentované. Přehled využívaných formátů lze získat z tabulky v příloze A.

Formát Annotea neposkytuje dostatečnou flexibilitu a pro složitější strukturované anotace tak není vhodný. V Open Annotation v roce 2010 stále chyběly některé klíčové části specifikace a dle dostupných dokumentů byl tento formát stále příliš nestabilní. Rozhodl jsem se proto, že navrhnu vlastní formát anotace, který bude později srovnán s právě vyvíjeným standardem Open Annotation.

4A

Při návrhu formátu anotace nazvaném 4A jsem se zaměřil na univerzálnost, interoperabilitu a efektivitu komunikace mezi klientem a serverem. Oproti jiným formátům je moje řešení obecnější. Využil jsem obdobný přístup jako Bremer a Gertz [16], kde jsou definovány jednoduché a komplexní atributy, ale navíc jsem přidal podporu rozšířených datových typů (viz níže). Atributy tak umožňují nejenom seskupovat anotace, ale mohou obsahovat i různé datové typy a součástí anotace tedy může být např. geografické umístění apod.

[7] http://www.openannotation.org/phaseI.html

Vztahy mezi anotacemi propojenými pomocí atributů jsou popsány názvem a typem atributu, čímž se moje řešení liší od přístupu Bremera a Gertze [16], kde je vztah popisován názvem a samostatnou odkazovanou anotací. Moje řešení je tak méně komplexní z hlediska zpracování a jednodušší pro uživatele. Vzhledem k tomu, že složitější vztah, pro jehož popsání nestačí název a typ odkazované anotace, lze popsat pomocí více atributů či úrovní vnoření anotací, nejsou omezeny vyjadřovací schopnosti.

Navržený formát anotace umožňuje různé způsoby anotování. Nejjednodušší je pouhé připisování jednoduchých poznámek k textu nebo značkování textu (tagging). Formát tedy může být využit k účelům, ke kterým se využívají současně dostupné formáty anotací. Navíc však poskytuje možnost strukturování anotací, a lze jej tedy využít i k mnoha dalším účelům, mezi které patří i výše zmíněné anotování událostí. Strukturování je umožněno pomocí atributů, kterými lze vyjádřit vazby na jiné anotace, nebo v nich mohou být jiné anotace přímo obsaženy.

Vzhledem k tomu, že v případě změny v dokumentu je třeba určit, které anotace jsou nadále platné a které ne, je třeba mít možnost porovnat kopii dokumentu z okamžiku vzniku či poslední úpravy anotace s aktuálním stavem. Proto je nutné před anotováním uložit kopii anotovaného dokumentu na server a při dalších přístupech k tomuto dokumentu vyhodnotit rozdíly aktuálního stavu oproti anotované kopii a případně upravit anotace. Pozice anotovaných fragmentů jsou potom určeny v lokální kopii (stav dokumentu v okamžiku anotace či aktualizace dat).

K určení pozice anotovaného fragmentu ve strukturovaném dokumentu existují 2 přístupy:

- linearizace,
- cesta (XPath), offset a délka.

Při linearizaci je dokument nejprve linearizován a potom je pozice anotovaného fragmentu vyjádřena offsetem od začátku dokumentu a délkou či offsetem konce. Je však nutné, aby linearizace byla prováděna vždy stejným způsobem (stejný způsob průchodu DOM[8] dokumentu). Pokud se potom změní některá z předcházejících částí v dokumentu, určení pozice je neplatné a je třeba jej upravit. Předchozí část však může být nejenom předcházející odstavec, ale také reklama v záhlaví webové stránky či jiný nesouvisející obsah. V takovém případě může být přepočítávání pozice potřebné při každém přístupu k anotovanému dokumentu, přičemž se musí vyhodnotit celý dokument a porovnat s anotovanou kopií na serveru.

Pokud je pozice určena cestou v DOM dokumentu, offsetem a délkou, není třeba vždy vyhodnocovat celý dokument. Pokud se např. změní reklama v záhlaví webové stránky, tato reklama je obvykle v uzlu DOM, který není v cestě k anotovanému fragmentu. Celá cesta tedy při různých drobných změnách zůstává stále platná. Protože však pomocí cesty určíme pouze uzel DOM dokumentu, je stále potřeba ještě upřesnění pozice fragmentu (např. věty, slova či znaku) pomocí offsetu a délky. Tyto se však při výše uvedené změně rovněž nezmění. Navíc je snazší nalézt pozici anotovaného fragmentu v případě, kdy je dokument vložen do jiného kontextu, např. jiné webové stránky (přechod na novou verzi stránek). Pokud se totiž změní prvních několik uzlů v cestě (při změně rozložení stránky), ale cesta v rámci dokumentu zůstane stejná a dostatečně dlouhá, aby bylo možné téměř jednoznačné určení, je možné anotovaný fragment nalézt. Pokud by byl fragment určen pouze offsetem a délkou, při změně obalujícího rozložení stránky už by bylo možné hledat pouze podle obsahu fragmentu,

[8] Document Object Model https://www.w3.org/DOM/Overview

což je v případě, kdy je anotováno jedno slovo, problém (nejednoznačnost, nemožnost určit, zda se nejedná o jiný dokument). Ukládání pozice v DOM dokumentu navíc podle Brushe, Bargerona, Gupty a Cadize [18] i podle Phelpse a Wilenskyho [68] patří mezi robustnější způsoby určení pozice anotovaného fragmentu textu.

Vzhledem k výše popsaným výhodám jsem zvolil určení pozice fragmentu pomocí cesty, offsetu a délky. Pokud se následně bude pracovat s nestrukturovaným (linearizovaným) dokumentem, cesta bude ukazovat na kořen dokumentu a pozice bude určena pouze offsetem a délkou.

Formát anotace je založen na RDF/XML [11], kde podmětem je vždy anotace. Formát XML je vhodný zejména pro možnost popsat libovolné struktury a pro efektivní zpracování ve webových prohlížečích, jak zmiňuje Sannomiya a kol. [80]. Oproti RDF jsem však zavedl určitá zjednodušení, především využití N-tic namísto pouhých trojic, čímž se zjednodušilo zpracování na serveru i na klientovi a zmenšil objem přenášených dat. Transformace do čistého RDF je následně triviální.

Součásti anotace jsou:

- typ anotace,
- datum a čas vytvoření,
- autor,
- URI anotovaného dokumentu (kopie na serveru),
- cesta k anotovanému fragmentu (XPath),
- offset anotovaného fragmentu,
- délka anotovaného fragmentu,
- textový obsah anotovaného fragmentu,
- textový obsah anotace,
- atributy.

Součástí návrhu je i reprezentace typů anotací, jsou hierarchické a jsou tedy uspořádané do stromové struktury. Pozdější vylepšení ve verzi protokolu 1.1 přidalo možnost vícenásobné dědičnosti. V poslední verzi se tedy jedná o obecný acyklický graf typů.

Typy anotací mohou být jak základní tradiční typy anotací (poznámka, popis, komentář, ...), tak i typy entit (člověk, věc, místo, ...) nebo dokonce konkrétní instance entit. Tím je poskytnuta dostatečná variabilita pro různé případy použití. Uživatelé mohou následně přidávat nové podtypy, čímž budou vytvářet komplexní graf typů. Tímto je umožněno odlišit nejenom různé typy anotací, ale také typy anotovaných fragmentů. Typy potom lze využívat jako tagy a uživatel tedy může anotacemi pouze tagovat. Každá skupina uživatelů sdílí svůj strom typů. Globálně tedy lze na typy pohlížet jako na strom, kde skupiny uživatelů vytvářejí jednotlivé větve. V tomto stromě jsou typy identifikovány anotačním serverem, skupinou uživatelů, cestou od kořene stromu typů a názvem. Z těchto vlastností je sestavena jednoznačná URI typu.

Pro prezentaci uživateli byl navržen tzv. „linearizovaný název“, což je cesta v hierarchii typů, kde jsou jednotlivé typy odděleny sekvencí znaků „ -> “ (např. „Person -> Artist -> Painter“). Skupina uživatelů v linearizovaném názvu není, neboť si ji uživatel zvolí v nastavení a nemá smysl, aby ji opakovaně zadával. Tato reprezentace je pro uživatele intuitivní a jednoduchá na použití, což bylo ověřeno v projektu Decipher.

Autor anotace je identifikován svým jednoznačným identifikátorem (URI), který má na serveru přiřazen. Zobrazované jméno (přezdívka) bude v atributu `name` a v dalších volitelných atributech mohou být jiné doplňující informace.

URI anotovaného dokumentu identifikuje anotovanou kopii daného dokumentu umístěnou na anotačním serveru. Před zahájením anotování nejprve dojde k synchronizaci, při které klient zašle kopii dokumentu společně s jeho URI na server. V procesu synchronizace server z URI odstraní číslo sezení a další nepotřebné informace a rozpozná, o který dokument se jedná. Následně klientovi zašle URI, kterým má být dokument identifikován v anotacích.

Cesta k anotovanému fragmentu jednoznačně určí oblast dokumentu (nadpis, odstavec, buňku tabulky apod.), ve které se nachází anotovaný fragment. Offset a délka určí přesnou pozici fragmentu. Textový obsah anotovaného fragmentu obsahuje samotný anotovaný text, který je třeba mít uchován pro případ, že se dokument změní. Pokud po změně fragment nebude v textu nalezen, bude označen jako neplatný, ale stále zůstane součástí anotace, aby informace o anotovaném textu zůstala kompletní. Anotace může náležet i k více fragmentům textu, i když v návrhu klienta z hlediska uživatele tato funkcionalita nebude obsažena, protože obvykle není potřebná a často ani žádoucí (je-li anotovaný text přerušen, je pravděpodobné, že by jednotlivé části měly být anotovány odděleně a následně propojeny na vyšší úrovni). Takové fragmenty tedy lze anotovat samostatnými anotacemi, které se následně vloží do atributů jiné anotace popisující daný celek či vztah mezi těmito částmi.

Vzhledem k tomu, že dle Marshalla a Brushe [55] jsou nejčastěji anotovány části vět a celé věty, obvykle nedojde k situaci, kdy by byl text vybrán přes více uzlů DOM dokumentu. Pokud však k této situaci dojde, budou součásti vybraného textu rozděleny do více fragmentů. Alternativou by bylo označení prvního a posledního uzlu, nebo označení počátečního uzlu a využití délky textu. Obě alternativní řešení se sice prakticky využívají, ale jsou méně robustní (odolná vůči změnám v dokumentu), a proto nejsou pro naše účely vhodná.

Kromě textového obsahu anotovaného fragmentu by bylo možné ukládat i kontext, který je tvořen částmi textu před a za anotovaným fragmentem. Význam kontextu při určování platnosti anotace je však malý, což bylo dokázáno v článku *Robust Annotation Positioning in Digital Documents* [18], a v případě potřeby je možné jej jednorázově vyhodnotit při aktualizaci anotované kopie dokumentu na serveru. Právě při této aktualizaci je totiž nutné vyhodnotit platnost anotací a ty případně upravit, přesunout na globální úroveň dokumentu či zneplatnit (vymazat). Pro úpravy, případně přesouvání na globální úroveň, se využívají tzv. algoritmy pro změny pozic anotací (repositioning algorithms či reattachment algorithms), jejichž využití je blíže popsáno v literatuře [9], [68] a [69]. Přehledná tabulka srovnání těchto algoritmů, přesněji jejich robustnosti vůči jednotlivým změnám, je uvedena v článku Phelpse a Wilenskyho [69]. Nicméně jak bylo uvedeno výše, v systému 4A byl implementován vlastní algoritmus, aby bylo umožněno variabilní nastavení a bylo jej možné optimalizovat pro využití v projektu Decipher, kde se objevily specifické potřeby nedostatečně pokryté standardními algoritmy, jako např. korektní aktualizace při připsání či zkrácení druhého jména osoby apod.

Přesunutí anotace na globální úroveň lze nazvat také osiření (orphaning), kdy již v dokumentu nelze nalézt anotovaný fragment, nebo se tento fragment příliš změnil a původní anotace pro nový obsah fragmentu neplatí. Podrobněji se touto problematikou zabývá Brush a kol. [18]. V případě, že anotace náleží k více fragmentům, může při osiření jednoho fragmentu zbytek anotace zůstat platný.

Textový obsah anotace obsahuje uživatelem zadanou textovou informaci. Význam může být blíže specifikován typem anotace, ale typicky se jedná o textovou poznámku, komentář, popis, doplňující informaci apod.

Atributy umožňují strukturování anotace. Každý atribut má název, typ a hodnotu. Typem může být některý ze základních datových typů z XSD (číslo, řetězec, datum apod., viz např. XML Schema Tutorial [101]), rozšiřujících datových typů (např. geografické umístění ve formátu W3C Basic Geo [17]), vnořená anotace či odkaz na anotaci. V případě vnořené či odkazované anotace je typ atributu přesněji určen typem anotace.

V řadě existujících formátů jsou součástí anotace i speciální atributy umožňující spolupracujícím uživatelům reagovat na dané anotace, přičemž vznikají vlákna či množiny anotací. Tento přístup využívají např. Agosti a Ferro [5]. V mnou navrženém formátu tyto speciální atributy nejsou potřebné, protože lze k těmto účelům pouze vyhradit jeden typ atributu, který navíc může mít podtypy dle charakteru reakce.

Ukázka strukturované anotace je společně se specifikací formátu umístěna v příloze C.

Open Annotation

Open Annotation [78] je nový formát strukturované anotace vyvíjený konsorciem W3C. První verze návrhu byla oficiálně vydána v březnu 2012 a nejnovější verze je z března 2013 (v době finálního dokončování této práce v roce 2016 byla nahrazena specifikací Web Annotation [79], která přináší podrobnější dokumentaci a drobná vylepšení). Formát je založen na RDF a jeho specifikace má několik částí:

- jádro,
- specifikátory a specifické zdroje,
- konstruktory pro vícenásobnost,
- publikování,
- rozšíření.

Jádro specifikace popisuje základní strukturu anotace. Ta má obvykle typ, cíl (co anotuje) a tělo (obsah anotace, tedy samotná přidaná informace). Cílů a těl může být i více a mohou být různých typů.

Cíle se popisují pomocí tzv. *specifikátorů*, které specifikují anotované zdroje (např. dokumenty, videa apod.) a vybírají jejich konkrétní části. Jsou-li např. anotována 2 po sobě následující slova v různých elementech dokumentu (např. jméno a příjmení, kde příjmení je zvýrazněno tučným písmem), vytvoří se cíl, jehož obsahem je konstruktor pro vícenásobný obsah. Ten obsahuje 2 položky, přičemž každá z nich obsahuje specifický zdroj (zdroj s upřesněním části). Specifický zdroj je pak popsán obecným zdrojem (popisuje typ, formát a URI dokumentu) a selektorem, který je dále specifikován selektorem fragmentu. Samotný fragment je pak popsán s využitím jazyka XPointer[9].

[9] Jazyk pro adresování částí dokumentů ve formátu XML [25]

Tělo anotace může být textové nebo strukturované, přičemž jeho typ může být ze slovníku DCMI[10] nebo libovolný obsah v RDF ve vestavěném textovém těle. Pro reprezentaci atributů strukturovaných anotací lze využít pouze RDF, kdy se do těla anotace vloží graf ve formátu Trix[11].

Popis publikování specifikuje správný postup serializace. Rozšíření pak přidávají další typy selektorů, trojice pro sémantické tagy apod.

Srovnání

V první verzi systému nasazené v projektu Decipher byl využit mnou navržený formát 4A. Ve druhé verzi systému byl následně nahrazen formátem Open Annotation. Oba formáty tak byly srovnány nejen teoreticky, ale i v praktickém nasazení.

Z hlediska jednoduchosti a rychlosti zpracování v anotačním systému je výrazně lepší formát 4A. U Open Annotation je třeba generovat řadu URI pro jednotlivé součásti anotace a zpracování musí mít více fází. Obtížnější jsou rovněž syntaktické kontroly, protože je zde velké množství elementů. Příkladem budiž výše uvedený popis specifikace anotovaného textu při anotování 2 slov, kde ve 4A postačují 2 fragmenty specifikované celkem 10 elementy XML, zatímco u Open Annotation potřebujeme 20 elementů, z nichž 5 je třeba identifikovat vygenerovaným URI. S tím souvisí i vyšší efektivita přenosu dat při využití 4A.

Z hlediska vyjadřovacích schopností poskytuje více možností formát Open Annotation. Kromě textu totiž umožňuje anotovat i multimediální data jako obrázky, videa apod. Nicméně v tomto ohledu by 4A bylo možné snadno rozšířit. Libovolnou anotaci ve 4A lze beze ztrát transformovat do Open Annotation.

Pro další použití výsledných anotací je výhodnější formát Open Annotation, neboť se jedná o čisté validní RDF, pro jehož zpracování existuje řada knihoven. Lze nad ním jednoduše vyhledávat a ukládat jej do specializovaných databází.

Open Annotation neumožňuje vytvořit šablony pro anotace navázané na jednotlivé typy anotací. Vytvoření šablon tak vyžaduje netriviální externí rozšíření, což může komplikovat interoperabilitu. V rámci testování jsem pro typy anotací využil specifikaci z formátu 4A.

Open Annotation (Web Annotation) se však stále vyvíjí a lze očekávat, že bude v dohledné době považován za standard a bude postupně nasazován ve větším měřítku. Nicméně implementace pokročilých nástrojů s využitím tohoto standardu a dalších souvisejících standardů, které využívá, není triviální. Jeho využití tak znamená značné náklady i vyšší režie při zpracování většího množství dat.

Vývojem a následným testováním formátu 4A jsem prokázal, že lze vytvořit formát, který může úspěšně konkurovat Open Annotation a stále je zde tedy značný prostor pro zlepšení a vývoj nových formátů anotací.

Ve výsledném systému vytvořeném v rámci této práce lze využít jak starší verzi klienta využívající 4A, tak i novou verzi využívající Open Annotation.

5.5 Protokol pro přenos anotací

Pro přenos anotací mezi klientem a serverem jsem navrhl nový protokol, který kromě přenosu samotných anotací umožňuje jednoduchou autentizaci, přenos nastavení, přenos anotovaného dokumentu (synchronizaci), přihlášení k odběru anotací z vybraných zdrojů, přenos

[10] Dublin Core Metadata Initiative http://dublincore.org/documents/dcmi-type-vocabulary/
[11] Triples in XML https://www.w3.org/2004/03/trix/

typů anotací a další potřebnou komunikaci. Navíc je možné nechat si anotace nabídnout serverem.

Oproti API, které navrhl Lee Feigenbaum ve firmě IBM [30], mnou navržený protokol předpokládá jednodušší formát anotace (bez explicitně uloženého popisu struktury) a zjednodušuje identifikaci v prostředí Internetu (URI místo GUID). Rovněž zaslání anotovaného textu na server a nabídnutí anotací je na rozdíl od tohoto API podporováno. V protokolu jsem nenavrhl kompletní správu řízení přístupu k anotacím, ale protokol je připraven pro toto rozšíření. Alternativou je i využití vyhrazeného typu atributu anotace a vyhodnocení oprávnění na serveru.

Protokol umožňuje obousměrnou asynchronní komunikaci mezi klientem a serverem. Pokud jeden uživatel zadá anotaci, server ji ihned pošle všem ostatním, kteří od daného uživatele odebírají anotace a anotují stejný dokument. Okamžitě se zasílají i aktualizace typů anotací a obsahu dokumentu. Server může kdykoliv za běhu změnit i nastavení apod.

Zprávy jsou zasílány ve formátu XML a mohou být dle potřeby sdružovány (lze zaslat více zpráv v jednom požadavku). Zprávy lze v protokolu verze 1.1 rozdělit do následujících skupin:

- správa sezení,
- uživatelé a skupiny,
- řízení odběru anotací,
- synchronizace dokumentu,
- přenos typů anotací,
- přenos anotací,
- nabízení anotací,
- práce s kontrolovaným slovníkem,
- přenos nastavení,
- chyby a varování,
- prázdná odpověď.

Popis jednotlivých zpráv v těchto skupinách je umístěn v příloze D. Zjednodušený příklad komunikace mezi klientem a serverem je umístěn v příloze E.

Po ukončení projektu Decipher byla vyvinuta 2. verze protokolu, která odstranila zjištěné nedostatky. Mezi nejvýznamnější vylepšení patří podpora alternativních nabídek anotací (viz níže), možnost rychlého nastavení odběrů anotací a vylepšená podpora kontrolovaného slovníku.

V protokolu 2.0 jsou následující skupiny zpráv (viz příloha F):

- Sezení a přihlášení uživatele,
- Uživatelé a uživatelské skupiny,
- Odběry anotací,
- Synchronizace dokumentu,

- Manipulace s typy anotací,
- Manipulace s anotacemi,
- Návrhy anotací,
- Kontrolovaný slovník,
- Nastavení,
- Chyby a varování.

Do protokolu byla přidána také podpora více editorů anotací na 1 webové stránce. Podle RFC2616[12], sekce 8.1.4 by totiž počet spojení mezi klientem a serverem měl být omezen na 2. Řada webových prohlížečů to v rámci jedné webové stránky dodržuje, což přináší komplikace, protože každý editor anotací potřebuje 2 kanály, z nichž na 1 je stále otevřené spojení. Kdyby si každý editor otevřel svůj Comet kanál, nezbyl by volný kanál pro AJAX (či pro 3. a dalšího klienta) a každá další zpráva by pak mohla být odeslána až při obnově Comet kanálu po vypršení časového limitu, což může trvat desítky sekund. Proto je nutné, aby klienti sdíleli 1 Comet kanál a na straně klienta docházelo ke směrování zpráv. Server pak musí mít informaci o tom, které klienty má připojené na daném kanálu. Za tímto účelem bylo nutné pozměnit formát některých typů zpráv.

5.6 Pokročilé vlastnosti systému

Vytvořený anotační systém ve srovnání z výše uvedených kritérií vyšel jako jeden z nejlepších a v porovnání pokročilých funkcí pro strukturované anotování vyšel nejlépe, což je patrné z tabulky v příloze B.

V této podkapitole se však nebudu zabývat pouze funkcionalitou, kterou poskytuje navíc oproti jiným systémům, ale i funkcionalitou, kterou jiné systémy nabízejí v méně pokročilém provedení, nebo je natolik zásadní, že ji zde nelze opomenout.

Mezi klíčové cíle, kterými byl motivován vývoj systému, patří [89]:

- zvýšení úrovně detailu metainformací ukotvením anotací v textu,
- umožnění práce v prostředí, na které jsou uživatelé zvyklí a využívají jej každý den,
- podpora objevování znalostí jako vedlejší efekt kolaborativního (sociálního) tagování,
- umožnění intuitivního strukturování znalostí ve formě přístupné běžnému uživateli,
- využití pokročilého zpracování textu a extrakce informací ke generování nabídek anotací a zvýšení efektivity procesu anotování.

Od výše uvedených cílů se odvíjí i základní pokročilé vlastnosti navrženého systému. K těm však přibyly i další klíčové vlastnosti, které rovněž vedou k usnadnění procesu anotování, zvýšení kvality vytvářených anotací a rozšíření možností nasazení systému:

- alternativní nabídky anotací,
- anotování při psaní textu,

[12] https://tools.ietf.org/html/rfc2616

- synchronizace pro kolaborativní anotování,
- pokročilé řízení odběrů anotací.

Níže detailněji popisuji jednotlivé pokročilé funkce.

Ukotvení anotací v textu

I přes to, že výsledné struktury anotací v RDF mohou být připojeny k celému dokumentu, je výhodné každou anotaci asociovat s konkrétním slovem, frází, větou či odstavcem, kterých se týká. Asociovanou část textu zde nazývám „fragment“.

Když přidáváme komentář, je přirozené označit část textu, ke které patří. Pokročilá uživatelská rozhraní řady dostupných služeb tuto funkcionalitu rovněž podporují (např. PLoS ONE[13] a Utopia Documents[14]). Ukotvení anotací v textu má efekt zvýšení úrovně detailu a zúžení rozsahu potenciálního využití přidané metainformace tím, že exaktně ukazuje, která část textu (fráze, věta, odstavec, ...) je anotována. Automatické nástroje pro extrakci informací z textu pak mohou tyto informace využít pro trénování a zlepšení generování dalších nabídek anotací.

V každém úzce specializovaném vědním oboru se mohou objevit nekonzistence. Ukotvení anotací v textu je základem pro budoucí ujasnění významu uvedených informací. Na anotace tak lze pohlížet jako na způsob reprezentace znalostí extrahovaných z textu [43]. Zkušenost z provedených anotačních sezení toto potvrzuje, neboť je pro určení správnosti anotace často nezbytné přihlédnout k obsahu původního anotovaného textu.

Přesně ukotvená anotace je zásadní pro pozdější strojové zpracování anotovaných dat a využitelnost výsledných anotací. Metodám pro strojové učení umožňuje vytvářet lepší modely a při indexaci a následném vyhledávání umožňuje vytvořit náhled na správnou část dokumentu. Současně umožňuje i využití kombinace dotazování v jazyce SPARQL[15] nad strukturami anotací v RDF s běžným plnotextovým vyhledáváním nad částmi textu spojenými s jednotlivými anotacemi.

Detailní metainformace umožňují identifikovat pokročilé typy vztahů mezi anotovanými zdroji. Výňatky anotovaných textů ze sady dokumentů z určité oblasti mohou poukázat na obecné vzory a jejich realizace v textu. Vědec pak může využít ukotvení v textu ke zvýraznění nejlepších postupů v určité vědní oblasti a standardizovat procedury ke kontrole konzistence a kompletnosti technických zpráv. Tím umožní automaticky porovnat přínos daného textu oproti průměru, navrhnout rozšíření apod. Takový druh kontroly je zcela zásadní nejen pro kontrolu vlastní práce ale i pro interdisciplinární studie, které musí řešit smíšenou terminologii. Seskupování společných částí textu z určité domény pak pomůže analyzovat vztahy mezi dotčenými sémantickými poli.

Specifickým způsobem využití ukotvení anotací může být i anotování důvodu, proč je určitý článek citován a jak je odkazovaný text relevantní pro daný kontext. Bylo experimentálně prokázáno, že tento druh anotací výrazně napomáhá pozdějšímu vyhledávání relevantních textů, zejména v oblastech, které jsou v daném textu zmíněny spíše okrajově a nejsou proto pokryty názvem ani abstraktem článku (viz níže).

[13] http://journals.plos.org/plosone/
[14] http://utopiadocs.com/
[15] Simple Protocol and RDF Query Language https://www.w3.org/TR/sparql11-overview/

Podpora překrývání fragmentů

Systém má netriviální podporu překrývání fragmentů a to i částečného. Za účelem dodržení validity XML dokumentů a dalších hierarchických formátů dochází k automatickému dělení fragmentů na hranicích jednotlivých elementů dokumentu a k opětovnému spojování při prezentaci vytvořených anotací. Je tedy možné anotovat např. větu: „*Renoir and Manet made copies of Delacroix' paintings*“, kde identifikujeme dvě nezávislé události se dvěma různými osobami a sdíleným fragmentem reprezentujícím aktivitu (sloveso) a objektem vztahu.

Umožnění práce v prostředí, na které jsou uživatelé zvyklí

K využití potenciálu navrženého systému bude nezbytné, aby byli všichni zúčastnění motivováni aktivně přispět anotováním svého materiálu, zdrojových dat apod. Toho nelze dosáhnout, pokud anotování nebude extrémně jednoduché. Systém 4A proto vychází z myšlenky všudypřítomného anotování. Uživatelé nemusí měnit svoje zvyklosti, aby mohli anotovat v jednotném anotačním nástroji, ale anotování bude integrováno do nástrojů, které běžně využívají.

Některé anotační scénáře vyžadují spolupráci mezi anotátory. Ta může mít formu jednoduchého znovupoužití anotací, ale může být potřeba i podpora pro současné anotování stejného textu více uživateli. Anotační sezení pak musí být plně synchronizované a jakákoliv změna, kterou provede jeden uživatel, musí být ihned promítnuta do práce všech ostatních uživatelů v dané skupině.

Z hlediska podpory anotování je pak důležité rozlišit dva typy prostředí, ve kterých mohou uživatelé pracovat:

1. prohlížeč,
2. editor.

Prohlížeč nemění text, takže postačuje přenášet pouze změny v anotacích. V editoru je však situace mnohem komplikovanější, protože změny v textu mohou zneplatnit některé anotace. Rozšíření editoru pak musí umožnit nejen synchronizaci anotací, ale i synchronizaci samotného anotovaného textu (neprovádí-li ji nějaká jiná služba).

Nasazení systému 4A do obou typů prostředí bylo otestováno s anotačním doplňkem do webového prohlížeče Mozilla Firefox a s editorem anotací pro WYSIWYG editory textů v jazyce JavaScript. Rozšíření do dalších prostředí lze provést analogickým způsobem.

Podpora objevování a strukturování znalostí

Sdílené koncepty ve formě ontologií jsou nutnou podmínkou k realizaci vize sémantického webu. I přes velkou snahu je však pokrytí vědeckých oblastí ontologiemi velmi omezené. Nedostupnost široce akceptované ontologie v oblasti počítačové vědy (s výjimkou klasifikace ACM) demonstruje daný problém. Propojení procesu vytváření ontologie s procesem anotování skrývá tuto nepříjemnou (nudnou) práci a zdá se tak jako optimální strategie k překonání tohoto problému.

Někdy je předem jasné, jaké vztahy jsou v dané doméně důležité. Jindy je však nutné pracovat s koncepty, jejichž struktura je nejasná. Vztahy se mohou objevovat postupně v průběhu kolaborativní práce na daném problému. K zachycení inovativního procesu vytváření znalostí je nezbytné, aby aplikace podporovaly práci se strukturami, které dosud nebyly jasně ustaveny.

K překonání současné praxe anotování jednoduchými klíčovými slovy (tagování) a následné organizace znalostí je nutné zavést intuitivní schéma pro strukturování znalostí a podpořit jeho instanciaci v uživatelských nástrojích. Strukturované tagy, jak je zmiňují i Bry a Kotowski [19], v systému 4A hrají tuto klíčovou roli. V níže popsaných experimentech byly úspěšně využity k popisu potřebných podmínek, konfliktních pohledů a srovnávacích vzorů. Klienti je pak prezentují ve formě stromů atributů anotací (vztahů mezi anotacemi).

Anotační systém 4A využívá konceptu sémantických šablon, které uživatele vedou skrze proces anotování. Například když anotuje text popisující vytvoření uměleckého díla, systém mu zobrazí běžné atributy z modelu CIDOC CRM[16]. Když klikne na atribut *Author*, který je obvykle vyplněn hodnotou reprezentující určitou osobu, systém v textu zvýrazní fragmenty, které odpovídají této sémantické preferenci. Šablony jsou vytvořeny v inicializační fázi při importu ontologií do systému. Efektivně se zde skrývá komplexita struktur ontologie před běžným uživatelem. Koncepty se zobrazují jako typy anotací využitelné v atributech a omezení na šablony atributů. Při jakémkoliv využití atributu je pak k dispozici odkaz, který jej váže zpět na původní ontologii. Toho pak lze využít i ke změnám a rozšiřování struktur dané ontologie.

Každá anotace se pak pomocí atributů může odkazovat na jiné anotace, které mohou existovat samostatně, nebo mohou být součástí anotace na vyšší úrovni. Systém podporuje vnořování bez omezení úrovně, takže je možné např. vytvořit anotaci události, která obsahuje anotace komplexních atributů s popisy jednotlivých aspektů dané události a současně může být součástí jiné anotace vyjadřující vztah mezi událostmi, který je dále navázán na popis stavu mysli člověka, který anotaci vytváří.

Generování nabídek anotací

Uživatelské rozhraní anotačního nástroje musí minimalizovat úsilí, které je potřebné k úspěšnému anotování obsahu. Studie ukazují, že jednoznačně pomáhá identifikace klíčových slov. Pokročilé klasifikační mechanismy využité pro nabízení anotací usnadní anotování základních kategorií a jejich zjednoznačňování s využitím kontextu a uživatel se tak může plně zaměřit na anotování specifické role, kterou entita v daném kontextu hraje. Např. slovo „Java“ lze v některém kontextu poloautomaticky anotovat jako programovací jazyk a uživatel se může následně rozhodnout, zda se jedná o jazyk, který byl využit k implementaci popisovaného nástroje, nebo zda se jedná pouze o příklad jazyka, který má pokročilý mechanismus zpracování výjimek.

Nabídky anotací lze přijímat či odmítat jednu po druhé a to přímo v textu či ve speciálním okně pro rychlé procházení a kontrolu. Pokud v daném okně najedeme na anotaci myší, zobrazí se i přímo v textu, aby bylo možné rychle zkontrolovat její kontext. Alternativou k tomuto přístupu je pak automatické potvrzení všech nabídek s určitou minimální mírou důvěry a následné manuální mazání či korekce chybných anotací. Nabídky anotací s nízkou mírou důvěry lze rovněž skrýt, aby uživatele zbytečně nerozptylovaly.

Alternativní nabídky anotací

Správnost nabídek anotací se výrazně liší dle obsahu daného dokumentu. Pro některé typy entit ve vhodném kontextu může být vysoká, pro jiné může být nízká a to zejména v případě komplexních a víceznačných popisů entit [60], [95]. Když nabídnutá anotace není korektní, uživatel ji může odmítnout a správnou anotaci vytvořit manuálně. Nicméně toto

[16] http://www.cidoc-crm.org/

zahrnuje zdlouhavý úkol spočívající ve vyhledání správné entity v kontrolovaném slovníku, případně (není-li ve slovníku) ve vyhledání odkazu na danou entitu a manuálním vyplnění všech potřebných atributů. Abychom tento nepříjemný proces zjednodušili, do systému byla zabudována podpora pro alternativní nabídky anotací. Odmítne-li uživatel (nejlépe hodnocenou) nabídku pro daný fragment, systém mu nabídne sadu alternativ, ze které si může vybrat jedním kliknutím. Není-li žádná alternativa správná, lze je jedním kliknutím odmítnout a zobrazí se další sada s nižší mírou důvěry. Počet zobrazovaných alternativ je konfigurovatelný. Uživatel si rovněž může nechat zobrazit alternativy bez odmítnutí nejlépe hodnocené nabídky, aby se mohl ujistit o správnosti jejího potvrzení. Alternativy mohou být zobrazeny také ihned po jejich vzniku. Různými způsoby práce s alternativními nabídkami se budu zabývat níže v popisu provedených experimentů.

Anotování při psaní textu

Když uživatel napíše novou část textu v textovém editoru, může tento text ihned anotovat. Text a anotace jsou zpracovány odděleně, což umožňuje lepší podporu současného editování a anotování textu. Pokročilé metody správy anotací pak garantují, že většina anotací zůstane po editaci stále validních a pouze skutečně vyřazené anotace (anotace smazaného textu) budou zneplatněny. Anotace jsou v průběhu editace průběžně aktualizovány anotačním serverem, který k tomuto účelu využívá kaskádu metod s různou citlivostí. Nejprve je obousměrně prohledáván text v uzlu, kde se nacházel původní fragment, a následně se postupuje do dalších uzlů. Fragment je považován za shodný (nalezený), pokud splňuje řadu stanovených kritérií jako jsou Levenshteinova vzdálenost textu, shoda prvního a posledního písmena, Levenshteinova vzdálenost cesty k uzlu apod. V průběhu psaní dochází i k aktualizaci nabídek anotací a než tak uživatel dopíše větu, jsou mu k ní již nabídnuty anotace zmíněných entit a dalších rozpoznatelných prvků.

Synchronizace pro kolaborativní anotování

Abychom se vyhnuli problémům při paralelní práci uživatelů a současně aby mohli lépe spolupracovat, systém má podporu pro synchronizaci změn v dokumentu, anotacích i znalostních strukturách (typech anotací se šablonami atributů) v reálném čase. Připojí-li se k práci s dokumentem jiný uživatel, může mít starší (či novější) verzi dokumentu, protože změny v dokumentu dosud nemusí být uloženy v příslušném systému pro správu obsahu (např. Drupal[17]), ve kterém je editor anotací zobrazen, nebo mohly být od zahájení anotačního sezení provedeny další změny jiným uživatelem, který anotační nástroj nevyužil. V takovém případě systém nabídne provedení aktualizace otevřeného dokumentu ze serveru, tedy načtení verze, se kterou pracují ostatní, případně spojení obou verzí dokumentu do jedné nové, která se následně zobrazí i všem ostatním uživatelům.

Pokročilá metoda generování sekvenčních čísel modifikací, strukturovaný popis modifikace a kontroly konfliktů a schvalování modifikací na straně serveru umožňují minimalizovat zpoždění mezi modifikacemi při práci většího množství uživatelů. Není-li modifikace konfliktní, není nutné čekat, až bude aplikována ve všech klientech (jako to dělá např. Google Docs[18]), ale lze ji ihned provést. Systém garantuje, že konfliktní změny budou vždy provedeny ve správném pořadí, zatímco nekonfliktní budou prováděny okamžitě.

[17] https://www.drupal.cz/
[18] https://docs.google.com/

Pokročilé řízení odběrů anotací

Pokročilé řízení odběrů anotací umožňuje definovat sady pravidel stanovujících zdroje a typy odebíraných anotací. Tyto sady nazýváme odběry a lze je libovolně kombinovat. Uživatel se k nim může přihlásit či se od nich odhlásit jedním kliknutím, přičemž přihlásí-li se k odběru, sada obsažených pravidel se zařadí mezi aktivní pravidla, seřadí podle specifičnosti a využije k filtraci anotací na serveru. Tímto způsobem si uživatel může např. zvolit, že požaduje zobrazit pouze umělce, ale druhou sadou pravidel z nich vyloučí sochaře a třetí zvolí autory, jejichž anotace považuje za relevantní. Toto lze provést separátně pro každou instanci editoru (na jedné stránce či v různých oknech nebo záložkách prohlížeče) a získat tak různé pohledy na anotovaný text, které usnadní vyhodnocení odpovídajících aspektů řešeného problému.

5.7 Případy použití

Využití vytvořeného systému není omezeno pouze na provádění experimentů v oblasti výzkumu anotování textu. Například v článku *Towards New Scholarly Communication: A Case Study of the 4A Framework* [89] bylo navrženo jeho praktické využití v oblasti publikování. Dalším případem použití pak bylo praktické nasazení v projektu Decipher a do budoucna je v plánu nasazení v pokročilém systému pro správu korpusových dat. Níže popisuji jednotlivé případy použití.

5.7.1 Nové způsoby publikování

Jak bylo uvedeno v článku [89], v posledních letech se objevují návrhy na změnu z tradičního modelu recenzí před publikováním článku a měření impact faktoru k měření sociální popularity článku [59], [2]. Umírněnější názory mají Nielsen [64] a Poeschl [70], kteří uvažují, jak využít úspěch komunitních služeb jako arXiv[19] a ACP[20] v dalších oblastech. Systém 4A je možné využít právě pro anotování zveřejňovaných článků, čímž lze podpořit životní cyklus výzkumu. Uložení anotací mimo dokument nevyžaduje vytváření řady anotovaných instancí PDF, ale lze využít např. knihovnu PDF.js[21] a článek jednoduše anotovat ve formátu HTML ve webovém prohlížeči. Rozšíření současných portálů s databázemi vědeckých článků o anotační funkcionalitu tak lze provést bez značného úsilí a potřeby speciálního programového vybavení na straně uživatelů.

Navrhovaná transformace je motivovaná výhodami, které může přinést do vědecké komunity. Na druhé straně je pak třeba zvážit obchodní modely aplikovatelné v nové publikační éře, jak uvádí Waltham [102]. Vydavatelé hledají svoji pozici ve vyvíjejícím se prostředí, což může být demonstrováno jejich snahou zapojit autory, získat jejich zpětnou vazbu a podpořit sdílení citací mezi výzkumníky ve formě služeb pro správu citací jako Connotea[22] nebo CiteULike[23].

Navržený systém vydavatelům umožní využít příležitost k získání peněz přidáním hodnoty k datům s otevřeným přístupem, spíše než omezováním přístupu k výsledkům výzkumu [36]. Budou mít možnost využít svoje zkušenosti s organizací expertů (např. recenzních týmů) schopných vyhodnotit kvalitu obsahu. S pomocí nástroje 4A mohou tyto

[19] http://arxiv.org/
[20] Atmospheric Chemistry and Physics (ACP) http://www.atmospheric-chemistry-and-physics.net/
[21] http://mozilla.github.io/pdf.js/
[22] http://www.connotea.org/
[23] http://www.citeulike.org/

kompetence rozšířit o sledování expertní práce ve specifických doménách a umožnit proces strukturování a opětovného využití znalostí. Podpoří tím práci komunity, která se zabývá danou oblastí a vyhodnocení významu jednotlivců i celých skupin na výzkum v daném tématu.

S využitím pokročilého anotačního systému dojde ke zvýraznění procesu anotování při vytváření znalostí. Anotace mohou být asociovány nejen s vědeckými texty obsahujícími výsledky výzkumu, ale i s jakýmikoliv jinými materiály, které jsou pro daný výzkum relevantní. Budou-li např. zveřejněna experimentální data popsaná v článku (jak je dnes v řadě oblastí vyžadováno), bude možné je anotovat, což umožní jednodušší verifikaci jejich interpretace v textu. Anotace dat z experimentů jiných výzkumníků také zjednoduší sledování jejich opětovného využití a poskytne tak alternativu k aktuálnímu stavu charakterizovatelnému znovupoužitím testovacích dat bez možnosti posuzování kvality, jak popisuje Enriquez [29].

Zvláštní důraz bude kladen i na sledování jednotlivých myšlenek popsaných v článcích až na úroveň zdrojového kódu programů popsaných v textu. Bude tak možné využít nejen koncept generování dokumentace ze zdrojových textů (jako nabízí např. nástroj Doxygen[24]), ale i propojit určité části kódu s jejich popisem na vyšší úrovni.

Anotační systém tedy podpoří celý životní cyklus výzkumu od myšlenky, vyhledání relevantních zdrojů a sesbírání dat, přes přípravu a spuštění experimentů, implementaci výsledného řešení a interpretaci dosažených výsledků, až po odeslání, recenzi, přípravu finální verze a publikování článku. Dále pak umožní získání zpětné vazby, diskusi a promítnutí výsledků do následující vědecké práce. Anotace na všech úrovních tak připraví cestu ke sdílenému porozumění získaným znalostem. Sociální dimenze všudypřítomných anotací znalostních artefaktů současně přinese okamžité výhody do výzkumné komunity tím, že poskytne lepší modely detailních charakteristik ovlivnění (náhradu současného měření impact faktoru).

5.7.2 Příprava výstav v projektu Decipher

Běžný proces přípravy galerijních a muzejních výstav, zaměřených na vyprávění příběhů, je takový, že kurátor výstavy nejprve dostane myšlenku či námět na novou výstavu. Dle tématu výstavy vyhledá prvotní sadu exponátů, které má k dispozici. O každém z těchto exponátů má k dispozici nějaké informace – texty, které daný exponát popisují. Na základě těchto informací vyhledá příběhy týkající se daného předmětu (stories). Dle informací v těchto příbězích pak hledá související umělecká díla a příběhy. Z příběhů následně sestavuje tzv. vyprávění (narrrative). Výstava pak obsahuje umělecká díla rozmístěná v pořadí dle vyprávění, ze kterého se stane text pro průvodce, který bude návštěvníkům výstavu komentovat.

Proces anotování může pomoci od samého počátku až po konečnou fázi sestavování výstavy. Na počátku může kurátor využít anotovaných příběhů ke snadnému vyhledání souvisejících informací. Nápomocny mu budou nejen odkazy v anotacích zmíněných osob, ale také informace zobrazované u uvedených uměleckých děl, které mohou obsahovat údaje o aktuálním umístění díla, možnostech jeho zapůjčení apod. V nalezených příbězích pak anotuje události, jejichž návaznost pomáhá najít souvislosti potřebné pro sestavení vyprávění. Provázání anotací v závěru usnadní i konstrukci celé výstavy. Anotace mohou současně posloužit jako prostředek pro komunikaci mezi kurátorem výstavy a jeho pomocníky, neboť kurátor může zdůraznit informace, které je třeba rozšířit, a jeho asistenti pak mohou vysvětlit historické souvislosti a návaznosti uvedené v doplněných textech. Anotace veřejně

[24] http://www.stack.nl/~dimitri/doxygen/

přístupných textů k výstavě od odborné i laické veřejnosti poskytne také zpětnou vazbu a umožní získání informací pro budoucí výstavy.

5.7.3 Rozšiřování ontologie

Sémantické obohacení textu je výchozím krokem různých technik pro analýzu textu, které jsou v poslední době aplikovány i na správu reputace obchodních značek, doporučování aktuálních zpráv, průzkum trhu a mnoho dalších domén. Komerční API jako IBM AlchemyLanguage[25], Cogito API[26] nebo Ontotext S4[27] automatizují úlohu sémantického obohacení textu a umožňují anotování klíčových entit a základních vztahů mezi nimi při dosažení akceptovatelné úrovně přesnosti. Ale kvalita výsledků dosahovaných pomocí plně automatických přístupů je velmi proměnlivá v závislosti na vstupních datech a specifikaci konkrétní anotační úlohy. Pro komplexní případy s mnohoznačnými jmény může být značně nízká [60], [95]. Metody využívané nejpokročilejšími nástroji se většinou opírají o strojové učení, které potřebuje trénovací data. V současnosti mohou být úspěšně aplikovány pouze na jednoduché anotační úlohy, kde jsou data dostupná, ale selhávají pro komplexní úlohy, kde není dostatek dat pro strojové učení. Ještě horším případem je pak situace, kdy struktura dat není předem známá. Potom je potřebné manuální anotování, kterým připravíme nejen trénovací datové sady, ale i samotné struktury, které budou při anotování využity.

Tento případ použití se zabývá anotováním textových zdrojů zaměřeným na návrh a vývoj znalostních struktur. Demonstruje výsledky práce v oblasti aspektově orientované analýzy sentimentu v doménách, které s výhodou využívají propojení otevřených datových zdrojů. Jeho popis začnu stručnou analýzou požadavků, která je výsledkem naší spolupráce s firmami zahrnující průzkum trhu a správu reputace obchodní značky. Následně popíšu klíčové rysy uživatelského rozhraní anotačního nástroje důležité pro zobrazení a správu znalostí vyjádřených v textech obohacených propojenými daty (Linked Data). Budu při tom klást důraz na kolaborativní návrh znalostních struktur, jejich prezentaci laikům a jejich export do standardních formátů ontologií. Následně popíšu případovou studii zaměřenou na kolaborativní práci na anotačních ontologiích aplikovatelných v aspektově orientované analýze sentimentu pro firemní prostředí ze sociálních médií, která demonstruje výhody implementovaného uživatelského rozhraní a vzorů interakce a vizualizační mechanismy, které mohou urychlit práci specialistů na průzkum trhu.

Propojená otevřená data (Linked Open Data – dále LOD) reprezentují pragmatický přístup k naplnění vize sémantického webu. Množství globálně dostupných LOD každý rok výrazně narůstá. Např. LOD Laundromat[28] v době psaní této práce indexuje více než 38 miliard trojic v RDF. Je proto přirozené položit si otázku, jak lze tato data využít v jednotlivých aplikacích, jak je lze vizualizovat a jak je rozšiřovat s pomocí uživatelů.

Nyní se zaměřím na roli LOD v aplikační doméně aspektově orientované analýzy sentimentu (ABSA – Aspect Based Sentiment Analysis [71]). Na rozdíl od obecné analýzy sentimentu, kde je cílem určení celkové polarity textu, ABSA identifikuje jednotlivé aspekty daných cílových entit a následně určuje sentiment pro jednotlivé aspekty [42], [48]. Nejúspěšnější přístupy k řešení tohoto problému zahrnují techniky strojového učení s učitelem, které pro budování svých klasifikačních modelů potřebují trénovací data [77], [96]. Manu-

[25] http://www.alchemyapi.com
[26] http://cogitoapi.com
[27] http://s4.ontotext.com
[28] http://lodlaundromat.org

ální anotování takových dat je však zdlouhavé a časově náročné – zjednodušení a zrychlení daného procesu je proto pro nás aktuální výzvou. LOD pak mohou s touto úlohou pomoci.

Při aspektově orientované analýze sentimentu se blíže zaměřím na správu reputace obchodních značek, která formuje jeden z případů použití řešených v projektu MixedEmotions[29], což je evropský projekt zaměřený na budování platformy s otevřeným zdrojovým kódem pro analýzu emocí nad velkými zdroji dat ve více jazycích. Platforma MixedEmotions je vybudována na celé řadě technologií pro extrakci informací z přirozeného jazyka, přepis, překlad apod., ale také na sémantických technologiích pro účely integrace informací na sémantické úrovni.

Typický scénář analýzy reputace zahrnuje zákazníkem definovaný zájem o určitý produkt, značku nebo oblast a poskytnutí výchozí sady slov a frází pro extrakci, která často obsahuje jen název firmy a krátký seznam produktů. Výchozí seznam je následně iterativně rozšiřován a zjemňován, aby bylo možné definovat selekční kritéria použitá pro úzce zaměřené stahování dat z webu. Sociální sítě jako Twitter[30], diskusní skupiny, internetová fóra a blogy jsou následně prohledávána na výskyt příspěvků zmiňujících výrazy ze seznamu zájmů. Následnou analýzou se určí relevance každé stažené položky, ke které je potom přiřazena polarita (pozitivní / negativní) a intenzita (významnost) sentimentu (celková a pro jednotlivé identifikované aspekty). Určují se rovněž specifické rysy osoby, jejíž názor je v datech vyjádřen.

LOD mohou poskytnout základ k naplnění slovníků či znalostních bází využitých v automatické extrakci zmínek o jednotlivých aspektech. Zmiňovanou entitou může být např. videohra *The Last of Us* a text k anotování může být následující:

> *The Last of Us is a great game. Gran Turismo 5 was a disappointment. Final Fantsy series turned to ****. Resident Evil is not horror anymore.*

LOD pak mohou pomoci s identifikací srovnávaných her *Gran Turismo 5*, *Final Fantsy* a *Resident Evil*. Stejným způsobem je lze využít i k anotování konkurence, jednotlivých součástí produktů apod.

Sémantické typy entit v LOD, které jsou zmíněny v textu, lze následně využít k nabízení názvů jednotlivých aspektů názoru. Např. pro zmínky o *Ryanair* a *EasyJet* v recenzi aerolinek lze nabídnout srovnání nízkonákladových aerolinek, zatímco názvy míst mohou být kategorizovány jako startovní a koncové body popisovaných letů. Při hierarchické klasifikaci aspektů by pak bylo možné s výhodou využít hierarchii tříd v LOD. Takové využití však zatím není příliš časté, což je způsobeno variabilitou a specificitou jednotlivých úloh extrakce názorů a aplikací, které tuto extrakci implementují. Ze stejného důvodu dle dostupných informací zůstává neprozkoumané i využití LOD k rozčlenění a zjemnění samotné analýzy sentimentu (např. identifikování obvyklých negativních vedlejších významů informací o zmíněných entitách).

Níže formulované požadavky na vizualizaci a vzory uživatelské interakce korespondují s perspektivou odborníků na průzkum trhu, kteří získávají názory z online zdrojů. Rychlost práce zde hraje klíčovou roli. Firmy vynakládají značné úsilí, aby minimalizovaly čas, který je potřebný k definování znalostních struktur relevantních pro dané studie a aby byly schopné okamžitě připravit nasazení svého řešení pro extrakci názorů do určité domény, vyladit parametry na základě prvotní zpětné vazby a odhadnout manuální práci na výsledcích interpretační fáze. Primární požadavek je tak řízený potřebou rychle určit, jak velké množství existujících LOD lze v daném konkrétním případě využít. Uvažujeme-li při

[29] http://mixedemotions-project.eu/
[30] https://twitter.com/

tom anotační proces, uživatelé preferují jednoduché vizuální prvky integrované do známých prostředí.

Vzhledem k tomu, že v projektu MixedEmotions jsou zúčastněné pouze malé a střední firmy, je kladen zvláštní důraz na těsnou spolupráci s koncovými zákazníky. Ti jsou často přímo zapojení do definice parametrů studií a určování aspektů zájmu. Je vyžadováno jednoduché sdílení informací a poskytování přístupu do primárních dat koncovým zákazníkům. Kolaborativní prostředí, které může být využito jak reprezentanty ze strany klienta, tak i ze strany poskytovatele služby, je proto vždy preferovanou volbou.

Uživatelé pro reprezentaci znalostí preferují jednoduché formáty, kontrolované slovníky, tezaury ve SKOS[31] apod. Znalostní inženýři kladou důraz na potřebu sdílení strukturálních vzorů přes všechny oblasti aplikace. Když však po koncových uživatelích požadujeme, aby naplnili či rozšířili existující znalostní struktury, je nutné využít nejjednodušší možné vzory interakce.

Obrázek 5.6: Nabídka anotace pro aspektově orientovanou analýzu sentimentu

Nyní se zaměřím na to, jak jsou výše uvedené požadavky na vizualizaci a vzory interakce podpořeny v nástroji 4A. Základní funkcionalitou je samozřejmě vizualizace anotací entit a vztahů v textu a jejich nabízení s využitím existujících znalostních struktur. Navíc zde lze rozšiřovat ontologii přímo při procesu anotování. LOD jsou zpracována v serveru pro sémantické obohacení (SEC), který generuje nabídky anotací pro prezentování uživatelům.

[31] Simple Knowledge Organization System https://www.w3.org/2004/02/skos/

K tomu, abychom uživatelům mohli zobrazit, která část LOD je potenciálně relevantní s danou sadou textů (např. recenzemi služeb) a jaké kategorie (aspekty) jsou zde pravděpodobně zmíněny, 4A využívá právě nabídky anotací.

Na obrázku 5.6 je příklad nabídky anotace ve větě z recenze restaurace. Data jsou zde automaticky naplněna z Wikipedie a DBPedie. Uživatel může nabídku potvrdit či odmítnout, případně si může zobrazit všechny potenciální alternativní entity korespondující s daným názvem (tedy např. všechna jídla z LOD).

Automatické anotování samozřejmě není bezchybné. Např. ve větě: „*I was booked on Qatar business class from Bali to Madrid*“, může být slovo *Qatar* chybně klasifikováno jako zmínka o *State of Qatar*. Po odmítnutí dané nabídky pak uživatel může využít alternativ z anotačního serveru, nebo tuto entitu anotovat manuálně.

Plně anotovaný text pak lze využít ke trénování modelů strojového učení, které budou nasazeny pro předanotování následujících textů. Navíc však uživatelé okamžitě uvidí, které třídy z ontologie jsou v anotacích využity a které jejich vlastnosti jsou důležité pro danou úlohu. Např. výrazy *first class, price, service* a *baggage drop* jsou identifikovány jako rozdílné (pod)aspekty (pokryté v příslušné ontologii), které jsou zmíněny v recenzi aerolinek na obrázku 5.7.

Relevantní aspekty sentimentu, které je třeba anotovat v online recenzích, se pro jednotlivé oblasti zájmu výrazně liší. I přes to je zde však sdílené jádro znalostních struktur a obecných subkomponent typických pro danou kategorii služeb či produktů. K definování ontologií pro určitou doménu průzkumu trhu je proto velmi výhodné znovupoužití dříve získaných vzorů. Systém 4A nabízí šablony, které uchovávají generické typy a vlastnosti, a tvůrcům ontologií tak umožňují zaměřit se pouze na doménově specifické aspekty.

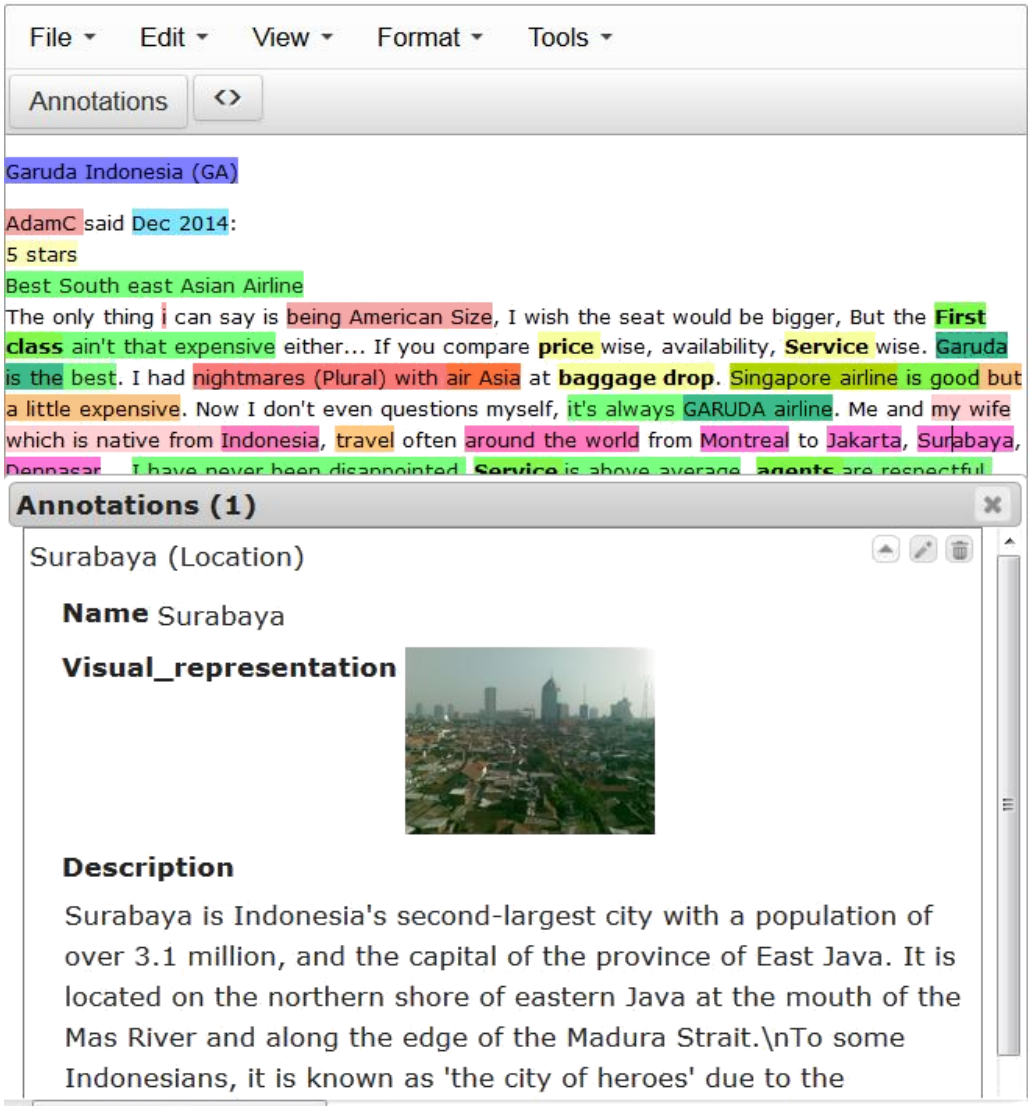
Fakt, že vytváření a rozšiřování ontologií je obvykle prokládáno s anotováním skutečných dat, přináší i další výhody. Fragmenty textu anotované tímto způsobem mohou být převzaty jako příklady použití a mohou být využity pro lepší vysvětlení přesného významu daného pojmu. Toto se ukazuje jako výhodné v situaci, kdy znalostní inženýr připravuje ontologii ve spolupráci se zástupci zákazníka a společně potřebují najít určitý společný rámec.

Laik často vnímá standardní nástroje pro tvorbu ontologií jako je Protégé[32] jako příliš komplexní na to, aby je mohl efektivně využít. Toto platí zejména ve chvíli, kdy je úkolem definovat jednoduché struktury, jakými jsou anotační ontologie. Nástroj 4A reprezentuje třídy a jejich vlastnosti jako typy anotací a jejich atributy a potenciální komplexitu vytvářených znalostních struktur v pozadí před uživatelem skrývá v maximální možné míře. Například je zde sice podpora vícenásobné dědičnosti, ale hierarchie typů je vizualizována jako strom, kde se typ může vyskytnout ve více větvích. Ukázka stromu typů anotací je na obrázku 5.8.

Zjednodušení pohledu v nástroji 4A má i další důsledky. Přesto, že nástroj podporuje import a export komplexních ontologií v OWL, preferovanou cestou je příprava jádra ontologie a složitých omezení ve specializovaném nástroji pro tvorbu ontologií a následný import do 4A, kde mají uživatelé možnost tuto ontologii rozšiřovat. Tento přístup byl zvolen i v níže popsaném experimentu.

Systém 4A nabízí kompletní podporu spolupráce. Jeho kolaborativní prostředí provádí synchronizaci textu i metadat v reálném čase. Umožňuje rovněž sdílení znalostních struktur v definovaných skupinách uživatelů. Uživatelé pak mohou pracovat primárně ve svých sadách dokumentů, ale v případě potřeby si mohou prohlížet i materiály od ostatních uživatelů

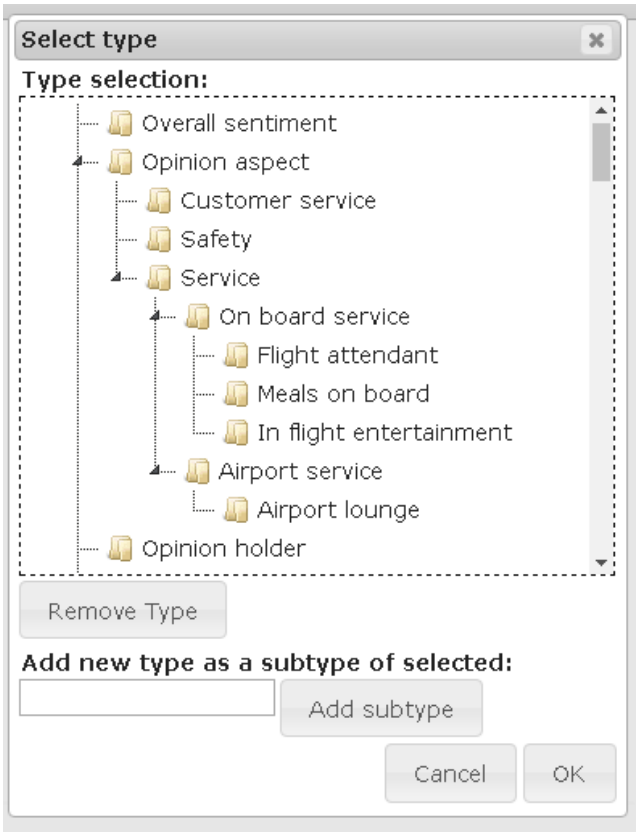
[32] http://protege.stanford.edu/



Obrázek 5.7: Recenze anotovaná ontologií pro extrakci názorů

ve skupině. To umožňuje jednoduchou efektivní spolupráci znalostního inženýra se skupinou uživatelů pracujících na konkrétním průzkumu trhu.

V experimentech níže jsou uvedeny výsledky jednoduchého experimentu srovnávajícího různé vzory interakce využívané při kolaborativní tvorbě a rozšiřování anotačních ontologií zmíněné v tomto případu použití.



Obrázek 5.8: Stromový pohled na typy anotací v editoru anotací 4A

5.7.4 Správa anotovaných korpusových dat

Moderní přístupy ke srovnávání nástrojů pro rozpoznávání pojmenovaných entit v textu, využité např. v nástroji Gerbil [75], potřebují datové sady tvořené různými předanotovanými korpusy. Výstupy z jednotlivých nástrojů jsou převedeny do jednotného formátu, srovnány s referenčním anotovaným korpusem a výsledky jsou detailně vyhodnoceny. Do srovnání pak lze zařadit nástroje využívající diametrálně odlišné přístupy (od nástrojů založených na konečných automatech se znalostní bází [24] až po nástroje využívající strojové učení) a vyhodnotit je na odpovídající datové sadě, se kterou mohou oba srovnávané nástroje pracovat.

Při vývoji moderních nástrojů pro rozpoznávání pojmenovaných entit založených na strojovém učení rovněž často potřebujeme korpusy, na kterých lze nástroje trénovat a následně i testovat. Anotované korpusy jsou potřebné i pro vývoj indexačních nástrojů, sémantických vyhledávačů a celé řady nástrojů, jejichž výzkum je nyní aktuální.

Manuální anotování korpusů je zdlouhavé a tím i značně finančně náročné. Pro větší korpusy jako je např. CommonCrawl[33] je v podstatě nemožné. Jak bylo zmíněno výše, přesnost automatických nástrojů není vyhovující. Poloautomatické anotování je však pro velké korpusy rovněž příliš náročné. Alternativou může být iterativní automatický přístup s manuálními korekcemi, které se využijí pro následující iteraci. Pro tento přístup je klíčová

[33] http://commoncrawl.org/

vhodná konzistentní vizualizace anotací v korpusu a umožnění rychlého výběru chybně anotované části, manuální opravy a následného začlenění opravy zpět do původního korpusu.

Sémantický vyhledávač, který vyvíjíme ve výzkumné skupině znalostních technologií, je založen na MG4J[34]. Vstupem celého procesu jsou různá textová data, přičemž pracujeme především s anglickou Wikipedií, korpusem CommonCrawl obsahujícím miliardy (náhodně) stažených webových stránek a příspěvky z diskusních fór. Z těchto dat je extrahován prostý text, který je následně vertikalizován a jsou z něj odstraněny duplicity. Po syntaktické a morfologické analýze jsou data zpracována nástrojem pro sémantické obohacení textu (SEC) a převedena do formátu vhodného k indexaci MG4J. V indexovaných datech pak lze vyhledávat na základě jejich sémantiky. Uvidí-li uživatel ve výsledcích vyhledávání chybné anotace, může si příslušný úryvek z dané části anotovaného korpusu otevřít v nástroji 4A. Nástroj skrze SEC umožňuje načtení anotací v libovolném formátu, tedy i v MG4J, a následně umožňuje přímý export výsledků anotování, které lze využít pro zpětnou vazbu pro SEC i pro opravu dat přímo v anotovaném korpusu. Nástroj umožňuje současné zaznamenávání provedených změn, které lze následně využít ke kumulativní aktualizaci indexu. Uživatelé tedy mohou při používání vyhledávače postupně zlepšovat jeho výsledky a vytvářet kvalitní anotovaný korpus. Jak uvádí např. Wang a kol. [103], větší množství lidí, kteří budou vyhledávač využívat a pomocí anotování mu poskytovat zpětnou vazbu, může současně vytvořit dostatek dat, po jejichž filtraci získáme dostatečně kvalitní anotace pro trénování potřebných modelů strojového učení využitelných pro další zpracování dat.

[34] http://mg4j.di.unimi.it/

Kapitola 6

Provedené experimenty

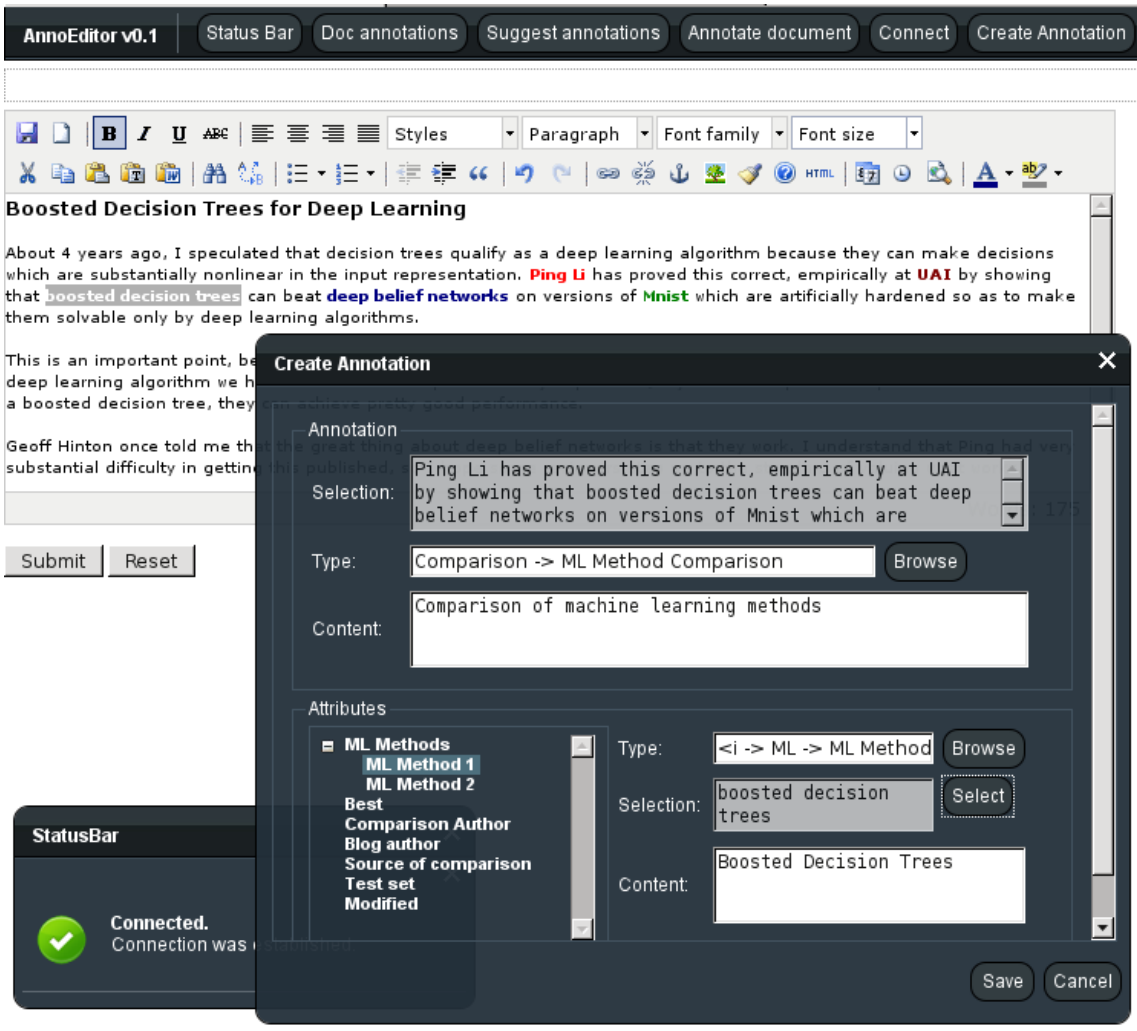
Všechny experimenty byly zaměřeny především na pokročilé anotační úlohy, jako je hierarchické anotování komplexních vztahů v textu a spojování entit s mnohoznačnými jmény a tomu odpovídajícími mnoha potenciálními položkami ve znalostní bázi. Hlavním účelem experimentů bylo sledování, jak určité aspekty uživatelského rozhraní anotačních nástrojů ovlivňují rychlost a kvalitu poloautomatického procesu anotování. Velká část této kapitoly odpovídá obsahu článků, ve kterých byly výsledky experimentů publikovány [89], [90], [27], [28].

Nejprve byly provedeny úvodní experimenty, jejichž cílem bylo vyzkoušet první verzi realizovaného anotačního nástroje a ověřit základní koncepty, jako je způsob výběru anotovaných fragmentů a kolaborativní vytváření znalostních struktur. Výsledky byly publikovány v článku *Towards New Scholarly Communication: A Case Study of the 4A Framework* [89]. Ukázka uživatelského rozhraní první verze klienta je na obrázku 6.1.

Následovalo praktické nasazení v projektu Decipher, kde sice přínos jednotlivých prvků uživatelského rozhraní nemohl být přesně hodnocen, neboť nebyl k dispozici potřebný přístup k serverům pro získávání dat (byly provozovány u jiných partnerů projektu), ani dostatečný prostor v rámci evaluace výsledného řešení, na druhé straně však poskytlo vyhodnocení z perspektivy uživatelů a jejich pohledu na prvky rozhraní. To se ukázalo jako klíčové pro vývoj druhé verze systému. Výsledky této fáze byly publikovány v článku *Advanced Features of Collaborative Semantic Annotators – the 4A System* [90].

Po skončení projektu Decipher už byly vytvořeny podmínky pro širší vyhodnocení realizovaných řešení i porovnání nástrojů v úlohách odpovídajících praktickým potřebám uživatelů. Byly provedeny dvě sady experimentů, jejichž uspořádání bude v této kapitole podrobně popsáno. Nejprve je však potřeba zasadit provedené experimenty do širšího kontextu (ne)existujících srovnání nástrojů a komponent uživatelských rozhraní sémantických nástrojů.

Jak již bylo zmíněno ve stanovení cíle práce pro tuto oblast, vysoký počet existujících anotačních systémů je v přímém kontrastu s faktem, že existuje pouze velmi málo studií, které se zabývají jejich srovnáním z hlediska funkcionality a využitelnosti pro konkrétní úlohy. V rámci projektu *Knowledge Web Project* bylo srovnáno 6 anotačních nástrojů z hledisek, jakými jsou použitelnost (instalace, dokumentace, estetika, ...), přístupnost (funkcionalita uživatelského rozhraní) a interoperabilita (platformy a formáty dat) [57]. Většina těchto hledisek je mimo rozsah mojí práce, ale některá z nich jsou pokryta ve srovnání nástrojů v příloze A, které se zabývá vybranými charakteristikami anotačních nástrojů z hlediska použitelnosti pro komplexní anotační úlohy.



Obrázek 6.1: Ukázaka uživatelského rozhraní 1. verze klienta

Diana Maynard pak ve svém článku [56] srovnává anotační nástroje z perspektivy manuálního anotátora, uživatele vytvořených anotací, tvůrce korpusů a vývojáře systémů. Opět při tom zkoumá kritéria jako použitelnost, přístupnost a interoperabilitu, ale navíc studuje i škálovatelnost (s jak velkými daty lze pracovat v návaznosti na interakci uživatele) a flexibilitu, tedy připravenost na situaci, kdy při vývoji nástroje není znám typ uživatele ani jím řešené úlohy. Přestože je tato studie více než 7 let stará, mnoho z uvedených kritérií je stále aktuálních. Proto tato studie částečně ovlivnila moji následující práci.

Wolfe ve svém článku [108] srovnává anotační nástroje z technického hlediska (zda lze využít myš i klávesnici, typ uživatelského rozhraní, anotovaný text, typ kotvy v textu a úložiště pro ukládání anotací) a popisuje jednotlivé související modely interakce. Obdobnou studii pak nabízí i Ovsianikov, Arbib a Mcneill [66], kteří mimo jiné sledují možnost sdílení, tvorby odpovědních vláken či psaní rukou, a Glover, Xu a Hardaker [33], kteří se zaměřili na aspekty jako možnost vytvářet privátní anotace, potřeba dalšího programového vybavení, integrovatelnost do prostředí, možnost vytváření grafických anotací apod. I když jsou některé zkoumané aspekty v těchto studiích relevantní (např. otevřenost licence, vícenásobné

odkazy a synchronizace), většina aspektů je příliš technicky zaměřených, neposkytuje informace pro srovnání možností vytváření strukturovaných anotací a např. vyhodnocení, zda lze využít myš i klávesnici, případně mikrofon, se v době širokého rozšíření dotykových zařízení zdá jako nerelevantní. Rovněž srovnávané nástroje v prvních dvou uvedených studiích nejsou zcela aktuální a relevantní.

Yee sestavil tabulku srovnávající vybrané existující nástroje pro anotování webových stránek. Jejich nevýhodami byl motivován k vývoji nového nástroje CritLink [109], který umožňuje připojit anotaci k libovolnému místu na libovolné veřejně přístupné webové stránce. Na rozdíl od mojí práce se jeho srovnání zaměřuje pouze na základní anotační úkoly, a to spíše z technického hlediska než z pohledu uživatelského rozhraní.

Reeve a Han [73] ve své studii srovnávající platformy pro sémantické anotování zabývají výkonností komponent pro předanotování textu na pozadí (tedy pro tvorbu nabídek anotací). Vzhledem k tomu, že řada moderních anotačních nástrojů může libovolně měnit služby pro extrakci informací v pozadí, přičemž si lze vybírat z celé řady existujících systémů, je toto srovnání a další obdobné studie relevantní pouze z historického hlediska.

V průběhu své práce jsem neobjevil žádnou studii, která by se zabývala návrhovými vzory využitými v nástrojích pro sémantickou anotaci textu, relevantní pro anotování komplexních událostí a zjednoznačňování mnohoznačných entit. Proto jsem se v dalších experimentech zaměřil právě na tento druh srovnání.

Nejprve je nutné vysvětlit, co bylo v experimentech popsaných níže bráno jako komplexní událost. Lidé pojem *událost* intuitivně chápou jako „*kdo kdy co komu udělal a kde*“. Ale definice tohoto termínu v článcích o extrakci událostí se mění dle zaměření jejich autorů, kteří u nich studují rozdílné aspekty v závislosti na řešeném problému. Například událost v biomedicinském dolování informací z textu (text mining) může mít význam „*změna stavu molekuly či molekul*“ [47]. Já se však budu držet intuitivní neutrální definice události a pracovat s ní jako se situací nebo akcí, která nastala nebo se vyskytla na určitém místě v určitém čase. Událost tedy bude mít následující atributy:

- aktor (actor),
- aktivita (activity),
- místo (location),
- čas počátku (start time),
- čas konce (end time).

Totožné schéma bylo využíváno i v projektu Decipher[1], kde však byly navíc další atributy umožňující detailnější popis události vzniku uměleckého díla:

- object (objekt kulturního dědictví),
- style or movement (umělecký styl či směr),
- genre (žánr),
- theme (téma, např. výjev na obraze),
- dimensions (rozměry),

[1] http://decipher-research.eu/

- value (finanční hodnota),
- material (materiál, ze kterého bylo dílo vyrobeno).

Abychom zůstali u scénáře dolování informací z textů, událost bude dále reprezentována jako komplexní kombinace vztahů (reprezentovaných atributy) vázaných na empirická pozorování v textu [41].

První sada experimentů, která byla provedena s verzí systému vyvinutou po ukončení projektu Decipher, se zabývá srovnáním tří nástrojů pro sémantické anotování textu – 4A, vyvinutého v rámci této práce, RDFaCE [46] a Gate Developer [23] – na úloze hierarchického anotování událostí a komplexních vztahů v textu. Tyto nástroje byly zvoleny, protože vyšly jako nejlepší ze srovnávacích tabulek v přílohách A a B. Proces anotování se skládal z volby části textu korespondující s událostí specifického typu, vyplnění atributů (slotů) entitami a hodnotami zmíněnými v textu. Entity je při tom nutné zjednoznačnit jejich spojením s referenčními zdroji (převážně Wikipedia[2] a DBPedia[3]).

Zbývající experimenty, provedené ve stejném období, využívají nástroj 4A ke studii srovnávající různá nastavení uživatelského rozhraní anotačního nástroje a souvisejících vzorů interakce uživatelů. Ukazuje se, že praxe běžně využívaná v anotačních nástrojích, která požaduje zjednoznačňování entit pouze na základě typu a zobrazovaného URL, vede k nízké kvalitě výsledků. Opačný přístup, kdy uživatelé okamžitě dostanou rozsáhlé informace o daných entitách potenciálně korespondujících s daným víceznačným jménem, však vede nejen k delšímu času anotování, ale překvapivě ani nezvyšuje kvalitu vytvářených anotací. Je tedy potřeba najít určitý kompromis mezi nedostatečnými a příliš rozsáhlými informacemi. Takové optimální nastavení parametrů bylo v rámci experimentů vyhledáváno.

Druhá sada experimentů této série se potom zabývá hledáním optimálního množství zobrazovaných informací a formy jejich prezentace k dosažení nejlepších možných výsledků zjednoznačňování entit s víceznačnými jmény. Poslední sada následně prokazuje, že sémantické filtrování implementované v nástroji 4A zrychluje proces anotování událostí a přináší vyšší konzistenci než alternativní přístupy. Výsledky výše uvedených experimentů byly publikovány v článku *Interaction Patterns in Computer-Assisted Semantic Annotation of Text – An Empirical Evaluation* [27].

V další sadě experimentů pak bylo blíže zkoumáno sémantické filtrování a optimalizace množství zobrazovaných informací a byly zkoumány přínosy alternativních nabídek anotací, aby bylo možné zodpovědět některé otevřené otázky z předchozích experimentů. Výsledky byly přijaty k publikování v knize *Springer Book of ICAART 2016* ze série *Lecture Notes in Computer Science* [28].

Poslední experiment, který bude popsán v této kapitole, se zabýval kolaborativním vytvářením ontologie pro aspektově orientovanou analýzu sentimentu. Výsledky tohoto experimentu dosud nebyly publikovány – článek na konferenci je momentálně v rozpracovaném stavu. V rámci projektu MixedEmotions k němu budou doplněny další výsledky.

[2] https://en.wikipedia.org/
[3] http://dbpedia.org/

6.1 Testování prvotního prototypu systému

Jako první byl připraven anotační experiment s 1 docentem a 6 studenty doktorského studia, zaměřený na přípravu metadat pro získávání znalostí z vědeckých článků a souvisejících materiálů. Práce byla motivována návrhy na změnu tradičního modelu publikování vědeckých článků (viz výše). Připravené texty k anotování obsahovaly vědecké články, zprávy z blogů a Twitteru[4] relevantní pro dané téma a další materiály odkazované z těchto zdrojů (prezentace, části datových sad, zdrojové texty programů apod.). Jednalo se převážně o materiály z oblasti zpracování přirozeného jazyka, extrakce informací z textu a strojového učení (odpovídající zaměření účastníků experimentu). Byla zde však diskutována i některá obecná témata (především v textech z blogů).

Kvůli omezenému počtu účastníků experimentu jsem vyhodnocoval výsledky pouze kvalitativně a následně využil k vylepšení editoru anotací. Ukázalo se, že efektivitu anotování ovlivňuje velké množství parametrů, mj. způsob vizualizace, podpora editace a kontextově závislá kvalita nabídek anotací. Výsledky tohoto experimentu není snadné zobecnit, ale poskytly hodnotný náhled na řešený problém a bylo z nich možné identifikovat oblasti, na které jsem se zaměřil v dalších experimentech.

První sada experimentů byla zaměřena na identifikování relevantních částí textu (výběr fragmentů), které korespondují se specifickým typem vzorů obsahu (např. tvrzení ukazující ekvivalenci dvou přístupů nebo doporučující zdroj pro určitou skupinu čtenářů). Součástí úlohy bylo také anotování fragmentů poskytující záznamy o tom, zda byla určitá metoda opravdu použita pro danou úlohu a ne pouze uvedena jako alternativa, přičemž výzkumník s ní ve skutečnosti nepracoval. Pro citace pak bylo úkolem identifikovat část odkazovaného textu ilustrující odkazující kontext.

Primárním úkolem pozorování v první sadě experimentů bylo zjistit, jakou formu anotací uživatelé preferují v daných situacích (jednoduché či strukturované) a jak jim pomáhají nabídky anotací (i když jejich kvalita není zdaleka ideální). Nabídky anotací založené na automatické extrakci informací byly nejprve vypnuty, aby neovlivňovaly výsledky pozorování pro první část řešeného problému. Toto nastavení simulovalo specializované anotování, kde je k dispozici pouze značně omezená sada anotovaných příkladů a není tak možné využít metody založené na strojovém učení, a současně příliš komplexní na to, aby ji bylo možné popsat sadou pravidel v rozumném rozsahu. Účastníci byli instruováni, aby anotovali tak, aby jejich anotace byly v budoucnu využitelné pro získávání znalostí z anotovaných textů. Nabízení anotací pak bylo zapnuto až pro 2. část experimentu.

Výsledky první části experimentu zaměřené na preferovanou formu anotace jsou jednoznačné. Všichni účastníci se při anotování srovnání metod rozhodli pro strukturované anotace. Pro anotování jednotlivých metod se však rozhodli využít pouze jednoduché anotace (např. „conjugate gradient“ anotovali jako „ML method“ (metoda strojového učení)). Jako vedlejší efekt pokročilé funkcionality nástroje bylo rovněž možné pozorovat, že forma anotací velmi pravděpodobně zůstane nezměněná, když jeden uživatel definuje strukturu anotace pro určitou úlohu a ostatním postačuje ji pouze využít (potřeba redefinice struktury je nepravděpodobná).

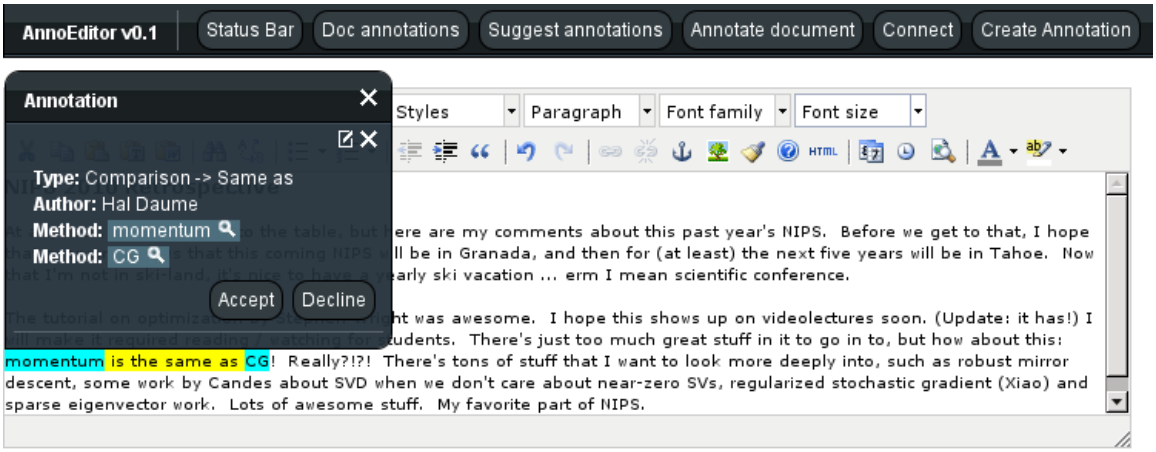
Otázky položené ve druhé části experimentu se ukázaly jako nejednoznačné. Automatické nabízení anotací zrychlilo proces anotování, ale také rozptýlilo pozornost anotátorů. Úroveň akceptování nabídky je subjektivní a výrazně se mění v závislosti na přesnosti procesu extrakce informací. Ukázalo se také, že je sporné, jaká by měla být granularita nabízení strukturovaných anotací, tedy jaké informace by do nich měly být kombinovány a poten-

[4] https://twitter.com/

ciálně potvrzeny či odmítnuty jedním kliknutím, aby nebylo potřeba příliš mnoho korekcí. Např. jeden z anotátorů poukázal na to, že slovo „momentum“ by nemělo být označeno jako metoda v kontextu „momentum is the same as CG!“[5], což by pravděpodobně mělo vést k odmítnutí celé nabídky anotující vztah „same as“, ale stále by mělo být možné přijmout nabídku označující „CG“ jako metodu.

Obecná motivace pro první sadu experimentů (využití výsledků pro budoucí sémantické vyhledávání a proces získávání znalostí) se také ukázala jako nejednoznačná. Účastníci totiž měli problém se rozhodnout, jak modelovat některé vztahy, když účel vytvářené anotace nebyl předem detailně specifikován. Dále byl kritizován fakt, že identifikace části textu pro kontext citace je často obtížná. Např. ve výše uvedeném případě jsou jako zdroje využity pouze slidy z prezentací a má-li tato část být navázána na jiný dokument, je nutné hlubší porozumění řešenému problému, neboť termín „momentum“ se nevyskytuje nikde jinde v daném textu. Obdobně spojení zkratky „CG“ s plnou formou „Conjugate Gradient“ v textu nenalezneme a musí být explicitně uvedeno anotátorem.

Druhá sada experimentů byla zaměřená na kolaborativní aspekt anotačního procesu. K ujasnění účelu anotování (aby se neopakoval problém z 1. experimentu a byla redukována vágnost zadání) byl omezen rozsah anotování pouze na texty srovnávající 2 a více metod, struktur nebo zdrojů a identifikaci „vítěze“, případně anotování shody, jak je patrné z obrázku 6.2.



Obrázek 6.2: Strukturovaná anotace shody dvou metod

Nastavení experimentu umožnilo studium konvergence znalostních struktur využitých anotátory. Tedy cílem bylo zodpovědět otázku, zda anotační systém podporuje objevování znalostí a pomáhá tak porozumět vědeckým tématům společnou prací na vytváření znalostních struktur. Vedlejším cílem analýzy bylo určení úrovně jednotnosti typů strukturovaných anotací a názvů atributů.

Odpovědi na položené otázky byly jednoznačně pozitivní. Uživatelské rozhraní editoru anotací simulovalo znovupoužití existujících datových struktur (typy anotací i názvy atributů). Protože je jednodušší využít struktury definované ostatními, a případně si je přizpůsobit, než vytvářet vlastní, anotátoři byli dostatečně motivovaní vyhledávat a sdílet konceptualizaci znalostních struktur.

[5]NIPS 2010 Retrospective http://nlpers.blogspot.cz/2011_01_01_archive.html

Podpora kolaborativního strukturování znalostí však byla shledána nedostatečnou. Anotátoři by totiž ocenili asistenci při změnách znalostních struktur a podporu pro diskutování struktury přímo u dané struktury. Ocenili by také možnost zobrazení kontextu, ve kterém ostatní anotátoři strukturu aplikovali. Toho lze docílit mimo jiné i začleněním editoru anotací do vhodného prostředí, jakým může být např. redakční systém Drupal[6].

Obecné komentáře k experimentu také ukázaly, že ani detailní specifikace cíle anotování nevedla ke zjednodušení rozhodování o strukturách znalostí. Nebylo jasné, zda by mělo být při anotování použito obecné schéma anotace pro porovnání nebo specifická schémata pro určité podmnožiny. Potom je však otázkou, zda detailní specifikace zamýšleného využití anotací odráží reálné scénáře, kdy je často potřeba, aby byly texty anotovány, aniž bychom znali všechna potenciální budoucí využití. Z perspektivy uživatelského rozhraní pak anotátoři kritizovali fakt, že neměli dostatečnou kontrolu nad atributy, které jsou nabízeny v rámci šablon anotací. Někteří se cítili příliš vázáni znalostními strukturami, které vytvořili ostatní, a stěžovali si na nedostatečnou možnost diskuse s nimi.

6.2 Nasazení v projektu Decipher

Pro demonstraci možností reálného využití vybraných funkcí systému zde uvedu příklady z domény kulturního dědictví, které korespondují s nasazením systému v projektu Decipher[7]. Decipher byl projektem Evropské komise (v rámci 7. rámcového programu pro výzkum a technologický rozvoj) zaměřeným na podporu objevování a zkoumání kulturního dědictví skrze vyprávění příběhů. Sémantické obohacení textů přineslo nový rozměr do celého procesu konstrukce vyprávění, vizualizace znalostí a zobrazení pro muzejní profesionály zapojené v projektu.

Níže uvedu tři příklady, ve kterých nelze plně využít automatické rozpoznání informací v textu a které vytvořily prostor pro nasazení anotačního systému 4A v projektu Decipher.

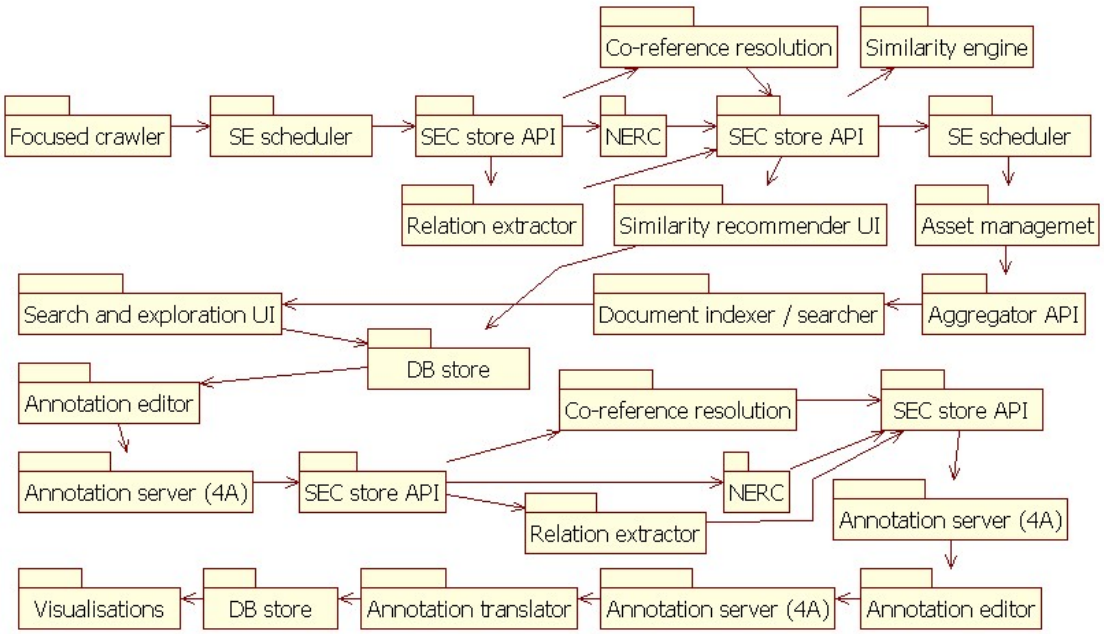
První příklad vychází z faktu, že konstrukce přirozeného jazyka pro vyjádření sémantických vztahů jsou značně variabilní a často nemáme dostatek dat pro natrénování modelů pro metody strojového učení. V projektu Decipher byl studován např. vztah ovlivnění umělců (mezi umělci, díly, tématy, styly, technikami, místy apod.) a ukázalo se, že přes veškerou snahu některé výrazy jako „svým dílem vzdává hold“ nejsou zcela pokryty. Přesto, že je zde řada přístupů využívajících rozsáhlé datové sady z webu (viz např. StatSnowball [113]), stále potřebujeme určitou minimální, uživatelem poskytnutou sadu příkladů, na základě které bude možné vytvořit automatické metody. Manuální anotování tak zde slouží pouze jako zdroj trénovacích dat a umožňuje tak vylepšit výsledky automatických anotačních procesů.

Ve druhém příkladu zmíním situaci, kdy je struktura znalostí (šablona, kterou je třeba vyplnit) komplexní a metody pro zpracování přirozeného jazyka a strojové učení s ní nejsou schopny dostatečně spolehlivě pracovat. V doméně kulturního dědictví je typickým příkladem znalostní schéma popisující různé postoje k umělci a jeho dílu. O takovém tématu může být napsáno mnoho knih a lidé mohou mít různé názory, které jsou často protichůdné. V takové situaci je pro metody strojového učení velmi obtížné generalizovat dostupné informace a natrénovat využitelný model. Komplexní struktury se však často skládají z podčástí, které lze rozpoznat automaticky. Nabízení anotací pak může značně urychlit proces sémantického anotování.

[6] https://www.drupal.cz/
[7] http://decipher-research.eu/

Posledním příkladem je situace, kdy může být i samotná struktura znalostí nejasná či těžce srozumitelná. Při anotování určité části relevantních textů si uživatelé často uvědomí přítomnost obecných sémantických vzorů znalostí reprezentovaných daným textem. Jsou schopni určit, které vlastnosti jsou důležité pro řešenou úlohu, a mohou jednoduše navrhnout schéma pro reprezentaci těchto znalostí, které odráží jejich specifické potřeby. Tento aspekt se v projektu Decipher ukázal velmi častým. I přes to, že byly v maximální míře využity existující znalostní struktury jako např. široce zavedené ontologie a konceptuální hierarchie CIDOC CRM[8], Getty Thesaurus[9] apod., mnoho činností vyžadovalo specifické struktury znalostí, které bylo třeba vytvářet až v průběhu práce muzejních profesionálů. Nelze očekávat, že se koncový uživatel naučí pracovat se sofistikovanými nástroji jako Protégé[10], aby si mohl doplnit chybějící části znalostního schématu (využité ontologie). Je tedy velkou výhodou, když anotační nástroj tuto funkcionalitu poskytne, a to v jednoduché intuitivní formě, která nevyžaduje složité zaškolení.

Mnou navržený anotační nástroj splňuje výše popsané potřeby. Jeho nasazení v projektu tak značně usnadnilo řadu činností, které uživatelé v muzeích běžně provádějí při přípravě výstav (viz výše). Způsob integrace ve výsledném systému projektu Decipher je patrný z diagramu balíčků v příloze G. Pracovní postup (Workflow) využívaný v Decipher je na obrázku 6.3.



Obrázek 6.3: Pracovní postup (Workflow) v projektu Decipher

Focused crawler (komponent pro specializované zaměřené stahování) nejprve stáhne data z webu. Ta jsou následně zpracována komponenty zajišťujícími automatické sémantické obohacení (předanotování) a uložena do databáze skrze *Aggregator API* od firmy System

[8] http://www.cidoc-crm.org/
[9] http://www.getty.edu/research/tools/vocabularies/
[10] http://protege.stanford.edu/

Simulation[11]. Následně jsou data zaindexována a lze v nich vyhledávat skrze *Search and exploration UI* . V něm uživatel nalezne počáteční sadu dokumentů, se kterou bude pracovat. Na základě dokumentů, které uživatel označí za relevantní, jsou mu následně nabízeny související dokumenty. Vyhledaná data pak anotuje nástrojem 4A, přičemž jsou výsledné anotace ihned exportovány do indexu v *SEC store API*. SEC následně na základě vytvořených anotací vylepší nabídky anotací, které jsou opět zobrazeny uživateli v editoru anotací. Vybrané typy anotací pak anotační server exportuje do *StoryScope* (uživatelské rozhraní bylo pojmenováno *StorySpace a celá daná část systému se později přejmenovala na StoryScope* – na obrázku v příloze G je původní jednodušší verze diagramu). Skrze *Annotation translator* jsou pak vybrané typy anotací transformovány na stránky příslušných historických událostí, objektů kulturního dědictví apod., které jsou uloženy do databáze a lze je vizualizovat uživatelům. Výsledky jsou pak současně exportovány skrze *Aggregator API* a mohou je tak vyhledat profesionálové z jiných muzeí (preferovaný způsob nasazení je, že každé muzeum má svůj *StoryScope* a *Aggregator* a SEC jsou poskytovány centrálně). Detailní informace jsou uvedeny rovněž v technických zprávách projektu Decipher [91], [92].

6.3 Výběr textů pro pokročilé experimenty

Texty k anotování pro experimenty, provedené po skončení projektu Decipher, byly vybrány z obecných webových stránek korpusu CommonCrawl[12] z prosince 2014. První experiment, srovnávající anotační systémy, spočíval v anotaci událostí. Volba textu nesledovala žádné zvláštní cíle (na rozdíl od dalších experimentů), takže byly jednoduše vyhledány texty obsahující zmínky o pojmenovaných entitách automaticky rozpoznatelných použitými nástroji NER (named entity recognizers) a obsahující také slovo (nejčastěji sloveso) vyjadřující cestu či návštěvu osoby na různých místech. Data byla následně manuálně anotována dvěma anotátory, neshody byly vyřešeny a několik málo nejasných případů bylo z datové sady odstraněno.

Pro druhou sadu experimentů, zaměřenou na optimální množství zobrazovaných informací, bylo nutné připravit datovou sadu obsahující víceznačná jména s proporcionální reprezentací dvou či více alternativních entit. Jako inspiraci bylo možné využít projekt WikiLinks[13] a v korpusu vyhledali data pro případy, kdy lze pod stejným jménem nalézt dva nebo více různých odkazů na Wikipedii. K vyfiltrování potenciálních závislostí mezi dostupnými možnostmi a umožnění zaměřit se na klíčové atributy byly v první části experimentu zvoleny páry textů obsahující jméno sdílené dvěma různými entitami. Ve výsledných datech se tak například objevují následující věty:

1. *Charles Thomson was a Patriot leader in Philadelphia during the American Revolution and the secretary of the Continental Congress (1774–1789) throughout its existence.*

2. *Charles Thomson's best known work is a satire of Sir Nicholas Serota, Director of the Tate gallery, and Tracey Emin, with whom he was friends in the 1980s.*

Menší část (34 sad) dat obsahovala věty s názvy, které korespondují s pěti a více entitami v anglické Wikipedii. V sedmi takových případech byla datová sada dále rozšířena o text,

[11] http://www.ssl.co.uk/
[12] http://commoncrawl.org/
[13] http://www.iesl.cs.umass.edu/data/wiki-links

který obsahuje stejné víceznačné jméno, které však koresponduje s entitou, která ve Wikipedii není pokryta. Takto vytvořená datová sada byla využita v druhé části experimentu (viz níže).

Pro poslední sadu experimentů, kombinující zjednoznačňování entit a anotování událostí, byly zvoleny odstavce textu s větami obsahujícími víceznačná jména, pokrytá Wikipedií, a odpovídající klíčové slovo reprezentující určitý typ události. Obdobně jako při přípravě dat pro první experiment byla data sestavena z případů, kdy proces manuálního předanotování vedl ke shodě obou anotátorů. Ve finále bylo pro experiment využito pouze 12 textů zmiňujících události.

6.4 První série pokročilých experimentů

První série provedených experimentů byla zaměřena na následující zkoumané otázky:

1. Jak jednotlivé prvky uživatelského rozhraní (rozhodnutí při návrhu jednotlivých nástrojů) ovlivňují kvalitu výsledků a rychlost anotování?

2. V jakém rozsahu množství zobrazovaných informací ovlivňuje výsledky zjednoznačňování entit?

3. Je pro uživatele výhodou znalost potenciálních alternativních anotací a poskytnutí míry důvěryhodnosti jednotlivých nabídek anotací?

4. Jaké výhody do procesu anotování přináší koncept sémantického filtrování?

Úvodní experiment z této série byl zaměřen na první uvedenou otázku. Bylo provedeno srovnání různých uživatelských rozhraní a vzorů interakce reprezentovaných třemi různými anotačními systémy. Při tom byly zkoumány různé funkce, které mohou ovlivnit výkonnost a kvalitu při procesu anotování. U některých nástrojů není žádný viditelný rozdíl mezi automaticky vytvořenými anotacemi z procesu automatického předanotování a anotacemi, které manuálně vytvořil uživatel. Jiné nástroje naopak striktně odlišují nabídky anotací od anotací, které uživatel schválil a potvrdil či manuálně vytvořil. Toto může mít zásadní vliv na kvalitu anotací (např. tím, kolik anotací uživatel opomene překontrolovat).

Vzory anotací událostí a dalších komplexních vztahů a jejich atributů, které nástroje poskytují, se pro jednotlivé nástroje také liší. Pokročilé nástroje umožňují definovat sofistikované šablony a omezení typů, kterými lze vyplnit jednotlivé atributy, zatímco jednodušší nástroje mohou nabízet pouze lineární seznamy atributů s textovými hodnotami. Systémy se samozřejmě liší i v obecném přístupu k implementaci práce s těmito vzory, což opět může ovlivnit výkonnost uživatelů.

U různých hodnot, které uživatelé zadávají, je často potřebné, aby korespondovaly s entitou z kontrolovaného slovníku či jiným seznamem potenciálních entit. Příkladem může být URL spojující zmínku o dané entitě s referenčním zdrojem (např. Wikipedia či DBPedia). Nástroje mají různou podporu pro zadávání tohoto typu hodnot. Některé (např. 4A) využívají automatické doplňování prezentující nejen URL, ale i další informace, které uživateli pomohou s výběrem správné hodnoty, jako je celé jméno osoby a stručný popis sloužící jako zjednoznačňovací kontext. Jiné nástroje však nabízejí pouze automatické doplňování samotné URL (např. RDFaCE). Jak je uvedeno níže, tento aspekt může rovněž ovlivnit výsledky anotování.

Další dvě z uvedených otázek jsou pokryty druhou sadou experimentů. Je jasné, že množství zobrazovaných informací a způsob jejich zobrazení mohou ovlivnit rychlost a přesnost

anotování. Jsou-li zobrazené informace pro rozhodnutí nedostatečné, uživatel musí dohledat doplňující informace, což jej při práci bude zdržovat. Naopak pokud nástroj zobrazí příliš mnoho informací, rychlost anotování se může opět snížit, neboť uživatel stráví čas čtením nadbytečných informací.

Většina ze srovnávaných systémů (viz příloha A) zobrazuje URL stránky v referenčním zdroji, kterou uživatel může prozkoumat vždy, když si není jistý, že odkazované informace korespondují se zmíněnou entitou. Toto sice může mírně urychlit proces anotování, ale také to může vést k větší náchylnosti k chybám. Nástroj 4A umožňuje filtrování zobrazovaných informací a nastavení způsobu jejich zobrazení. Detailní atributy mohou být sbaleny a zobrazeny pouze v případě, když je uživatel bude potřebovat a klikne na dané tlačítko. Tím je umožněno otestování a srovnání různých nastavení.

Bez podpory systému se uživatel nemusí dozvědět, že je dané jméno víceznačné. Pro časté výskyty převážně využívaného významu to nezpůsobuje žádné problémy. Ale pokud uživatel není expertem na danou oblast a jsou zde dva nebo více potenciálních významů, může jednoduše potvrdit nabídnutý nesprávný význam. Toto riziko lze snížit tak, že nástroj uživatele upozorní na existenci alternativ. Otázkou je, jak má vypadat optimální konfigurace této funkcionality. Tedy zda se mají alternativy vždy zobrazit ihned, nebo zda mají být zobrazeny např. pouze v případě, kdy si o ně uživatel zažádá, nebo kdy je míra důvěry nižší než prahová hodnota, či při malém rozdílu mezi mírou důvěry nabídnuté anotace a druhé nejlepší nabídky. Tyto aspekty byly zkoumány v rámci provedeného experimentu.

Cílem třetí sady experimentů pak bylo zodpovězení poslední položené otázky zaměřené na roli sémantického filtrování v procesu anotování. Obvykle není jednoduché vytvořit anotace komplexních vztahů, jakými jsou např. události. Pokročilé mechanismy napomáhající s vyplněním jednotlivých atributů mohou vést ke zrychlení anotačního procesu a zlepšení výsledků. Nástroj 4A umožňuje vytváření hierarchických anotací, přičemž zvýrazňuje potenciální kandidáty pro vyplnění atributů (vytvoření odkazů na anotace), je-li znám typ či nadtyp požadované hodnoty atributu. Níže uvedený experiment zkoumá dopad této funkcionality na proces anotování.

6.4.1 Srovnání nástrojů

Cílem tohoto experimentu bylo porovnat pokročilé anotační nástroje z hlediska funkcí uživatelského rozhraní a vzorů uživatelské interakce, které mohou ovlivnit kvalitu výsledků a rychlost anotování. Konkrétně zjistit, zda se bude lišit kvalita výsledků anotování získaných různými nástroji. Kvalita byla měřena jako kompletnost a korektnost typů entit vyplněných v atributech komplexních vztahů a vazeb na referenční zdroje (převážně Wikipedii[14] či DBPedii[15]). Navíc byl každému uživateli měřen čas na každý experiment a následně průměrován na každý vyplněný atribut.

Využité nástroje reprezentují různé přístupy k vytváření komplexních anotací. Nástroj 4A, navržený v rámci této práce, klade zvláštní důraz na hierarchické anotace a potenciální překrývající se fragmenty textu. Uživatelé mohou využít pokročilé nabízení anotací a jednoduchý mechanismus pro výběr správných hodnot atributů jednoduchým přijímáním poskytnutých nabídek anotací.

RDFaCE je obdobný jako 4A v tom, jak jsou anotovány fragmenty textu a v tom, jakým způsobem jej lze nasadit – tedy jako zásuvný modul do WYSIWYG textového editoru

[14] https://en.wikipedia.org/
[15] http://dbpedia.org/

TinyMCE[16]. Rovněž poskytuje možnost předanotování textu, ale na rozdíl od 4A neumožňuje rozlišit mezi automaticky vytvořenými anotacemi a anotacemi, které vytvořil uživatel. Také zde není žádný jednoduchý způsob, jak anotovat dvě překrývající se části textu dvěma rozdílnými anotacemi (událostmi). Proto bylo testerům povoleno usnadnit si práci a zvolit celé věty či odstavce jako fragmenty korespondující s danými událostmi.

Existuje celá řada rozšíření a zásuvných modulů pro GATE, které by pro anotační experimenty bylo možné využít. Jedním z nejpokročilejších je např. GATE Teamware[17], což je webový systém pro kolaborativní anotování textu. Tento však zatím překvapivě nemá podporu pro anotování vztahů a koreferencí, což ve svém článku uvádí i Bontcheva a kol. [12]. Z obdobného důvodu byla z experimentů vyloučena i jednoduchá uživatelská rozhraní generovaná pomocí zásuvného modulu GATE Crowdsourcing plugin[18] [13], protože ani tato by nebylo možné efektivně využít pro komplexní hierarchické anotace. Pro experimenty byla proto zvolena standardní desktopová aplikace GATE Developer[19], v jejímž uživatelském rozhraní lze požadovanou úlohu otestovat. Předanotování pomocí nástrojů na pozadí pak bylo nastaveno stejně jako pro zbývající 2 nástroje. Práce uživatelů byla obdobně jako u RDFaCE zjednodušena na výběr atributů událostí, jejich odkazování na referenční zdroje a na následný výběr celého textu obsahujícího vybrané atributy a jeho anotování jako události.

Při přípravě experimentálního srovnání anotačních nástrojů je důležité si uvědomit, že se funkcionalita jednotlivých nástrojů výrazně liší. Současně s funkcionalitou se liší i různé aspekty anotačního procesu, což může výrazně ovlivnit jeho výsledky. Toto se projeví zejména ve chvíli, kdy srovnáváme rychlost anotování a kvalitu vytvářených anotací. Následuje popis rizik, která by mohla negativně ovlivnit spolehlivost výsledků srovnání, a strategie pro zmírnění dopadu těchto rizik na výsledky prováděných experimentů.

Lze rozlišit 3 různé úrovně potenciálních problémů:

- specificita anotačních úloh řešených v experimentech,
- nesouměřitelnost jednotlivých nástrojů,
- rozdílné znalosti a zkušenosti uživatelů.

Poloautomatické anotování textu zahrnuje celou řadu různých úloh od jednoduché identifikace několika málo specifických typů entit zmíněných v textu až po kompletní provazování potenciálně víceznačných názvů entit na znalostní bázi v pozadí, anotování komplexních hierarchických vztahů a jejich jednotlivých atributů. Doména anotovaného textu (obecné vs. např. biomedicinské či historické), jeho žánr, zdroj (např. novinové články či příspěvky z Twitteru[20]) a další vlastnosti se pro jednotlivé experimenty mohou rovněž lišit a vyžadovat různé přístupy pro předzpracování textu, což může vést k různým výsledkům automatického předanotování či přípravy nabídek anotací.

Datové sady k anotování mohou korespondovat s určitou reprezentativní podmnožinou textů, nebo mohou být zaměřené na určitý zvolený fenomén. Rozdílnost může být demonstrována různými původy datových sad využitých v předchozích anotačních výzvách, jako např. Entity Recognition and Disambiguation (ERD) Challenge 2014[21], kde byl zdůrazněn kontext omezený na to, co se běžně vyskytovalo ve vyhledávacích dotazech na web

[16] https://www.tinymce.com/
[17] https://gate.ac.uk/teamware/
[18] https://gate.ac.uk/wiki/crowdsourcing.html
[19] https://gate.ac.uk/family/developer.html
[20] https://twitter.com/
[21] http://web-ngram.research.microsoft.com/ERD2014/

ze soutěží TREC[22]. Oproti tomu SemEval-2015 Task 10[23] částečně pracuje s anotacemi pro analýzu sentimentu na mikroblozích, tedy ve zprávách na Twitteru. The Entity Discovery and Linking (EDL) na NIST TAC-KBP2015[24] se následně pokouší o extrakci zmínek o pojmenovaných entitách, jejich navázání na znalostní bázi a o následné seskupování entit, které nejsou pokryty znalostní bází. Stupeň víceznačnosti entit zmíněných v anotovaných textech, stejně jako podíl výskytů s odpovídajícím významem, tedy zjevně mohou mít zásadní dopad na rychlost a přesnost procesu anotování.

Výsledky experimentů byly ovlivněny různými aspekty uživatelského rozhraní. Některé nástroje jsou vytvořeny pro obecný proces sémantického anotování, jiné jsou částečně zaměřeny na scénář, kdy je zaplacen větší počet nezávislých nezkušených anotátorů, jako popisuje např. Bontcheva a kol. [13], což může mít za následek, že budou nevhodné pro nasazení v kolaborativním prostředí. Některé nástroje mohou být pevně vázány s konkrétním nástrojem pro extrakci informací v pozadí, zatímco jiné jsou jen volně napojeny na preferovaný nástroj, který může být pro specifický úkol snadno rozšířen či nahrazen. Další funkce nástrojů, zejména ty, které souvisejí s uživatelským rozhraním a vzory interakce, jsou popsány níže. Znalosti, aktuální stav mysli a motivace uživatelů, kteří se zúčastní experimentu, mohou rovněž ovlivnit jeho výsledky. Měřená kvalita anotací a čas anotování musejí být vždy interpretovány s ohledem na tyto faktory. Lze očekávat, že uživatelé se zkušenostmi s využitím daného nástroje budou lépe rozumět jeho uživatelskému rozhraní a budou s ním schopni dosáhnout lepších výsledků. Někteří uživatelé mohou naopak dodat zcela nesprávné výsledky, jako uvádí např. Hinze a kol. [40]. Znalost domény anotovaných textů může rovněž urychlit proces anotování, a to zejména proces zjednoznačňování entit.

Nastavení experimentu preferující kvalitu před kvantitou a naopak může vést k výrazně rozdílným časům a množstvím chyb v anotacích. Po uživatelích v provedeném experimentu však byl požadován určitý kompromis mezi časem stráveným na jedné anotaci a výslednou kvalitou (tedy mírou jistoty uživatele, že zvážil dostatečně rozsáhlý kontext pro to, aby se mohl správně rozhodnout). Zatímco uživatelé v experimentu byli ohledně preferencí v této situaci instruováni, stejný způsob lze očekávat i ve scénářích placených anotátorů, kdy je nutné využít sofistikovanou kontrolu kvality, abychom anotátorům zabránili v podvádění, jak uvádí Weng a kol. [104].

Pro minimalizaci rizik nespravedlivého porovnání bylo pro experimenty zaměřené na srovnání anotačních nástrojů zvoleno 6 uživatelů, kteří neměli žádné předchozí zkušenosti s testovanými nástroji, s řešenými úlohami, ani žádné zvláštní znalosti z domény anotovaných textů. Testeři byli studenty doktorského studia ve věku 26 – 34 let. Každý strávil 20 minut seznámením s daným nástrojem, přičemž pracoval s daty, která následně nebyla zahrnuta do testovací sady, ale obsahovala všechny typy případů, které se vyskytovaly v následném reálném testování (např. více hodnot atributu, dvě různé události v jedné větě, nabídky, mezi nimiž nebyl správný význam, apod.). Ke zvýšení spravedlivosti srovnání se pro každého uživatele lišilo i pořadí, ve kterém byly nástroje testovány. Pořadí, ve kterém uživatelé nástroje testovali, bylo pro každého uživatele unikátní, neboť pro 3 nástroje máme právě 6 možných kombinací. Instrukce požadovaly, aby se uživatelé snažili dosáhnout maximální možné přesnosti, ale čas na dokončení celého úkolu byl omezený. Současně zde byl neformální závod, kdo nejrychleji vytvoří nejpřesnější výsledky, ale nebyly zde slíbeny ani uděleny žádné zvláštní odměny (s výjimkou piva pro vítěze).

[22] http://trec.nist.gov/
[23] http://alt.qcri.org/semeval2015/task10/
[24] http://nlp.cs.rpi.edu/kbp/2015/

V průběhu experimentu měl každý uživatel 40 minut na anotování každým z nástrojů. Byly při tom měřeny tři hodnoty: Jak je uvedeno v tabulce 6.1, patří mezi ně množství nesprávně vyplněných hodnot atributů, počet chybějících atributů (entity, které byly zmíněny v textu, ale uživatel je neasocioval s anotovanou událostí) a průměrný čas na anotaci události. Nesprávně vyplněné atributy při tom zahrnovaly všechny druhy chyb, tedy nesprávnou volbu anotovaného fragmentu, prázdný nebo nesprávný typ, koreference či URL spojující jméno entity s nesprávnou položkou v referenčním zdroji apod.

Tabulka 6.1: Výsledky experimentu zaměřeného na srovnání anotačních nástrojů

tool	**Nesprávné hodnoty**	**chybějící hodnoty**	**čas na událost**
GATE	9.4 %	8.3 %	135 s
RDFaCE	8.7 %	8.8 %	193 s
4A	6.2 %	5.6 %	116 s

Celkový vysoký počet chyb (ve sloupci „Nesprávné hodnoty“) lze vysvětlit velmi striktním porovnáním s referenčními anotacemi (tzv. zlatý standard). Od uživatelů se např. očekávalo, že vypočítají a zadají interval let pro událost, jejíž popis obsahoval text: „*a woman in her 50s who travelled around ...*“ („*žena ve svých padesáti letech, která cestovala kolem ...*“). Někteří z nich však zadali hodnotu „1950“, neboť textu nevěnovali dostatečnou pozornost a „*50s*“ pochopili jako padesátá léta.

Výsledky rovněž korespondují s tím, že způsob, jakým GATE Developer prezentuje anotace atributů událostí, často vede k nejasným či nekonzistentním výsledkům. Atributy jsou totiž k události přiřazeny pouze faktem, že fragment označující událost překrývá fragment s hodnotou atributu. RDFaCE je v tomto ohledu jen o málo lepší, protože sice umožňuje přiřadit atribut k dané události, ale u překrývajících se událostí problémy přetrvávají.

Jedním z problémů, kvůli kterému atributy zůstávaly prázdné i přes to, že jejich správné hodnoty byly v textu obsaženy (sloupec „Chybějící hodnoty“), byly zájmena (koreference), u kterých bylo doporučeno jejich propojení s příslušnou odkazovanou entitou a absence tohoto propojení se počítala do chybějících hodnot.

Ve výše uvedených výsledcích se jasně projevil rozdíl mezi RDFaCE či GATE a 4A, který ukazuje, že je přínosné vizuální odlišení nabídek anotací generovaných systémem od anotací zkontrolovaných či vytvořených uživatelem. Způsob potvrzování v systému 4A tedy vede na lepší a konzistentnější anotace.

Průměrný čas na anotování události byl nejvyšší u RDFaCE, což je způsobeno především strohým uživatelským rozhraním tohoto nástroje s limitovanými možnostmi jednoduché nápravy chyb a dvoufázovým vytvářením anotace (nejprve je třeba označit fragment a vytvořit anotaci a pak teprve ji lze přes kontextovou nabídku editovat a vyplnit atributy). Další negativní dopad pak má způsob prezentace nabídky typů anotací, kde je druhá (neméně používaná) polovina lineárního seznamu skrytá pod položkou „*More...*“.

Rozdíl časů mezi 4A a GATE Developer lze vysvětlit tím, že u 4A je využito sémantické filtrování, které urychlí práci uživatele (viz níže), ale u GATE Developer byla anotační úloha zjednodušena (viz výše), což efekt tohoto vylepšení částečně kompenzovalo.

6.4.2 Optimalizace množství zobrazovaných informací

Druhá sada experimentů zkoumala dopad množství zobrazovaných informací a způsobu jejich prezentace na rychlost anotování a kvalitu vytvářených anotací. Otázkou bylo i to, zda je pro uživatele výhodou znalost alternativních anotací a míra důvěry automatického nástroje v pozadí ve správnost poskytnutých nabídek.

Tyto experimenty nebylo možné provést v nástroji, který neumožňuje nastavení daných aspektů a nabízí pouze fixní způsob zobrazení. Byly proto využity výhody flexibility anotačního nástroje A4 a jeho uživatelské rozhraní bylo nastaveno dle požadavků experimentu. Nastavení zahrnovalo zejména omezení sady zobrazovaných atributů v prvním pohledu určeném pro zjednoznačňování a v pohledu, který se zobrazí při kliknutí na tlačítko pro zobrazení detailních informací.

Jak bylo uvedeno výše, experimenty byly zaměřeny především na úlohu komplexního zjednoznačňování entit. Data byla extrahována z korpusu CommonCrawl, přičemž byly vyhledány odkazy korespondující se jmény lidí a názvy míst ve Wikipedii. Pro experimenty bylo vybráno celkem 186 úryvků textu, které byly manuálně překontrolovány. Způsob přípravy dat garantoval, že náhodné hádání správné anotace povede k chybovosti kolem 50 % (či více, pokud jsou zde více než 2 potenciální významy).

Primárně byla srovnávána 3 nastavení pohledu pro zjednoznačňování, které se lišily zobrazenými atributy, a byl sledován dopad daného nastavení na rychlost a přesnost zjednoznačňování. Uživatelé dostali instrukce, aby anotovali pouze zadanou entitu, která byla v každém úryvku textu zvýrazněna, a zvolili vždy právě jednu z poskytnutých nabídek anotací. Současně neměli přeskočit žádný z textů, takže bylo možné porovnat pouze přesnost a rychlost vytváření anotací bez ovlivnění výběrem textů, které určitý uživatel anotoval.

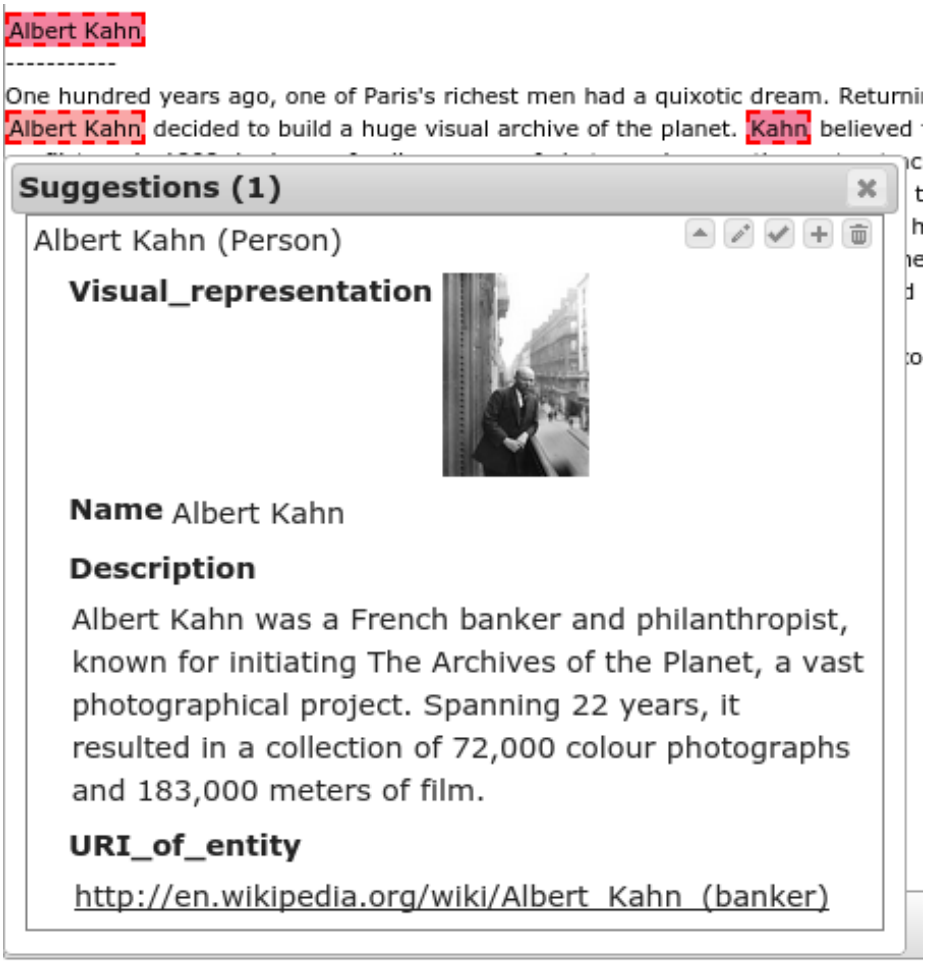
V prvním nastavení měli uživatelé pro každou nabídku s nejvyšší mírou důvěry zobrazený rozsáhlý seznam atributů a jejich hodnot. Zobrazené atributy zahrnovaly typ entity, celé jméno či název, popis korespondující s prvním odstavcem z Wikipedie nebo Freebase[25], vizuální reprezentaci (obvykle první obrázek z Wikipedie, byl-li dostupný) a URL. Tento pohled je na obrázku 6.4. Když to bylo potřeba, uživatel měl možnost využít odkaz s URL na Wikipedii či Freebase a rozhodnout se dle informací obsažených na příslušné stránce se všemi informacemi o dané entitě.

Druhé nastavení korespondovalo s omezeným zobrazením, které pro zjednoznačňování poskytuje celá řada nástrojů (např. RDFaCE). Byl tedy zobrazen pouze typ entity a URL. Uživatel se pak mohl rozhodnout podle samotné URL, nebo si zobrazit cíl daného odkazu a rozhodnout se dle obsahu celé příslušné stránky. Nutno poznamenat, že URL na Wikipedii často obsahuje slovo či frázi v závorkách usnadňující zjednoznačňování. Tyto závorky jsou pak uvedeny u entit, které se jmenují stejně jako primární (nejznámější) entita s daným jménem pokrytá Wikipedií.

Třetí nastavení umožňuje využít výhodu speciálního zjednoznačňovacího atributu, který je automaticky určen z popisů dostupných alternativ. Kombinuje zjednoznačňovací slovo či frázi z URL na Wikipedii se zvolenou částí popisu entity. Funkce, která slouží ke generování hodnoty tohoto atributu, může být snadno přizpůsobena i pro jiné referenční zdroje než jsou Wikipedie a Freebase. Společně s zjednoznačňovacím atributem byl zobrazen i typ entity a URL, takže uživatelé měli stále možnost se rozhodnout dle kompletního obsahu související stránky v referenčním zdroji. Popsané zobrazení je na obrázku 6.5.

Zatímco sekvence testovacích textů (40 pro každé nastavení) zůstala fixní, každý ze šesti testerů měl jiné pořadí tří uvedených nastavení (obdobně jako pořadí nástrojů v první sadě

[25] http://wiki.freebase.com/wiki/Main_Page



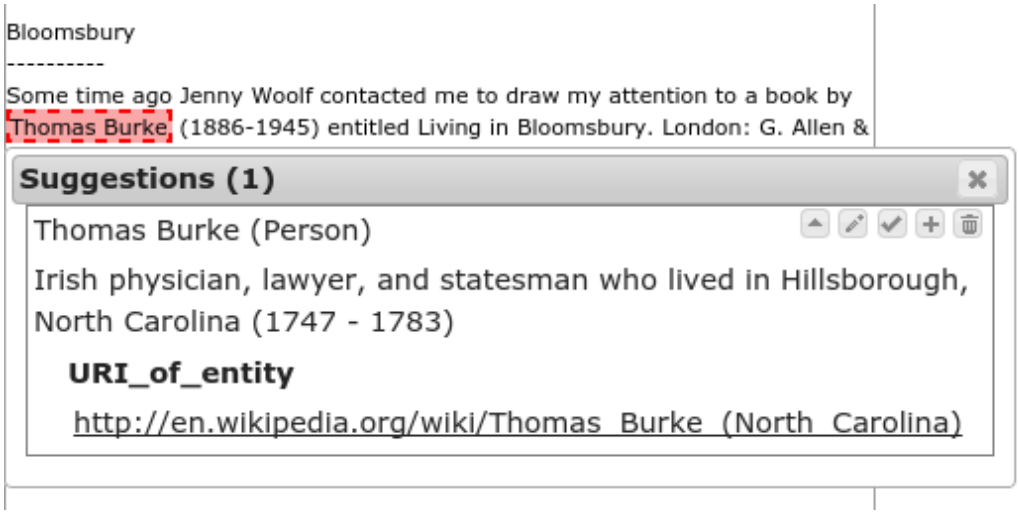
Obrázek 6.4: Příklad zobrazení detailních informací u nabídky anotace

experimentů). Každý měl 30 minut na každé nastavení. V tabulce 6.2 je srovnání časů, chybovosti a toho, jak moc kliknutí na URL a prohlédnutí příslušné stránky s dalšími informacemi v referenčním zdroji uživatelé potřebovali.

Tabulka 6.2: Výsledky experimentu srovnávající tři různá nastavení zobrazení pro zjednoznačňování

nastavení	**průměrný čas**	**množství chyb**	**kliknutí na URL**
detailní informace	33.92 s	6.2 %	1.3 %
pouze typ a URL	37.26 s	27.9 %	41.7 %
zjednoznačňovací atribut	32.98 s	2.1 %	1.5 %

I když mezi jednotlivými testery byly určité rozdíly, celkové výsledky (nastavení s nejlepším a nejhorším průměrným časem a chybovostí) byly pro všechny testery stejné. Počet případů, ve kterých uživatel potřeboval informace z kompletní stránky v referenčním zdroji,



Obrázek 6.5: Příklad zobrazení zjednoznačňovacího atributu u nabídky anotace

byl vždy nejvyšší při druhém nastavení (pouze typ a URL), ale uživatelé se nezanedbatelně lišili v tom, jak moc věřili, že pouhý obsah URL je postačující ke správnému rozhodnutí. To pak vedlo ke zvýšenému počtu chyb pro dané nastavení.

Relativně vysoký počet chyb je způsoben i komplexitou zjednoznačňovací úlohy, na což poukázala i zpětná vazba od uživatelů získaná z dotazníků vyplňovaných po dokončení všech tří sezení. Uživatelé se sice snažili udělat co nejméně chyb, ale sezení delší než 20 minut se jim zdála velmi náročná a tvrdili, že by pracovali rychleji, pokud by se mohli více zaměřit na rychlost než na kvalitu. Nutno zmínit, že ve chvíli vyplňování dotazníku uživatelé ještě nevěděli, kolika chyb se při anotování dopustili. Po konfrontaci s množstvím chyb ve výsledcích svojí práce pak uvedli, že se pokoušeli o kompromis mezi rychlostí a kvalitou a navrhovali zavedení kontextově závislé funkcionality, která by jim pomohla v konkrétních případech, se kterými se setkali (např. obrázek v případě rozhodování mezi názvem lodi a jménem člověka, datum úmrtí v případě, kdy dva nabízení lidé žili v různých stoletích, apod.).

Fakt, že uživatelé podcenili komplexitu zadané zjednoznačňování úlohy, také pravděpodobně vysvětluje překvapivě vysoký počet chyb v případě okamžitého prezentování rozsáhlých informací o nabízených entitách (první popsané nastavení). Příliš mnoho informací, ve kterých nejsou zdůrazněny klíčové rozdíly mezi jednotlivými alternativami, tedy pravděpodobně vede k méně soustředěné práci. V dalším výzkumu se se pokusím zjistit, zda se toto změní, budou-li uživatelé zkušenější. Nicméně průměrný čas na rozhodnutí a související nízký počet kliknutí na odkaz do Wikipedie korespondují s faktem, že po prohlédnutí uvedených textových popisů a obrázků uživatelé nabyli dojmu, že mají dostatek informací pro správné rozhodnutí.

Pro nastavení zobrazující pouze typ a URL byly mezi jednotlivými uživateli největší rozdíly. Někteří z nich navštívili více než 2/3 ze všech odkazů a četli si informace ze stránek na Wikipedii, zatímco jiní se rozhodovali mnohem rychleji, ale také dělali velké množství chyb. Přestože druhý efekt by bylo možné částečně eliminovat zvýšenou penalizací za chyby, toto nastavení se pro danou úlohu ukázalo jako jasně nejhorší. Nástroje, které pro zjednoznačňování nabízejí pouze typ a URL, by se tedy zavedením detailnějších pohledů mohly výrazně zlepšit.

Jako jasně nejlepší nastavení v této části experimentů se ukázalo využití zjednoznačňovacího atributu s možností kliknutí na URL pro zobrazení detailů. Uživatelé s tímto nastavením udělali méně chyb než s ostatními a dosáhli rovněž nejnižšího průměrného času na anotaci. Wikipedii potřebovali navštívit jen velmi zřídka. Pět ze šesti uživatelů pak v dotazníku uvedlo, že pro ně toto nastavení bylo nejpohodlnější.

6.4.3 Zobrazení alternativních nabídek anotací

Tento experiment byl zaměřen na způsob prezentace alternativních nabídek anotací uživateli. Na rozdíl od zjednodušené situace připravené pro předchozí experiment byly pro tento experiment připraveny realističtější podmínky, kdy uvedené jméno nemusí korespondovat se žádnou z entit pokrytých využitými referenčními zdroji (v tomto případě znalostní bází připravenou z Wikipedie, DBPedie a Freebase). Žádná z poskytnutých nabídek anotací tedy nemusela být správná. Byla při tom využita mnohoznačná jména s pěti či více alternativními významy ve Wikipedii. Tím byla rovněž vyloučena jednoduchá strategie výběru využitelná v předchozím experimentu. Uživatelé si tedy nemohli zjednodušit práci jednoduchým vyloučením nesprávné alternativy a bezmyšlenkovitým potvrzením druhé zbývající možnosti.

Byly připraveny dvě datové sady po 15 úryvcích textu zmiňujících vybrané entity, kde 3 z nich (20 %) v každé sadě nebyly pokryty referenčními zdroji a tedy nekorespondovaly s žádnou z nabídnutých alternativ. První nabídka byla správná v 9 z 15 případů v každé sadě (60 %), což korespondovalo s empirickým měřením přesnosti nástroje pro automatické generování nabídek anotací v pozadí systému pro dané texty. Na úroveň míry důvěry ve správnost nabídek anotací při tom nebyl aplikován žádný práh, takže nejlepší nabídka byla poskytnuta i v případě, že žádná z nabídek nebyla správná.

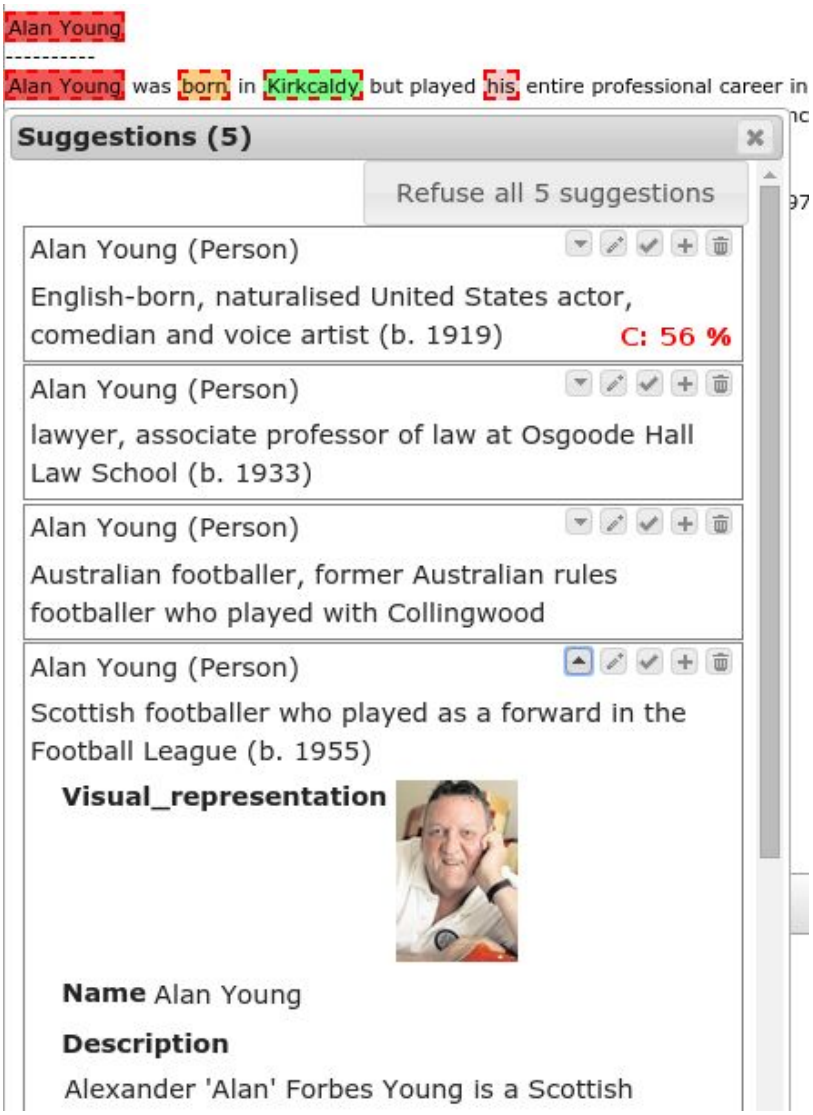
První testované nastavení korespondovalo s obecně využívanou prezentací nejlepší (nejpravděpodobnější) alternativy anotace se zobrazením zjednoznačňovacího atributu. Uživatelé si mohli rozbalit i další atributy s detailními informacemi, případně navštívit odpovídající stránku na Wikipedii. Při odmítnutí této nabídky se pak zobrazily alternativy.

Druhé nastavení ihned zobrazilo 5 nejpravděpodobnějších nabídek a míru důvěry ve správnost nejlepší z nich (ostatní byly z technických důvodů pouze seřazeny podle globálního hodnocení odrážejícího především návštěvnost odpovídající stránky na Wikipedii, neboť výpočet reálné kontextově závislé hodnoty pro každou alternativu by byl časově příliš náročný). Zobrazení s tímto nastavením je na obrázku 6.6.

Tři uživatelé začali s prvním uvedeným nastavením, 3 s druhým. V případech, kdy nebyla žádná z nabídek správná, se za správné rozhodnutí považovalo odmítnutí všech nabídek, což bylo možné udělat po jednotlivých nabídkách nebo pro všechny současně jedním kliknutím. Výsledky jsou shrnuty v tabulce 6.3.

Tabulka 6.3: Srovnání zobrazení jedné nabídky anotace a všech alternativ

nastavení	**průměrný čas**	**množství chyb**	**kliknutí na URL**
jedna nabídka	41.2 s	14.4 %	3.3 %
všechny alternativy	42.9 s	12.2 %	2.2 %



Obrázek 6.6: Zobrazení alternativ v pohledu pro zjednoznačňování

Průměrný čas na výběr z nabídnutých možností oproti předchozímu experimentu značně narostl, a to na 40 sekund. Odpovídá to značně zvýšené složitosti úlohy, neboť se uživatel rozhoduje z více alternativ. Jak je patrné z posledního řádku tabulky, na posouzení všech alternativ je potřeba o něco více času, ale jejich okamžité zobrazení vede ke zvýšení správnosti.

Přesto že výsledky nejsou zcela přesvědčivé a obecně závisí na kvalitě procesu generování nabídek anotací v pozadí a na četnosti případů, kdy zmíněná entita není pokryta referenčními zdroji, zobrazení všech alternativ se ukazuje jako vhodnější, chceme-li v daném čase preferovat kvalitu vytvářených anotací před kvantitou. Na druhou stranu se však z uživatelských odpovědí v dotaznících neukázalo, že by jim znalost míry důvěry automatického nástroje ve správnost dané nabídky anotace pomohla v pečlivějším posuzování méně pravděpodobných nabídek. Tuto skutečnost bude v budoucnosti potřeba blíže prozkoumat.

6.4.4 Sémantické filtrování

Poslední sada experimentů z této série srovnávala dva způsoby anotování komplexních událostí a jejich atributů. Jejím cílem bylo měření, jakého zrychlení lze dosáhnout s využitím pokročilého sémantického filtrování. Uživatelé dostali instrukce, aby anotovali pouze části uvedených odstavců textu popisující zahraniční cesty zmíněných osob. Data byla zvolena tak, aby byla pokud možno co nejrealističtější – obsahovala tedy i věty korespondující s jinými druhy událostí a zmínky o entitách, které v příslušných událostech nehrají žádnou roli.

Byly připraveny dvě sady textů, z nichž každá obsahovala 6 událostí správného typu. V první sadě textů měli uživatelé identifikovat atributy událostí manuálně, ve druhé sadě bylo využito automatické předanotování potenciálních hodnot atributů všech potřebných typů (nabídky anotací) a pokročilé sémantické filtrování systému 4A, které zvýraznilo potenciální kandidáty na hodnoty právě vyplňovaného atributu. Druhé nastavení tak zjednodušilo práci uživatele na kompletaci nabídnutých anotací entit, doplňování chybějících a sestavování jednotlivých částí do celých událostí.

Výsledky pro jednotlivá nastavení jsou porovnány v tabulce 6.4. Manuální proces bez předanotování je zdlouhavý a uživatelé tak strávili více než 5 minut anotováním jedné události. Výsledky také obsahovaly mnoho chyb (vč. nadbytečných nesprávných hodnot). Oproti tomu sémantické filtrování zapnuté ve druhé části experimentu vedlo k dosažení kvalitních výsledků anotování v relativně krátkém čase. Na samotný koncept pokročilého sémantického filtrování se blíže zaměřím ve druhé sérii experimentů.

Tabulka 6.4: Manuální vyhledávání hodnot vs. nabízení anotací se sémantickým filtrováním

vyplněné atributy	**nesprávné hodnoty**	**chybějící hodnoty**	**čas na událost**
manuálně	11.7 %	6.6 %	303.5 s
s nabízením	4.5 %	3.4 %	109.3 s

6.5 Druhá série pokročilých experimentů

V této sérii experimentů byly rozšířeny předchozí experimenty publikované v článku *Interaction Patterns in Computer-Assisted Semantic Annotation of Text – An Empirical Evaluation* [27] o detailnější analýzu výhod sémantického filtrování v nástroji 4A. Rovněž je zde rozšířena studie zabývající se anotováním mnohoznačných jmen zaměřená na zkoumání, jakého zrychlení lze dosáhnout pomocí různých módů prezentace pokročilých zjednoznačňovacích kontextů.

Oproti předchozí sérii experimentů byl zvýšen počet anotátorů a pro získání nových pohledů na vzory interakce byly připraveny nové anotační úlohy. Pro zkoumání výhod sémantického filtrování v nástroji 4A jsou zde dva související úkoly, přičemž prvním je určit autorství uměleckého díla a druhým najít vztahy ovlivnění mezi těmito díly či jejich autory.

Druhá sada experimentů z předchozí série zaměřená na hledání optimálního množství zobrazovaných informací pro spolehlivé zjednoznačňování entit ukázala, že běžná praxe anotačních nástrojů, při které musí uživatel zjednoznačňovat pouze s využitím typu a URL vede k nízké kvalitě výsledků. V této sérii experimentů byly získány další výsledky z práce

na úloze zjednoznačňování mnohoznačných jmen, se kterými jsou demonstrovány výhody využití zjednoznačňovacího atributu připraveného dle dané úlohy. Současně je v tomto experimentu zkoumáno, zda je výhodné ihned zobrazit i další doplňující atributy, nebo jejich zobrazení ponechat na uživateli, který se rozhodne, zda tyto informace potřebuje, nebo mu pro rozhodnutí postačuje pouze zjednoznačňovací atribut.

6.5.1 Přínos sémantického filtrování

Výsledky předchozích experimentů ukázaly, že i přes to, že automatický proces anotování nemůže identifikovat komplexní vztahy, je výhodné jej využít pro předanotování entit a základních vztahů. Uživatel se pak může soustředit na vysokoúrovňové anotační úlohy jako současná validace odkazů na znalostní bázi a spojování jednotlivých komponent. Z předchozích výsledků však nebylo zcela jasné, v jakém rozsahu zvýraznění preferovaných typů anotací v textu využitelných jako hodnoty atributů napomáhá k vyšší kvalitě a rychlosti anotování vztahů a zda zvyšuje komfort uživatelů. Následujícím experiment se tedy na rozdíl od předchozího experimentu spojujícího roli nabízení anotací se sémantickým filtrováním zaměřuje výhradně na sémantické filtrování.

Pro tento experiment byla připravena sada 20 úryvků z dokumentů o vizuálních uměleckých dílech (obrazech a sochách) a vztazích vzájemného ovlivnění umělců, včetně ovlivnění mezi těmito díly. Každý z těchto textů zmiňoval několik uměleckých děl, jejich autorů a okolnosti jejich vytvoření (místa, data, vyobrazené osoby apod.). Současně zde byly odkazy na jiná díla, jimiž byli příslušní umělci inspirováni či jinak ovlivněni. Ukázka části jednoho z využitých textů (pocházejícího z online galerie allart.biz[26]) je v následujícím odstavci:

> *Le déjeuner sur l'herbe is a large oil on canvas painting by Édouard Manet created in 1862 and 1863. Manet's composition reveals his study of the old masters, as the disposition of the main figures is derived from Marcantonio Raimondi's engraving of the Judgement of Paris (c. 1515).*

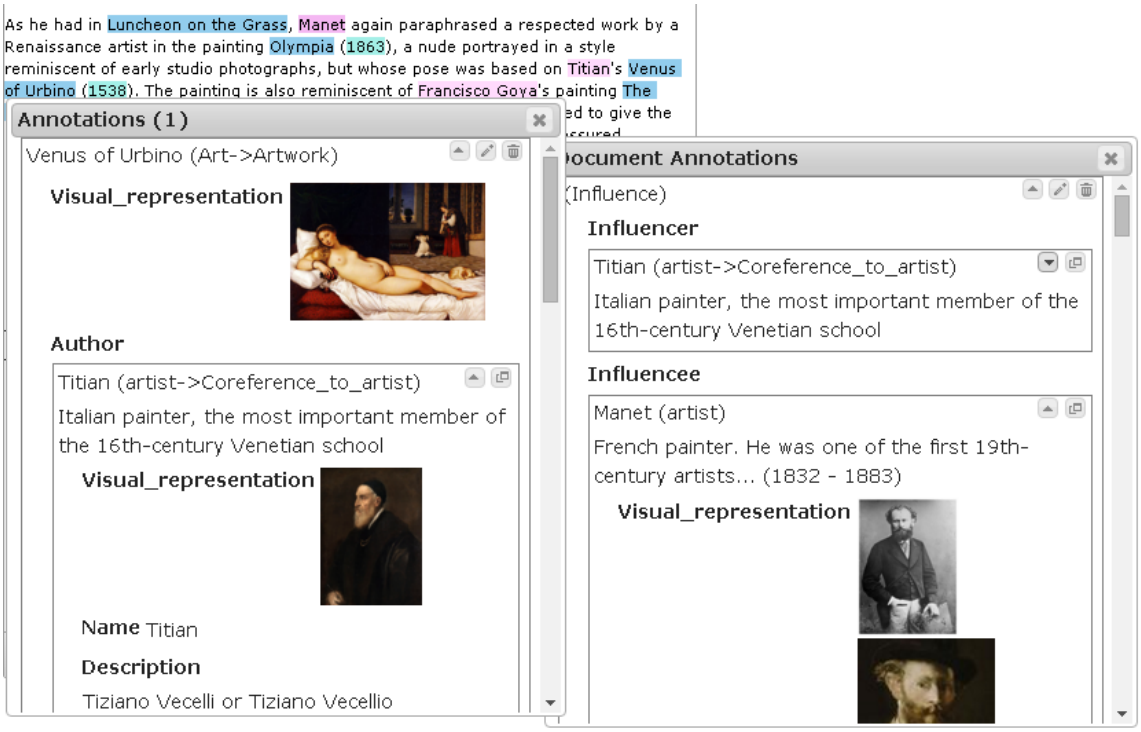
Skupina 14 uživatelů v textech nejprve identifikovala umělecká díla a jejich atributy (převážně autory a data vytvoření). Následně anotovali vztahy ovlivnění mezi těmito díly a jejich autory. Výsledek je demonstrován na obrázku 6.7.

Byly připraveny 2 různé konfigurace systému, z nichž s každou uživatelé anotovali 10 textů. Při první konfiguraci byly zvýrazněny anotace příslušných typů korespondující s možnými hodnotami právě vyplňovaného atributu. Při druhé konfiguraci bylo sémantické filtrování vypnuté a uživatelé tak viděli, že mají pro vyplnění hodnoty atributu na výběr ze všech anotací v daném textu.

Aby bylo vyloučeno ovlivnění pořadím anotací, uživatelé texty dostali v náhodném pořadí. Metoda výběru však garantovala, že každý text byl anotován 7 uživateli se zapnutým sémantickým filtrováním a 7 s vypnutým. Polovina uživatelů při tom anotovala nejprve s vypnutým a pak se zapnutým sémantickým filtrováním a u poloviny to bylo naopak.

V tabulce 6.5 je uvedeno srovnání výsledků získaných s jednotlivými variantami nastavení. Je zřejmé, že zapnuté sémantické filtrování nástroje 4A vedlo k vyšší kvalitě výsledků. Relativní snížení obou typů chyb překročilo 25 %. Proces anotování byl také o 15 % rychlejší. Dotazníky vyplňované po skončení experimentu pak ukázaly, že 11 ze 14 uživatelů je toho názoru, že funkce sémantického filtrování výrazně zlepšuje jejich komfort při anotování. Zbývající 3 se shodli na tom, že jim tato funkcionalita pomohla „středně".

[26] http://allart.biz/photos/image/Edouard_Manet_1_The_Luncheon_on_the_Grass.html



Obrázek 6.7: Atributy uměleckých děl a vztahy ovlivnění

Tabulka 6.5: Přínos sémantického filtrování

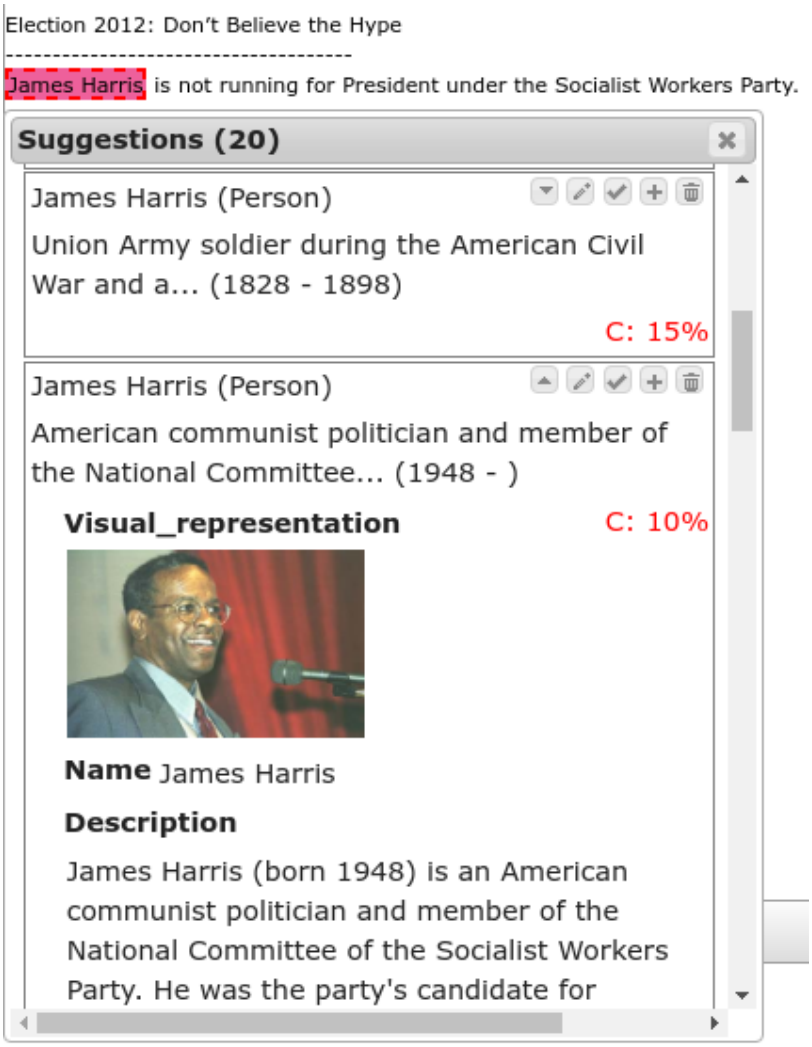
sémantické filtrování	**nesprávné hodnoty**	**chybějící hodnoty**	**čas na vztah**
vypnuté	6.9 %	5.7 %	41.4 s
zapnuté	5.1 %	4.2 %	35.1 s

6.5.2 Alternativní nabídky anotací

V předchozí sérii experimentů byla srovnána dvě nastavení uživatelského rozhraní pro zjednoznačňování – v jednom byly přímo uvedeny známé entity sdílející dané jméno zmíněné v textu a ve druhém byla zobrazena pouze nejpravděpodobnější entita, přičemž měl uživatel možnost zobrazení dalších alternativních entit kliknutím na příslušné tlačítko. Ukázalo se, že první nastavení vede k vyšší kvalitě vytváření anotací. Toto nás motivovalo k rozšíření daného experimentu a detailnějšímu vyhodnocení situace, kdy budou uživateli zobrazeny všechny alternativy.

Počet entit ve zvolených textech, které sdílely dané jméno, byl vysoký (mezi 10 a 30). Toto koresponduje se situací, kdy má automatický zjednoznačňovací nástroj pouze značně omezené informace pro rozhodnutí, neboť míry důvěry ve správnost jednotlivých alternativ jsou nízké. Pozice správné volby (nabídky anotace) se v připravených datech lišila – pro 16 z 20 textů byla na seznamu uvedena, zatímco ve 4 zbývajících případech (20 %) s entitou zmíněnou v textu nekorespondovala žádná z nabídnutých alternativ.

Byly srovnány dvě možnosti prezentace seznamu alternativ uživateli. Ve stručnější variantě byly vypsané pouze zjednoznačňovací kontexty a umožněno rozbalení kompletního seznamu atributů entity ze znalostní báze jedním kliknutím (včetně popisu, vizuální reprezentace apod.). Ukázka tohoto nastavení po rozbalení atributů jedné z entit je na obrázku 6.8. Ve druhé variantě nastavení pak byly všechny atributy ihned rozbalené a uživatel měl možnost je pro každou alternativní nabídku jedním kliknutím sbalit. Při druhém nastavení sice uživatel musí procházet výrazně delší seznam, ale v popisech jednotlivých entit může rychle vyhledávat klíčová slova, která se vyskytují v textu.



Obrázek 6.8: Příklad rozbalení atributů pro jednu z alternativních nabídek anotací

Každý ze 20 zjednoznačňovacích úkolů byl řešen 7 anotátory se stručným pohledem a 7 s rozbaleným pohledem. Polovina uživatelů začala se stručným pohledem a polovina s rozbaleným, přičemž po anotování poloviny textů přešli na druhý pohled.

Aby bylo možné korektně pracovat s různým počtem alternativ a různými pozicemi správné alternativy v seznamu nabídek pro jednotlivé zmínky o entitách v textu, naměřený celkový čas byl vždy vydělen pořadím správné alternativy. To koresponduje s nejčastěji pozorovaným scénářem (86 % případů), kdy se uživatel rozhodne ihned po přečtení po-

pisu entity, která odpovídá anotovanému textu. Pro 4 případy, kdy zmíněná entita nebyla pokryta znalostní bází, byl uvažován celkový počet zobrazených alternativ.

Výsledky tohoto experimentu jsou shrnuty v tabulce 6.6. Stručný pohled uživatelům umožnil rychleji procházet alternativy a zvolit jednu z nich. Toto bylo potvrzeno rovněž dotazníky, které uživatelé vyplňovali po experimentu. Všichni kromě jediného uživatele zde jako preferovaný označili stručný pohled.

Tabulka 6.6: Zjednoznačňování se stručným a s rozbaleným pohledem na alternativní nabídky anotací

nastavení	**čas na alternativu**	**množství chyb**	**kliknutí na sbalení/rozbalení**
sbalené	11.2 s	12.1 %	1.1 %
rozbalené	15.1 s	11.4 %	5.3 %

Relativně vysoký počet chyb koresponduje s komplexitou zjednoznačňovací úlohy. Jak se ukázalo i z odpovědí na dotazníky, správnost by bylo možné zvýšit tak, že by uživatelé neskončili u alternativy, která se jim zdá dostatečně správná, ale vždy by prošli celý seznam alternativ až do konce a pak teprve mohli zvolit tu správnou. Dle sesbíraných dat by to však mohlo vést k více než dvojnásobnému zpomalení procesu zjednoznačňování. Takové potenciální snížení rychlosti anotování se zdá příliš vysoké na to, aby jej bylo možné akceptovat.

Poslední sloupec v tabulce 6.6 charakterizuje vzor interakce, který bylo možné sledovat při práci uživatelů se seznamem alternativ. Ve stručném pohledu bylo rozbaleno pouze 1,1 % kandidátních alternativních entit, aby si uživatelé prohlédli jejich popis a další detailní informace. Stručný zjednoznačňovací atribut byl tedy většinou uživatelů shledán dostatečným. Naopak při práci s rozbaleným pohledem uživatelé klikli na sbalovací tlačítko v 5,3 % případů. Analýza dotazníků později ukázala, že sbalovací tlačítko bylo v tomto případě nejčastěji využíváno pro označení (vyloučení) nesprávných voleb. Tlačítko pro definitivní odmítnutí některé z nabídek a odstranění ze seznamu však až na 1 výjimku uživatelé vůbec nevyužili, protože až do výběru správné entity chtěli mít možnost se k libovolné nabídce vrátit. V budoucnosti budu tuto interakci zkoumat detailněji.

6.6 Rozšiřování ontologie

V této podkapitole popíšu miniexperiment zaměřený na srovnání různých vzorů interakce při kolaborativním zjemňování a doplňování anotačních ontologií pro aspektově orientovanou analýzu sentimentu popsanou ve výše uvedených příkladech použití. Připravené prostředí experimentu koresponduje s reálnou situací, kdy klient spolupracuje se znalostním inženýrem na přípravě jádra ontologie pro extrakci názorů z textu s využitím konceptů a vlastností specifických pro určitou doménu analýzy reputace obchodní značky.

Dva zkušení tvůrci ontologií v roli „znalostních inženýrů“ se zkušenostmi s využitými nástroji v rámci experimentu spolupracovali se dvěma dalšími uživateli, kteří byli v roli „informovaných klientů“. Byly připraveny 4 sady textů korespondující s recenzemi restaurací, aerolinek, hotelů a počítačových her. V každé kategorii bylo 15 recenzí získaných ze 2 – 3 různých serverů. Každá recenze byla v rozsahu 3 – 12 vět.

Pro rozšiřování ontologie byly využity dva různé nástroje – 4A a WebProtégé[27]. Vždy, když uživatelé vytvořili nový koncept v nástroji 4A, ihned pomocí něj anotovali zmínky o jednotlivých aspektech, které se vyskytovaly ve zpracovávaných textech recenzí. Mohli využít pokročilé funkcionality tohoto nástroje umožňující jednoduché pojmenování jednotlivých konceptů i jejich vlastností. Při využití WebProtégé však texty sloužily pouze jako doplňkový zdroj znalostí.

Dané nastavení experimentu poskytovalo základ pro 4 běhy kolaborativní přípravy ontologie. Dvě dvojice (první znalostní inženýr s 1. informovaným klientem v prvních 2 bězích a s druhým ve druhých 2 bězích) vždy dostaly stejnou sadu textů ale pracovaly buď s 4A (první dvojice v lichých bězích a druhá v sudých), nebo s WebProtégé (první dvojice v sudých bězích a druhá v lichých).

Dokumenty s recenzemi byly nejprve dány znalostním inženýrům (jako hypotetické výsledky primárního stahování z webu), kteří si je přečetli, aby se seznámili s jednotlivými aspekty názorů zmíněných v reálných datech. Měřený úkol se pak v každém běhu skládal ze 2 kroků. V prvním kroku byla zahrnuta individuální práce obou účastníků. Druhý krok zahrnoval společnou práci celé dvojice. V prvním kroku znalostní inženýr připravil svůj první výchozí návrh ontologie s využitím přiřazených nástrojů, zatímco informovaný uživatel dostal za úkol pokusit se vypsat potenciální aspekty v dané oblasti v prostém textu. Ve druhém kroku každá dvojice porovnala výchozí návrh ontologie s neformálním seznamem aspektů, identifikovala konfliktní pohledy a výchozí ontologii modifikovala tak, aby vyjadřovala odsouhlasené společné porozumění dané oblasti.

V tabulce 6.7 jsou shrnuty průměrné časy k dokončení popsaných dvou kroků s využitím obou nástrojů. I když se čtyři výsledné ontologie výrazně liší v počtu aspektů, na kterých se uživatelé dohodli, aby mohly být zahrnuty ve výsledné struktuře, obecný vzor ukázaný v tabulce zůstal přes jednotlivé běhy stejný. Přidaná práce spojená s anotací příkladů v textu prokládaná doplňováním struktury znalostí v nástroji 4A vedla k mírnému zpomalení individuální přípravy ontologie znalostními inženýry. Dostupnost reálných dat anotovaných navrženou ontologií však výrazně zrychlila řešení identifikovaných konfliktů a redukovala potřebu vysvětlovat a diskutovat kategorizaci každého jednotlivého aspektu.

Tabulka 6.7: Výsledky experimentu na srovnání nástrojů pro rozšiřování ontologií

nástroj	**průměrný čas**	
	individuálního kroku	**kolaborativního kroku**
WebProtégé	18.6 min.	15.3 min.
4A	19.1 min.	12.6 min.

Dotazníky, které účastníci experimentu vyplňovali po všech čtyřech bězích, potvrdily, že konzultování ontologie nad reálnými anotovanými daty vnímali jako zásadní vylepšení práce. Pro dosažení shody na struktuře aspektů názorů pak uživatelé jako jednodušší úlohu shledali práci s recenzemi restaurací a hotelů a jako obtížnější práci s recenzemi aerolinek a počítačových her.

[27] http://webprotege.stanford.edu/

Kapitola 7

Dosažené výsledky a jejich analýza

V této kapitole shrnu nejdůležitější dosažené výsledky této práce a analyzuji jejich potenciální dopad na aktuální stav poznání a budoucí vývoj v dané oblasti.

Mezi hlavní technické výsledky patří navržený protokol pro přenos anotací a realizovaný anotační systém, který přináší nejen řadu nových konceptů a inovací oproti existujícím systémům, ale poskytuje také platformu pro testování různých aspektů uživatelských rozhraní anotačních nástrojů.

Následné experimenty pak prokázaly správnost konceptů optimalizujících množství zobrazovaných informací, čímž lze dosáhnout vyšší rychlosti anotování i kvality vytvářených anotací.

Zvláštní důraz byl kladen na prozkoumání nového konceptu alternativních nabídek anotací a konceptu sémantického filtrování. V závěru jsem pak zkoumal výhody realizovaného anotačního nástroje pro kolaborativní vytváření ontologie.

7.1 Protokol pro přenos anotací a formát anotace

Dosud neexistoval žádný otevřený protokol pro přenos anotací, který by umožňoval komplexnější komunikaci než pouhé uložení, aktualizaci či smazání samotné anotace. API, navržené Lee Feigenbaumem ve firmě IBM [30], má oproti mnou navrženému protokolu značnou nevýhodu v tom, že je potřeba knihovna, která na straně klienta zajistí komunikaci se serverem skrze dané API. Při definování jasného protokolu lze přitom jednoduše implementovat klienta, který nebude závislý na konkrétní knihovně, ani na platformách, pro které byla knihovna implementována.

Použitelnost a univerzálnost navrženého protokolu pro přenos anotací byla prokázána při nasazení v evropském projektu Decipher, popsaném v kapitole 6. Změna ze mnou navrženého formátu na formát Open Annotation přitom nevyžadovala žádný zásah do základu protokolu – změny ve verzi 2.0 oproti 1.0 byly motivovány pouze nedostatky ve verzi 1.0 zjištěnými při nasazení v projektu Decipher a požadavky na nově přidávanou funkcionalitu. Tím byla potvrzena možnost univerzálního využití protokolu pro různé formáty anotací.

Navržený formát anotace 4A se při testování ukázal jako vhodná alternativa k formátu Open Annotation umožňující jednodušší a efektivnější komunikaci mezi klientem a serverem i jednodušší zpracování dat. I přes to, že formát Open Annotation ve své nejnovější verzi přejmenované na Web Annotation [79] poskytuje více možností než mnou navržený formát a jednodušší zpracování výsledných anotací různými nástroji, protože se jedná o standardní RDF, stále nenabízí vhodnou specifikaci pro popis šablon pro atributy jednotlivých séman-

tických typů anotací. Tím je prokázáno, že je zde stále prostor pro vylepšení i pro vývoj nových formátů anotací umožňujících jednodušší zpracování, které by bylo výhodou pro mobilní platformy. Do budoucna by na základě formátu 4A mohlo být vytvořeno rozšíření formátu Web Annotation, které by rozšířilo jeho možnosti pro využití pro komplexní anotační úlohy.

Do budoucna mám v plánu navržený protokol a rozšíření formátu Web Annotation nabídnout kolegům ze skupiny *W3C Web Annotation Working Group*, aby mohl být masivněji rozšířen a využíván. Kdyby v budoucnu došlo ke standardizaci protokolu pro přenos anotací, umožnilo by to širokou interoperabilitu anotačních nástrojů s modelem klient – server, a tím i rychlejší vývoj v oblasti sémantického anotování textu.

7.2 Vytvořený anotační systém

Na vytvořený anotační systém se lze dívat z více úhlů pohledu. Z praktického pohledu jej lze chápat jako univerzální anotační nástroj, který lze snadno nasadit do systémů pro správu obsahu a využít pro hierarchickou anotaci komplexních událostí. Použitelnost k tomuto účelu byla prokázána v projektu Decipher.

Hlavním cílem nástroje však bylo připravit platformu umožňující výzkum v oblasti anotování. Celá řada konceptů byla v minulosti otestována na nástrojích, které byly navrženy a implementovány primárně pro otestování daného konceptu. Např. Zheng navrhl Bundle Editor [111] pro ověření konceptu strukturování anotací a Yee a jeho tým implementovali CritLink [109], aby získali nástroj s kombinací všech funkcí, které identifikovali jako základ pro úspěšný editor anotací. Znovupoužitelnost takových nástrojů pro ověření dalších konceptů však často byla malá a celá řada nástrojů posloužila pouze k jednorázovému výzkumu, po kterém byla jejich podpora ukončena. Některé nástroje pak skončily v počátcích pokusu o komerční provoz (např. SharedCopy [87]). Implementace základu anotačního nástroje je však náročná, přičemž se složitostí struktur anotací a jejich vizualizace značně narůstá. I pro ověření poměrně jednoduchých konceptů je tak vynakládáno značné úsilí na vývoj nástrojů, ve kterých tyto koncepty bude možné ověřit.

Nástroj 4A byl proto vyvinut tak, aby byl v maximální možné míře modulární, konfigurovatelný a opakovaně využitelný pro ověřování nových konceptů. Např. při přidání alternativních nabídek anotací tak postačovalo přidat jeden modul serveru a mírně rozšířit komunikační protokol, modul pro komunikaci a klienta, aniž by byly potřebné výraznější zásahy do základní funkcionality systému. Vyřazení alternativ pak lze provést pouhým vypnutím či vyřazením příslušného modulu na serveru. Pro níže uvedenou optimalizaci množství zobrazovaných informací dokonce nebylo potřeba provádět žádné změny implementace, neboť vše potřebné bylo možné řešit úpravami konfigurace. Vzhledem k tomu, že je systém navržen i pro práci s komplexními hierarchickými anotacemi, nabízí široké spektrum možností anotování, a poskytuje tak vynikající základ pro rozšiřování a testování dalších konceptů. Výsledné řešení je licencováno svobodnou licencí (Apache[1]) a může být dále využíváno výzkumníky, kteří budou objevovat nové koncepty a budou potřebovat systém, ve kterém by je mohli otestovat a vyhodnotit.

Využitelnost systému 4A pro ověřování nových anotačních konceptů byla demonstrována provedenými experimenty, jejichž vyhodnocení následuje níže. Vytvořený systém mi tak umožnil dosažení dalších cílů mojí práce.

[1] http://www.apache.org/licenses/LICENSE-2.0.txt

7.3 Vyhodnocení srovnání anotačních nástrojů

Dle kritérií uvedených v kapitole 4 jsem provedl srovnání anotačních nástrojů, jehož výsledky jsou v přílohách A a B. Na toto teoreticky zaměřené srovnání navazoval praktický experiment srovnávající vybrané pokročilé anotační nástroje, popsaný v kapitole 6.

Z tabulek v příloze A je patrné, že existuje velké množství nástrojů. Některé nástroje již nejsou dostupné a některé nové do tabulky dosud nebyly zařazeny. Je však překvapivé, že novější nástroje často nepřinášejí rozšířenější funkcionalitu, ale pouze jiné způsoby, jak realizovat základní funkce. Jen omezené množství nástrojů nabízí možnost strukturování anotací a pouze málo z nich umožňuje strukturování jiným způsobem než pouze ve využité ontologii, kterou drtivá většina nástrojů neumožňuje přímo modifikovat.

Ze zastoupení webových aplikací a doplňků do webových prohlížečů je patrný trend vedoucí k anotování v prostředí webu a aplikací, které uživatelé běžně využívají. To potvrzuje správnost myšlenky, na které je postaven nástroj 4A.

Poněkud neutěšenou situaci lze sledovat v oblasti využitých formátů anotací a protokolů využívaných pro jejich přenos. Standardizované formáty jako Open Annotation, Annotea či RDFa (který však vyžaduje přenos anotovaného textu s anotacemi) zatím využívá jen malé množství nástrojů. Další nástroje pak využívají standardní obecné formáty jako XML či JSON, ale struktura dat je zde určena autorem systému, a nejsou tak vzájemně interoperabilní. Téměř pětina nástrojů pak pracuje s uzavřeným proprietárním formátem, což jakoukoliv otevřenou interoperabilitu zcela vylučuje. Za dobu svého výzkumu však pozoruji jasný trend vedoucí k širšímu využití formátu Web Annotation, který je patrný i z vícenásobného zastoupení jeho předchůdce Open Annotation v tabulce v příloze A.

Pro přenos anotací je nejvíce využíván protokol HTTP. Tím jsou pak přenášeny přímo anotace (uložení, aktualizace či smazání – často se využívá jednoduché REST API[2]), nebo nějaký jednoduchý proprietární protokol na nižší úrovni. Definovaný protokol na vyšší úrovni pak využívá jen zanedbatelné množství nástrojů, které definují několik typů zpráv ve formátu JSON[3]. Interoperabilita nástrojů je díky nejednotnosti protokolu značně komplikovaná. Zde se nabízí prostor pro mnou navržený protokol 4A.

Srovnání pokročilých nástrojů v tabulce v příloze B ukazuje, že i nejpokročilejší z dostupných nástrojů mají značné rezervy. Některé z nich neumožňují vytvoření odkazů mezi anotacemi, které je potřebné pro jejich strukturování, a některé tuto činnost nemají intuitivní pro laiky (přímé skládání trojic RDF). Vnořování pak většinou není umožněno, nebo je pouze omezené. Problémem je i překrývání anotací, které je potřebné při anotování dvou nezávislých událostí popsaných v jedné větě (např. když různí malíři zcela nezávisle vytvořili obrazy na dané téma nebo dokonce pouze stejnou technikou). Nabízení anotací s odlišením nabídek od uživatelských anotací kromě mnou navrženého nástroje z pokročilých nástrojů umí pouze nástroj Domeo [21], přičemž vedoucí týmu, který vytvořil tento nástroj (Paolo Ciccarese), je současně jedním z klíčových expertů ve skupině, která navrhla formát Web Annotation. Tyto výsledky poukazují na fakt, že je dostupných jen velmi málo pokročilých univerzálních anotačních nástrojů a jejich funkcionalita je často omezená, což je dané primárními případy použití, pro které byly navrženy. Je zde tedy stále velký prostor pro vylepšení, do kterého zasahuje i mnou navržený nástroj 4A.

Praktický experiment srovnávající tři vybrané nástroje pak zkoumal, zda a jakým způsobem jednotlivé aspekty uživatelského rozhraní jednotlivých anotačních nástrojů ovlivňují práci uživatelů. Výsledky ukazují, že příliš strohé uživatelské rozhraní, které nenabízí in-

[2] Representational state transfer http://www.restapitutorial.com/
[3] JavaScript Object Notation http://www.json.org/

tuitivní provádění základních operací s podporou systému, vede na vysoké procento chyb ve vytvářených anotacích i na degradaci rychlosti anotování. Rovněž způsob prezentace příslušnosti atributů ke komplexní anotaci pouhým překrytím fragmentů vede k nejistotě uživatele a zvýšení počtu chyb. Naopak přehledné uživatelské rozhraní odlišující automaticky vygenerované nabídky anotací od uživatelských anotací a vhodný způsob prezentace atributů komplexních vztahů umožňují zvýšit rychlost anotování i kvalitu vytvářených anotací. Vzhledem k tomu, že při anotování uživatel volí kompromis mezi rychlostí práce a kvalitou vytvářených anotací, je zlepšení v obou těchto kritériích, kterého nástroj 4A dosáhl díky zmíněným rozdílům, nezanedbatelné, i když je v řádu jednotek procent a jednotek až desítek sekund na komplexní anotaci události. V každém případě však prezentované výsledky jasně prokazují, že uživatelské rozhraní zásadně ovlivňuje nejen rychlost anotování, ale i kvalitu vytvářených anotací, což bylo jedním z cílů mojí práce.

Srovnání nástroje umožňujícího strukturování pouhým překrytím fragmentů s nástrojem využívajícím strukturované anotace pak prokázalo, že strukturované anotace umožní lepší popis sémantiky a poskytují možnosti lepšího strojového zpracování vhodného mimo jiné i pro jejich následnou vizualizaci.

7.4 Určení optimálního množství zobrazovaných informací

Ve výše popsaném experimentu byla srovnána 3 různá nastavení zobrazení atributů nabídek anotací v anotačním nástroji 4A a sledován dopad jednotlivých nastavení na rychlost a přesnost zjednoznačňování entit. Výsledky prokázaly, že často využívaný typ zobrazení obsahující pouze typ a URL entity v referenčním zdroji vede na největší množství chyb ve vytvořených anotacích. Vzhledem k časté potřebě prohlížení obsahu stránky, na kterou URL odkazuje, současně dochází ke zpomalení procesu anotování.

Následně však bylo prokázáno, že zobrazení většího množství informací, než je nezbytně nutné, vede nejen ke zpomalení procesu anotování, ale i k nárůstu počtu chyb. Uživatelé sice mohou s výhodou hledat klíčová slova z kontextu dané zmínky o entitě v popisu nabízených alternativ, ale velké množství zobrazených informací rovněž rozptyluje jejich pozornost a často se nesprávně rozhodnou na základě nepřesné či nerelevantní informace (např. informace, která je pro dvě zjednoznačňované entity společná, ale uživatel se již při nalezení prvního výskytu nesprávně rozhodne, protože se mu daná informace zdá dostatečně specifická).

Nalezení optimálního množství zobrazovaných informací je obtížnou úlohou a je jen velmi těžko generalizovatelné, neboť se liší pro jednotlivé oblasti zájmu (historie, biomedicína, chemie apod.). Provedené experimenty byly proto zaměřené na zjednoznačňování mnohoznačných názvů osob a míst, protože je tento typ zjednoznačňování velmi častým a uplatňuje se mimo jiné i v doménách (nejen evropských) projektů, kterými se zabývá Výzkumná skupina znalostních technologií na FIT VUT v Brně. Při této úloze je k dispozici celá řada informací od data narození konkrétní osoby, bydliště a dalších osobních informací, přes popis její činnosti a zahraničních cest až po výsledky její práce (např. umělecká díla). Pro člověka je volba nejdůležitějších informací intuitivní, zatímco pro stroj je obtížné určit optimální algoritmus, který dosáhne uspokojivých výsledků. Ideální zjednoznačňovací informace by obsahovala pouze to, co je podstatné pro rozhodnutí – tedy to, v čem se jednotlivé alternativy liší a je to přímo uvedeno v textu. Kdybychom takovou informaci měli, byla by úloha značně zjednodušená a pravděpodobně by ji bylo možné provádět i automaticky. V praktických příkladech je však situace složitější, neboť v dostupných popisech entit, které jsou převážně v přirozeném jazyce, není jednoduché identifikovat klíčové

aspekty pro vyhodnocení rozdílu. Potřebná informace z textu často pouze plyne, ale není v něm přímo uvedena (např. je v textu zmínka o tom, že osoba nebude kandidovat na prezidenta, a z popisu entity víme, že se jedná o politika, aniž by slovo prezident či politik bylo přímo zmíněno, zatímco ostatní alternativy politikem nejsou). V takovém případě je tedy třeba zvolit určitý kompromis, kdy jsou odhadnuty nejdůležitější informace, a jsou-li nedostačující, uživatel musí využít celý popis entity a dohledat chybějící fakta. Testovaný zjednoznačňovací atribut byl proto vytvořen ze zjednoznačňovacích informací v URL na anglickou wikipedii, které jsou vytvářeny lidmi, a vybrané části popisu entity z Wikipedie, kde se autoři obvykle snaží o jasný a výstižný popis. Obdobně lze postupovat i pro jiné oblasti – najít vhodný referenční zdroj a určit algoritmus výběru zjednoznačňovací informace. Námi vyvíjený komponent pro sémantické obohacení textu má architekturu umožňující snadné přizpůsobení pro libovolné textové referenční zdroje.

Nastavení s navrženým zjednoznačňovacím atributem v experimentu i přes relativně jednoduchý algoritmus generování zjednoznačňovací informace vedlo k dosažení nejlepších výsledků ze všech srovnávaných nastavení, a to nejen co se týče kvality vytvořených anotací, ale i rychlosti anotování. Právě zobrazením zjednoznačňovacího atributu a možností rozbalení dalších detailnějších informací, ze kterých se pomocí URL lze dostat až k plnému rozsahu informací dostupných v referenčním zdroji, se mi tedy v anotačním nástroji 4A v rámci možností podařilo optimalizovat množství zobrazovaných informací tak, že jsem zvýšil rychlost anotování i kvalitu vytvářených anotací oproti přístupům využívaných v nejlepších dostupných existujících nástrojích, se kterými byl nástroj 4A srovnán ve výše popsaném experimentu.

7.5 Přínos alternativních nabídek anotací

Dostupné nástroje srovnávané v příloze A, které umožňují poloautomatické anotování textu, při předanotování textu či vygenerování nabídek anotací vytvoří pro každý fragment textu zmiňující nějakou entitu či hodnotu, jednu anotaci či nabídku anotace, která koresponduje s nejpravděpodobnější alternativou. Uživatel pak může nabídku odmítnout (či smazat chybnou anotaci) a manuálně vytvořit správnou anotaci. Manuální vytváření anotace však může být velmi zdlouhavé – i při využití kontrolovaného slovníku je potřeba označit daný fragment, vyhledat příslušnou entitu ve slovníku apod., přičemž jsou všechny entity ze slovníku prezentovány stejně – tedy zcela nezávisle na anotovaném textu. Automatické anotační nástroje však často pracují ve dvou fázích, z nichž v první jsou vyhledány výskyty zmínek o entitách v textu a ve druhé je následně prováděno zjednoznačňování. Některé nástroje pak při zjednoznačňování určí míry důvěry ve správnost jednotlivých anotací a vyberou tu nejlepší, která bude prezentována uživateli. Právě uvedené míry důvěry jsem využil pro sestavení seznamu alternativních nabídek anotací, jehož myšlenkou je usnadnit anotování v daných případech tím, že uživatel bude jedním kliknutím vybírat ze seznamu nejbližších alternativ, nikoliv z celého kontrolovaného slovníku.

Experimenty byly zaměřeny na to, zda navržený koncept prezentace alternativních nabídek anotací napomáhá při anotování, zejména při zjednoznačňování mnohoznačných jmen. Následně byl vyhodnocován vhodný způsob prezentace alternativ uživateli.

Experimenty prokázaly, že využití alternativ značně zvyšuje komfort uživatelů při anotování a zrychluje proces anotování. Výsledky testování různých způsobů prezentace alternativ uživateli sice obecně závisí na kvalitě procesu generování nabídek anotací v pozadí a na četnosti případů, kdy zmíněná entita není pokryta referenčními zdroji, ale okamžité zobrazení všech alternativ se ukazuje jako vhodnější, chceme-li v daném čase preferovat kva-

litu vytvářených anotací před kvantitou. Naopak zobrazení alternativ až ve chvíli odmítnutí nejpravděpodobnější nabídky může mírně zrychlit proces anotování.

Ve druhé sadě experimentů pak bylo prokázáno, že preferujeme-li kvalitu vytvářených anotací před kvantitou, může být výhodnější v alternativách prezentovat více informací. Nicméně jako celkově výhodnější se ukazuje zobrazení stručné zjednoznačňovací informace a možnost zobrazení detailů po kliknutí, neboť takové zobrazení snižuje náročnost zjednoznačňovací úlohy pro uživatele a zvyšuje rychlost procesu anotování, přičemž množství chyb narůstá pouze nepatrně. Pokud bychom chtěli dosáhnout znatelného zvýšení kvality, bylo by možné vynutit to, aby uživatel před konečným rozhodnutím prošel celý seznam alternativ, což se však ukázalo jako nepoužitelné řešení pro více než dvojnásobné zpomalení procesu anotování.

Rovněž se ukázalo, že uživatelé preferují možnost vracet se k nabídkám, které dříve vyloučili. Tedy i když mají možnost definitivního vyloučení dané nabídky, preferují pouze určité označení vyloučených nabídek. Také se ukázalo, že různé výchozí zobrazení může vést na intuitivní využití jiného vzoru interakce uživatelem, neboť možnosti rozbalení detailních informací využilo menší procento uživatelů než možnosti jejich sbalení, přičemž sbalování intuitivně používali pro vylučování jednotlivých nabídek. Při rozbalování pak byl proces vylučování pomocí opětovného sbalení pozorovatelný výjimečně. Výchozí zobrazení a další možnosti interakce tedy mají zásadní vliv na činnost uživatele, což může vést na rozdílnou kvalitu anotací i na čas potřebný na splnění dané úlohy.

Nezávisle na způsobu zobrazení se však jasně prokázalo, že zobrazení alternativních nabídek anotací zvýší rychlost anotování i kvalitu vytvářených anotací, neboť sníží počet manuálně vyplňovaných anotací na nutné minimum a uživatelé primárně vybírají z menšího množství potenciálně korespondujících entit. Dotazníkové šetření rovněž prokázalo zvýšení komfortu pro uživatele.

7.6 Přínos sémantického filtrování

Nový koncept sémantického filtrování zahrnuje šablony pro jednotlivé sémantické typy anotací a asistenci uživateli při výběru hodnoty zvoleného atributu anotovaného konceptu. Šablony lze vytvořit importem ontologie, přičemž anotační systém před uživatelem skryje komplexitu ontologie a vizualizuje ji ve formě stromu anotací s atributy příslušných typů. Když uživatel např. vybere, že chce anotovat událost, jsou mu zobrazeny atributy vyplývající z popisu události v ontologii a jejich typy. Při výběru konkrétního atributu jsou pak v textu zvýrazněny entity či hodnoty korespondující s daným typem či jeho podtypy. Uživatel tak na první pohled vidí potenciální hodnoty daného atributu a může z nich vybrat jedním kliknutím. Není-li hodnota mezi rozpoznanými (zvýrazněnými), může ji doplnit manuálně.

Pro ověření přínosu sémantického filtrování byly provedeny dva experimenty. První se zaměřuje na přínos celkového přístupu k poloautomatickému anotování komplexních hierarchických anotací využitého v nástroji 4A oproti manuálnímu přístupu. Zde je patrný značný pokles doby anotování (téměř na jednu třetinu) se současným zvýšením kvality vytvářených anotací (pokles chyb na méně něž polovinu). Tím je prokázán jasný přínos využitého přístupu, ale není dostatečně určen příspěvek samotného sémantického filtrování.

Pouze na sémantické filtrování se proto zaměřil druhý provedený experiment, kde bylo srovnáno anotování událostí bez sémantického filtování, kdy bylo jako hodnotu atributu možné vybrat jakoukoliv anotaci, přičemž žádná nebyla oproti ostatním zvýrazněna (byly pouze odlišeny barvami dle jednotlivých sémantických typů), s anotováním se zapnutým

sémantickým filtrováním. Výsledky experimentu ukazují, že je proces anotování komplexních událostí pouze vlivem sémantického filtrování rychlejší o 15 %. Relativní snížení obou sledovaných typů chyb současně překročilo 25 %. Tím je jasně prokázáno, že sémantické filtrování zvýší rychlost a kvalitu vytvářených strukturovaných anotací.

7.7 Kolaborativní vytváření ontologie

Specializované nástroje na tvorbu ontologií jako je Protégé[4] vyžadují obecnou znalost tvorby ontologií, zahrnující mimo jiné i porozumění základním konceptům jako jsou třída, individuál a literál, typy omezení apod. Znalostní inženýři, ovládající tyto oblasti, však obvykle nemají dostatečné znalosti z domény, kterou mají pomocí cílové ontologie popsat. Například u moderního průzkumu trhu pro správu reputace obchodních značek při tom specialista vytváří ontologie pro zákazníky z různých odvětví. V této situaci je nutná spolupráce se zákazníkem, který má vynikající znalosti z dané oblasti, ale obvykle nemá znalosti z oblasti tvorby ontologií ani zkušenosti s využitím příslušných nástrojů.

Spolupráce znalostního inženýra se zákazníkem může probíhat pouze nad samotnou vytvářenou ontologií vizualizovanou ve specializovaném nástroji. To však vyžaduje vzájemné vysvětlování, kdy znalostní inženýr zákazníkovi popisuje, jak chápe jednotlivé popisované koncepty, a zákazník jeho informace upřesňuje a doplňuje. Navržené řešení předpokládá spolupráci přímo nad texty, při jejichž analýze má být výsledná ontologie využita. Uživatel pak může navrhovanou ontologii přímo využít k anotování daného textu, čímž vzniknou případy použití umožňující lepší pochopení popisovaných konceptů. Mnou navržený anotační nástroj 4A současně poskytne odstínění uživatele od komplexity formálního popisu ontologie, a umožní tak jednoduchou spolupráci laika se znalostním inženýrem.

Provedený experiment prokázal, že pokud je ke spolupráci znalostního inženýra s informovaným klientem na tvorbě ontologie využit nástroj 4A, dojde k relativnímu zrychlení kolaborativního kroku tvorby ontologie o více než 20 %. Toto zrychlení je sice částečně kompenzováno mírným zpomalením individuální přípravy (zejména na straně znalostního inženýra), ale z vyhodnocení dotazníků po ukončení experimentu plyne, že konzultování ontologie nad reálnými anotovanými daty všichni zúčastnění vnímali jako zásadní vylepšení práce. Tím bylo prokázáno, že kolaborativní rozšiřování ontologie při anotování s využitím reálných anotovaných dat značně usnadňuje proces její tvorby a aktualizace.

7.8 Souhrnné výsledky

V rámci vývoje anotačního systému a provedených experimentů se podařilo splnit všechny stanovené hlavní cíle práce.

Vyvinutý systém splnil nejen požadavky stanovené v rámci této práce, ale i požadavky evropského projektu Decipher, kde usnadnil práci muzejních profesionálů při tvorbě výstav. Jeho kvality ukázalo i srovnání s existujícími nástroji, a to jak v teoretickém tabulkovém přístupu v přílohách A a B, tak i při praktickém srovnávacím experimentu.

Při optimalizaci množství zobrazovaných informací se podařilo docílit zvýšení rychlosti anotování i kvality vytvářených anotací. Navržené koncepty zobrazení alternativních nabídek anotací a sémantického filtrování pak vedly k dalšímu zlepšení.

[4] http://protege.stanford.edu/

Kolaborativní přístup k vytváření ontologie nad reálnými anotovanými daty pak nejen ocenili účastníci experimentu, ale umožnil i dosažení efektivnější spolupráce, a tím i kratšího času v kolaborativním kroku experimentu.

Do budoucna se zaměřím na další vývoj anotačního systému a detailnější experimenty vycházející z vyhodnocení výsledků z jednotlivých kroků realizovaných v této práci. Inspirací pro budoucí vývoj anotačního systému bude např. editor Aloha[5] umožňující editaci přímo v zobrazené webové stránce a v plánu je rovněž nový anotační doplněk do webového prohlížeče Google Chrome[6]. V praktických experimentech bude třeba blíže prozkoumat vliv zobrazované míry důvěry ve správnost jednotlivých alternativních nabídek anotací, detaily ve vzorech interakce uživatele při zjednoznačňování entit (možnost označování či klasifikace alternativních nabídek při procházení seznamu) a další detaily, které by mohly vést k dalším zlepšením procesu anotování a zjednodušení a zpříjemnění této úlohy pro uživatele. Chystám rovněž experimenty na paralelní kolaborativní práci v různých prostředích a vyhodnotím při ní příslušné vzory interakce na týmových úlohách.

[5] http://www.alohaeditor.org/demo/aloha-ui/
[6] https://www.google.com/chrome/browser/desktop/index.html

Kapitola 8

Závěr

V rámci této disertační práce jsem nejprve prostudoval teorii a existující řešení v oblasti anotování textu. Pod pojmem anotace jsem se rozhodl chápat doplňující informaci přidanou k textu bez omezení významu této informace či charakteru anotované části textu. S tímto rozhodnutím souvisí i následné zaměření práce na vytváření komplexních hierarchických anotací bez omezení úrovně zanoření a zkoumání vzorů interakce uživatelů s pokročilými anotačními nástroji při jejich vytváření.

Protože se již po prvotním prozkoumání dostupných informací o existujících anotačních nástrojích ukázalo, že žádný z nich neposkytuje funkcionalitu potřebnou pro uvedený výzkum, v rámci práce jsem se rozhodl vytvořit nový anotační systém. Správnost tohoto rozhodnutí potvrdilo mimo jiné i následné srovnání s existujícími řešeními.

Anotační systém 4A vytvořený v rámci této práce má architekturu klient – server, kde klientskou stranu reprezentují rozšíření pro WYSIWYG editory textu v jazyce JavaScript a doplňky pro webové prohlížeče. Primární zaměření na anotování obsahu na webu vyplynulo z trendu vedoucího k online službám a potřeby metadat pro naplnění vize sémantického webu. Vytvořený systém zahrnuje nejen vlastnosti implementované v řadě současných anotačních systémů, které byly identifikovány jako důležité na základě dostupných studií a vlastního výzkumu, ale i řadu nových konceptů i různých vylepšení existujících metod. Hlavními přínosy jsou nový koncept nabízení anotací s alternativními nabídkami, sémantické filtrování usnadňující vytváření strukturovaných anotací a práce s odběry anotací umožňující pohledy na anotovaný text z různých perspektiv v několika panelech prohlížeče. Za zmínku stojí rovněž nová knihovna pro aktualizaci anotovaných fragmentů umožňující pokročilou aktualizaci anotací po offline editaci textu, pokročilá synchronizace dokumentu a anotací pro kolaborativní práci na paralelním vytváření a anotování textu a nový koncept klonování dokumentu s anotacemi.

V rámci vývoje systému byl navržen nový protokol pro přenos anotací a dalších informací potřebných při činnosti pokročilých anotačních nástrojů. Vzhledem k tomu, že dle dostupných informací bylo jediným existujícím řešením k tomuto účelu API od Lee Feigenbauma z firmy IBM z roku 2004, které není široce využívané, mohl by navržený protokol či koncepty v něm využité umožnit budoucí standardizaci v této oblasti a interoperabilitu různých anotačních nástrojů. Současně byl v rámci této práce navržen i nový formát anotace, který poskytl alternativu k paralelně vznikajícímu novému formátu Web Annotation (dříve Open Annotation) a umožnil vyhodnocení jeho výhod a nevýhod. Některé části specifikace formátu 4A by v budoucnu mohly být základem pro rozšíření formátu Web Annotation o šablony atributů pro jednotlivé sémantické typy anotací, které umožňují využití pokročilých technik pro vytváření hierarchických anotací.

Již prvotní testy systému 4A ukázaly, že podporuje objevování znalostí a pomáhá tak porozumět vědeckým tématům společnou prací na vytváření znalostních struktur.

Systém 4A byl úspěšně nasazen v evropském projektu Decipher, kde byl využit jeho potenciál pro zjednodušení tvorby hierarchických anotací. Muzejní profesionálové ocenili možnost vyhledávání komplexních událostí přímo v textu a jejich okamžitý export do systému, ve kterém vytvářejí popisy výstav na základě historických příběhů a vyprávění. Pokročilé anotování usnadnilo i řadu dalších činností, které uživatelé v muzeích při přípravě výstav běžně provádějí.

Po teoreticky zaměřeném tabulkovém srovnání nového systému s existujícími nástroji byly provedeny praktické srovnávací experimenty s vybranými nástroji z těch, které ze srovnání vyšly jako nejpokročilejší. Ukázalo se, že 4A díky vhodné kombinaci známých i nových funkcí umožňuje dosáhnout lepších výsledků anotování v kratším čase. Rovněž bylo prokázáno, že je přínosné vizuální odlišení nabídek anotací generovaných systémem od anotací zkontrolovaných či vytvořených uživatelem, a projevil se i přínos nového konceptu sémantického filtrování v nástroji 4A. Výsledky nepochybně prokazují, že uživatelské rozhraní zásadně ovlivňuje nejen rychlost anotování, ale i kvalitu vytvářených anotací, čímž byl splněn jeden z cílů mojí práce.

Nástroj 4A byl navržen nejen jako lepší řešení pro anotování textu, ale také jako platforma pro testování nových přístupů v oblasti anotování. Pro ověření řady konceptů totiž historicky vznikaly samostatné nástroje, jejichž primární zaměření na ověření daného konceptu často vedlo k omezením zabraňujícím širšímu využití a v řadě případů i k následnému ukončení vývoje i podpory daného nástroje. Jedním z mých cílů proto bylo vytvořit nástroj, který díky své modulární architektuře a širokým možnostem konfigurace umožní detailní ověřování různých konceptů při vynaložení minimálního úsilí na vývoj potřebného programového vybavení. Následné experimenty ukázaly, že se tento cíl podařilo splnit.

V následujících experimentech jsem se zaměřil na zkoumání vzorů interakce uživatelů s anotačním nástrojem a možností zrychlení procesu anotování a zlepšení kvality vytvářených anotací.

První pokročilé experimenty byly zaměřeny na optimalizaci množství zobrazovaných informací pro zrychlení procesu zjednoznačňování víceznačných jmen entit a zvýšení kvality vytvářených anotací. Výsledky ukazují, že často využívaný přístup, při kterém je zobrazen pouze typ entity a URL na stránku popisující danou entitu v referenčním zdroji, vede na největší chybovost i nejdelší čas anotování. Zobrazení detailních informací přímo u nabídky anotace však rovněž nevede k nejlepším výsledkům. Ve srovnání je zde sice relativní pokles množství chyb o 78 %, stále se však jedná o nezanedbatelné množství. Chyby jsou často způsobeny tím, že velké množství zobrazených informací vede k rozptýlení uživatele a rychlejší únava ze čtení většího množství textu sníží jeho pozornost. Jako nejvhodnější se ukázalo zobrazení stručného zjednoznačňovacího atributu a možnost následného zobrazení detailních informací dle potřeby. Oproti přímému zobrazení detailních informací je zde relativní pokles množství chyb o více než 66 %, oproti nejčastějšímu zobrazení typu a URL o více než 92 %. Současně se oproti zobrazení typu a URL snížil průměrný čas na anotaci o 11,5 % a četnost potřeby návštěvy referenčního zdroje o 96 %. Experiment prokázal, že zvolené optimální množství zobrazovaných informací tedy zvýšilo rychlost anotování i kvalitu vytvářených anotací, čímž byl splněn další z cílů mojí práce. Jednoduchý algoritmus pro sestavení obsahu zjednoznačňovacího atributu by do budoucna bylo možné dále vylepšit o kontextově závislé informace, a potenciálně tak dosáhnout dalšího zlepšení. Toto bude jedním z témat mého budoucího výzkumu v oblasti zjednoznačňování entit.

Druhá sada experimentů byla zaměřena na nový koncept zobrazování alternativních nabídek anotací a vhodný způsob jejich prezentace uživateli. Byl srovnán přístup, kdy jsou alternativy zobrazeny po odmítnutí nejpravděpodobnější nabídky či na vyžádání, s přístupem, kdy jsou ihned zobrazeny všechny alternativy. Zobrazení všech alternativ se ukázalo jako vhodnější, chceme-li v daném čase preferovat kvalitu vytvářených anotací před kvantitou. Nicméně současný nárůst průměrného času na anotaci vedl k nepřesvědčivým výsledkům z pohledu případného celkového zlepšení. Zcela jistě však mají oba způsoby prezentace znatelný vliv na způsob interakce uživatele a vedou k jiné intuitivní realizaci kompromisu mezi rychlostí a kvalitou vytvářených anotací.

Vzhledem k tomu, že bez zobrazení alternativ by nutně muselo dojít k dalšímu zpomalení, neboť manuální tvorba anotace se už v prvotních experimentech ukázala jako velmi náročná a byla by potřebná vždy, když nejpravděpodobnější nabídka není správná, což bylo v provedených experimentech ve 40 % (dle empirického měření přesnosti nástroje pro automatické generování nabídek anotací), je současně prokázáno, že zobrazení alternativních nabídek anotací zvýší rychlost anotování i kvalitu vytvářených strukturovaných anotací. Způsob prezentace pak pouze ovlivní úroveň pro kompromis mezi kvalitou a kvantitou.

Třetí sada experimentů ověřovala přínos nabízení anotací se sémantickým filtrováním při anotování komplexních událostí. Relativní pokles počtu nesprávných hodnot o 61.5 %, chybějících hodnot o 48.5 % a času na anotování události o 64 % jasně ukazují výrazný přínos konceptu nabízení anotací využitého v systému 4A. Nabízení anotací a sémantické filtrování tedy prokazatelně zvyšují rychlost a kvalitu vytvářených strukturovaných anotací.

Čtvrtá sada experimentů pak zjišťovala přínos samotného konceptu sémantického filtrování. Relativní snížení počtu chyb o 26 % a času na anotování vztahu o 15 % prokazují, že přínos sémantického filtrování je nezanedbatelný.

Pátá sada experimentů byla zaměřena na rozdíl mezi okamžitou prezentací detailních informací o entitě, včetně stručné zjednoznačňovací informace, a stručnou prezentací umožňující zobrazení detailů. Tímto byl rozšířen předchozí experiment na optimalizaci zobrazovaných informací a bylo ověřeno, že rozdíly nejsou způsobeny pouze faktem, že je zjednoznačňovací atribut uveden. Při okamžitém zobrazení všech informací dojde oproti stručnému zobrazení k relativnímu snížení počtu chyb o 6 %, ale také k nárůstu času anotování o 35 % a ke značnému snížení komfortu uživatelů, pro které se tak úloha stane výrazně náročnější. Stručné zobrazení s možností rozbalení detailních informací se tedy ukazuje ekonomicky výhodnější alternativou, zatímco detailní zobrazení umožňuje zvýšení kvality anotací. Dalšího zvýšení kvality by pak bylo možné dosáhnout vynucením procházení všech alternativ, které by vedlo na více než dvojnásobný nárůst času, což se zdá ekonomicky nepřijatelné.

Poslední prováděný experiment prokázal, že kolaborativní rozšiřování ontologie nad reálnými daty anotovanými právě vytvářenou ontologií vede k usnadnění a zrychlení tohoto procesu. Tím bylo současně dokázáno, že strukturované anotace umožňují lepší popis sémantiky, přičemž pokud tyto anotace využijeme při kolaborativním anotování v reálném čase, usnadní se tím spolupráce při tvorbě ontologií.

V budoucím výzkumu se zaměřím na zkoumání vlivu zobrazené míry důvěry ve správnost nabídky anotace na rychlost a správnost rozhodování uživatelů, optimalizaci počtu zobrazených alternativních nabídek anotací (tedy práh míry důvěry pro zobrazení alternativy), možnosti vylepšení přístupu ke kontrolovanému slovníku, kolaborativní paralelní anotování větším množstvím uživatelů a na další pokročilé přístupy k anotování textu.

Literatura

[1] Adida, B.; Birbeck, M.; McCarron, S.; aj.: RDFa Core 1.1 - Third Edition. RDFa Working Group, 2015, [Online; navštíveno 5. 1. 2016].
URL http://www.w3.org/TR/2015/REC-rdfa-core-20150317/

[2] Adler, B.; de Alfaro, L.; Pye, I.: Redesigning Scientific Reputation. *The Scientist*, ročník 24, č. 9, Září 2010: str. 30, online:
http://www.the-scientist.com/article/display/57645/.

[3] Agosti, M.; Albrechtsen, H.; Ferro, N.; aj.: DiLAS: a Digital Library Annotation Service. In *Proceedings of Annotation for Collaboration – A Workshop on Annotation Models, Tools and Practices*, 2005, [Online; navštíveno 19. 1. 2011].
URL
http://www.is.informatik.uni-duisburg.de/bib/pdf/ir/Agosti_etal_05.pdf

[4] Agosti, M.; Bonfiglio-Dosio, G.; Ferro, N.: A historical and contemporary study on annotations to derive key features for systems design. *International Journal on Digital Libraries*, ročník 8, 2007: s. 1–19, ISSN 1432-5012, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/93j367635270641h/fulltext.pdf

[5] Agosti, M.; Ferro, N.: An Information Service Architecture for Annotations. In *Pre-proceedings of the 6th Thematic Workshop of the EU Network of Excellence DELOS*, 2004, s. 115–126, [Online; navštíveno 19. 1. 2011].
URL http:
//delos-old.isti.cnr.it/eventlist/dla_04_preproceedings.pdf#page=120

[6] Agosti, M.; Ferro, N.: Annotations as Context for Searching Documents. In *Information Context: Nature, Impact, and Role*, *Lecture Notes in Computer Science*, ročník 3507, editace F. Crestani; I. Ruthven, Springer Berlin / Heidelberg, 2005, s. 155–170, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/he1rc9v2np17a6v3/fulltext.pdf

[7] Agosti, M.; Ferro, N.: A System Architecture as a Support to a Flexible Annotation Service. In *Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures*, *Lecture Notes in Computer Science*, ročník 3664, editace C. Türker; M. Agosti; H.-J. Schek, Springer Berlin / Heidelberg, 2005, s. 147–166, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/90xvqnnwajaktd7l/fulltext.pdf

[8] Agosti, M.; Ferro, N.: Search Strategies for Finding Annotations and Annotated Documents: The FAST Service. In *Flexible Query Answering Systems*, *Lecture Notes*

in Computer Science, ročník 4027, editace H. Larsen; G. Pasi; D. Ortiz-Arroyo; T. Andreasen; H. Christiansen, Springer Berlin / Heidelberg, 2006, s. 270–281, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/p701258v346k80n2/fulltext.pdf

[9] Agosti, M.; Ferro, N.: A formal model of annotations of digital content. *ACM Trans. Inf. Syst.*, ročník 26, Listopad 2007, ISSN 1046-8188, [Online; navštíveno 17. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=1292594&type=pdf&CFID=6442261&CFTOKEN=24665493

[10] Annozilla (Annotea on Mozilla). mozdev.org, 2016, [Online; navštíveno 2. 7. 2016].
URL http://annozilla.mozdev.org/screenshots/moz/annozilla_0.4/

[11] Beckett, D.; McBride, B.: RDF/XML Syntax Specification (Revised). W3C Recommendation, 2004, [Online; navštíveno 2. 1. 2011].
URL https://www.w3.org/TR/REC-rdf-syntax/

[12] Bontcheva, K.; Cunningham, H.; Roberts, I.; aj.: GATE Teamware: A Web-based, Collaborative Text Annotation Framework. *Lang. Resour. Eval.*, ročník 47, č. 4, Prosinec 2013: s. 1007–1029, ISSN 1574-020X, doi:10.1007/s10579-013-9215-6, [Online; navštíveno 2. 8. 2016].
URL http://dx.doi.org/10.1007/s10579-013-9215-6

[13] Bontcheva, K.; Roberts, I.; Derczynski, L.; aj.: The GATE Crowdsourcing Plugin: Crowdsourcing Annotated Corpora Made Easy. In *Proceedings of Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Association for Computational Linguistics, 2014, s. 97–100, [Online; navštíveno 2. 8. 2016].
URL https://gate.ac.uk/sale/eacl2014/gate-crowd/gate-crowd-demo-eacl2014.pdf

[14] Bottoni, P.; Levialdi, S.; Rizzo, P.: An Analysis and Case Study of Digital Annotation. In *Databases in Networked Information Systems*, *Lecture Notes in Computer Science*, ročník 2822, editace N. Bianchi-Berthouze, Springer Berlin / Heidelberg, 2003, s. 216–231, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/u1cu2mx7cexjf01h/fulltext.pdf

[15] mini-introduction to brat. Centro Nacional de Linvestigaciones Oncológicas, 2012, [Online; navštíveno 2. 7. 2016].
URL http://ubio.bioinfo.cnio.es/people/fleitner/treminer/static/introduction.html

[16] Bremer, M.; Gertz, M.: Web Data Indexing Through External Semantic-carrying Annotations. In *Eleventh International Workshop on Research Issues in Data Engineering on Document Management for Data Intensive Business and Scientific Applications*, Washington, DC, USA: IEEE Computer Society, 2001, ISBN 0-7695-0957-6, s. 69–76, [Online; navštíveno 17.1.2011].
URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=916493

[17] Brickley, D.: Basic Geo (WGS84 lat/long) Vocabulary. W3C Semantic Web Interest Group, 2004, [Online; navštíveno 2. 1. 2011].
URL http://www.w3.org/2003/01/geo/

[18] Brush, A. J. B.; Bargeron, D.; Gupta, A.; aj.: Robust annotation positioning in digital documents. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, New York, NY, USA: ACM, 2001, ISBN 1-58113-327-8, s. 285–292, [Online; navštíveno 17. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=365117&type=pdf&CFID=6442261&CFTOKEN=24665493

[19] Bry, F.; Kotowski, J.: A Social Vision of Knowledge Representation and Reasoning. In *SOFSEM 2010: Theory and Practice of Computer Science*, *Lecture Notes in Computer Science*, ročník 5901, editace J. van Leeuwen; A. Muscholl; D. Peleg; J. Pokorný; B. Rumpe, Springer Berlin / Heidelberg, 2010, s. 235–246, [Online; navštíveno 2. 7. 2016].
URL http://www.en.pms.ifi.lmu.de/publications/PMS-FB/PMS-FB-2010-2/PMS-FB-2010-2-paper.pdf

[20] Cadiz, J. J.; Gupta, A.; Grudin, J.: Using Web annotations for asynchronous collaboration around documents. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, CSCW '00, New York, NY, USA: ACM, 2000, ISBN 1-58113-222-0, s. 309–318, [Online; navštíveno 19. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=359002&type=pdf&CFID=6215385&CFTOKEN=18093147

[21] Ciccarese, P.; Ocana, M.; Clark, T.: Open semantic annotation of scientific publications using DOMEO. *Journal of Biomedical Semantics*, ročník 3, č. Suppl 1, January 2012, [Online; navštíveno 2. 8. 2016].
URL http://www.jbiomedsem.com/content/3/S1/S1

[22] Constantopoulos, P.; Doerr, M.; Theodoridou, M.; aj.: On Information Organization in Annotation Systems. In *Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets*, *Lecture Notes in Computer Science*, ročník 3359, editace G. Grieser; Y. Tanaka, Springer Berlin / Heidelberg, 2005, s. 189–200, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/7gh22qnha8uu77a8/fulltext.pdf

[23] Cunningham, H.; Maynard, D.; Bontcheva, K.; aj.: *Text Processing with GATE (Version 6)*. GATE, 2011, ISBN 978-0956599315, [Online; navštíveno 2. 8. 2016].
URL http://tinyurl.com/gatebook

[24] Daciuk, J.; Piskorski, J.; Ristov, S.: Natural Language Dictionaries Implemented as Finite Automata. In *Scientific Applications of Language Methods*, Imperial College Press, 2010, s. 133–204.

[25] DeRose, S.; Maler, E.; Daniel, R.: XML Pointer Language (XPointer) Version 1.0. W3C Last Call Working Draft, 2001, [Online; navštíveno 2. 1. 2011].
URL https://www.w3.org/TR/WD-xptr

[26] Désilets, A.; Gonzalez, L.; Paquet, S.; aj.: Translation the Wiki Way. In *Proceedings of the 2006 international symposium on Wikis*, WikiSym '06, New York, NY, USA:

ACM, Srpen 2006, ISBN 1-59593-413-8, s. 19–32, [Online; navštíveno 17. 1. 2011].
URL http://www.wikisym.org/ws2006/proceedings/p19.pdf

[27] Dytrych, J.; Smrz, P.: Interaction Patterns in Computer-Assisted Semantic Annotation of Text – An Empirical Evaluation. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*, Rome, Italy: SCITEPRESS, 2016, ISBN 978-989-758-172-4.
URL http://www.scitepress.org/DigitalLibrary/PublicationsDetail.aspx?ID=5F3T/EA+bgg=&t=1

[28] Dytrych, J.; Smrz, P.: *Springer Book of ICAART 2016*, kapitola Advanced User Interfaces for Semantic Annotation of Complex Relations in Text. Lecture Notes in Computer Science, Springer International Publishing, 2016.

[29] Enriquez, V.; Judson, S. W.; Weber, N. M.; aj.: Data citation in the wild. 2010, doi:10.1038/npre.2010.5452.1, [Online; navštíveno 2. 7. 2016].
URL http://precedings.nature.com/documents/5452/version/1

[30] Feigenbaum, L.: Standardize annotations with Web services. IBM, Duben 2004, [Online; navštíveno 17. 12. 2010].
URL http://www.ibm.com/developerworks/webservices/library/ws-annotation.html

[31] Garrett, J. J.: Ajax: A New Approach to Web Applications. Adaptive Path Inc., Únor 2005, [Online; navštíveno 28. 12. 2010].
URL http://www.adaptivepath.com/ideas/essays/archives/000385.php

[32] GATE Developer. The University of Sheffield, 2015, [Online; navštíveno 2. 12. 2015].
URL https://gate.ac.uk/family/developer.html

[33] Glover, I.; Xu, Z.; Hardaker, G.: Online annotation – Research and practices. *Computers & Education*, ročník 49, č. 4, 2007: s. 1308–1320, ISSN 0360-1315, doi:http://dx.doi.org/10.1016/j.compedu.2006.02.006, [Online; navštíveno 2. 8. 2016].
URL
http://www.sciencedirect.com/science/article/pii/S0360131506000455

[34] Date and Time on the Internet: Timestamps. Network Working Group, Červenec 2002, [Online; navštíveno 2. 1. 2011].
URL https://www.ietf.org/rfc/rfc3339.txt

[35] Grassi, M.; Morbidoni, C.; Nucci, M.; aj.: Pundit: Creating, Exploring and Consuming Semantic Annotations. In *Proceedings of the 3nd International Workshop on Semantic Digital Archives, Valletta, Malta*, 2013, [Online; navštíveno 2. 8. 2016].
URL http://ceur-ws.org/Vol-1091/paper6.pdf

[36] Hall, M.: Efficiency and Effectiveness: Digital Futures in Innovation. Říjen 2010, prezentace na JISC Future of Research Conference on 19th October 2010, slidy dostupné online [navštíveno 2. 7. 2016]: http://usir.salford.ac.uk/11343/.

[37] Handschuh, S.; Staab, S.; Ciravegna, F.: S-CREAM – Semi-automatic CREAtion of Metadata Knowledge Engineering and Knowledge Management: Ontologies and

the Semantic Web. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, *Lecture Notes in Computer Science*, ročník 2473, editace A. Gómez-Pérez; V. Benjamins, kapitola 32, Berlin, Heidelberg: Springer, Září 2002, ISBN 978-3-540-44268-4, s. 165–184, doi:10.1007/3-540-45810-7_32, [Online; navštíveno 2. 8. 2016].
URL http://dx.doi.org/10.1007/3-540-45810-7_32

[38] Heese, R.; Luczak-Rösch, M.; Paschke, A.; aj.: One Click Annotation. In *Proceedings of the 6th Workshop on Scripting and Development for the Semantic Web, collocated with ESWC*, Ruzica Piskac, Redaktion Sun SITE, Informatik V, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany, 2010, ISSN 1613-0073, [Online; navštíveno 2. 8. 2016].
URL http://ceur-ws.org/Vol-699/Paper4.pdf

[39] Hinze, A.; Heese, R.; Luczak-Rösch, M.; aj.: Semantic enrichment by non-experts: usability of manual annotation tools. In *Proceedings of the 11th international conference on The Semantic Web*, ročník Part I, Springer-Verlag Berlin Heidelberg, 2012, s. 165–181, [Online; navštíveno 2. 8. 2016].
URL http://iswc2012.semanticweb.org/sites/default/files/76490161.pdf

[40] Hinze, A.; Heese, R.; Schlegel, A.; aj.: User-Defined Semantic Enrichment of Full-Text Documents: Experiences and Lessons Learned. In *Theory and Practice of Digital Libraries*, *Lecture Notes in Computer Science*, ročník 7489, editace P. Zaphiris; G. Buchanan; E. Rasmussen; F. Loizides, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-33289-0, s. 209–214, doi:10.1007/978-3-642-33290-6_23, [Online; navštíveno 2. 8. 2016].
URL http://dx.doi.org/10.1007/978-3-642-33290-6_23

[41] Hogenboom, F.; Frasincar, F.; Kaymak, U.; aj.: An Overview of Event Extraction from Text. *DeRiVE*, 2011, [Online; navštíveno 2. 7. 2016].
URL https://pure.tue.nl/ws/files/3560040/55603588702841.pdf

[42] Hu, M.; Liu, B.: Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, New York, NY, USA: ACM, 2004, ISBN 1-58113-888-1, s. 168–177, doi:10.1145/1014052.1014073, [Online; navštíveno 2. 7. 2016].
URL http://doi.acm.org/10.1145/1014052.1014073

[43] Iwanska, L. M.; Shapiro, S. C. (editoři): *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*. Menlo Park, CA: AAAI Press, 2000, ISBN 978-0-262-59021-1.

[44] Kahan, J.; Koivunen, M.-R.: Annotea: An open RDF infrastructure for shared Web annotations. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, New York, NY, USA: ACM, 2001, ISBN 1-58113-348-0, s. 623–632, [Online; navštíveno 19. 1. 2011].
URL http://dl.acm.org/citation.cfm?id=372166

[45] Khalili, A.: Annozilla (Annotea on Mozilla). 2015, [Online; navštíveno 29. 12. 2015].
URL http://rdface.aksw.org/

[46] Khalili, A.; Auer, S.; Hladky, D.: The RDFa Content Editor – From WYSIWYG to WYSIWYM. In *Proceedings of COMPSAC 2012 – Trustworthy Software Systems for the Digital Society*, 2012, [Online; navštíveno 2. 8. 2016].
URL http://svn.aksw.org/papers/2012/COMPSAC2012_RDFaCE/public.pdf

[47] Kim, J.-D.; Ohta, T.; Pyysalo, S.; aj.: Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, Association for Computational Linguistics, Červen 2009, s. 1–9, [Online; navštíveno 2. 7. 2016].
URL http://dl.acm.org/citation.cfm?id=1572342

[48] Kim, S.-M.; Hovy, E.: Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, SST '06, Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, ISBN 1-932432-75-2, s. 1–8, [Online; navštíveno 2. 7. 2016].
URL http://dl.acm.org/citation.cfm?id=1654641.1654642

[49] Koivunen, M.-R.; Swick, R. R.: Metadata Based Annotation Infrastructure offers Flexibility and Extensibility for Collaborative Applications and Beyond. In *K-CAP 2001 Workshop October 21, 2001, Victoria B.C., Canada*, Calgary, Alberta Canada T2N 1N4: Department of Computer Science, University of Calgary, Říjen 2001, [Online; navštíveno 19. 1. 2011].
URL http://semannot2001.aifb.uni-karlsruhe.de/papers/1_annotea.pdf

[50] LaLiberte, D.; Braverman, A.: A protocol for scalable group and public annotations. *Comput. Netw. ISDN Syst.*, ročník 27, Duben 1995: s. 911–918, ISSN 0169-7552, [Online; navštíveno 19. 1. 2011].
URL
http://ac.els-cdn.com/0169755295000172/1-s2.0-0169755295000172-main.pdf?_tid=6c512aae-526b-11e6-b535-00000aacb35f&acdnat=1469453381_8adf36c91b925390962ad4a8560a43e0

[51] ISO 639-2 Language Code List - Codes for the representation of names of languages. Library of Congress, Říjen 2010, [Online; navštíveno 2. 1. 2011].
URL http://www.loc.gov/standards/iso639-2/php/code_list.php

[52] Macháček, J.: *Editor anotací pro CMS*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2013, [Online; navštíveno 2. 8. 2016].
URL http://www.fit.vutbr.cz/study/DP/BP.php?id=15960

[53] Marshall, C. C.: Annotation: from paper books to the digital library. In *Proceedings of the second ACM international conference on Digital libraries*, DL '97, New York, NY, USA: ACM, 1997, ISBN 0-89791-868-1, s. 131–140, [Online; navštíveno 19. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=263806&type=pdf&CFID=6215385&CFTOKEN=18093147

[54] Marshall, C. C.: Toward an ecology of hypertext annotation. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, HYPERTEXT '98, New York, NY, USA: ACM, 1998, ISBN

0-89791-972-6, s. 40–49, [Online; navštíveno 2. 8. 2016].
URL http://dl.acm.org/citation.cfm?id=276632

[55] Marshall, C. C.; Brush, A. J. B.: Exploring the relationship between personal and public annotations. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '04, New York, NY, USA: ACM, 2004, ISBN 1-58113-832-6, s. 349–357, [Online; navštíveno 19. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=996432&type=pdf&CFID=6215385&CFTOKEN=18093147

[56] Maynard, D.: Benchmarking Textual Annotation Tools for the Semantic Web. In *6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco: European Language Resources Association (ELRA), 2008, [Online; navštíveno 2. 8. 2016].
URL https://gate.ac.uk/sale/lrec2008/benchmarking.pdf

[57] Maynard, D.; Dasiopoulou, S.; Costache, S.; aj.: Knowledge Web Project: Deliverable D1.2.2.1.3 – Benchmarking of annotation tools. 2007, [Online; navštíveno 2. 8. 2016].
URL http://knowledgeweb.semanticweb.org/semanticportal/deliverables/D1.2.2.1.3.pdf

[58] MediaWiki. nadace WIKIMEDIA, 2010, [Online; navštíveno 28. 12. 2010].
URL http://www.mediawiki.org/wiki/MediaWiki

[59] Moody, G.: Abundance Obsoletes Peer Review, so Drop It. [Online; navštíveno 2. 7. 2016].
URL http://opendotdotdot.blogspot.com/2010/06/abundance-obsoletes-peer-review-so-drop.html

[60] Moro, A.; Navigli, R.: SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, Colorado, June 2015, s. 288–297, [Online; navštíveno 2. 8. 2016].
URL http://www.aclweb.org/anthology/S15-2049

[61] TinyMCE. Moxiecode Systems AB, 2011, [Online; navštíveno 2. 1. 2011].
URL http://tinymce.moxiecode.com/

[62] Annozilla (Annotea on Mozilla). Mozdev Community Organization, 2010, [Online; navštíveno 29. 12. 2010].
URL http://annozilla.mozdev.org/index.html

[63] Neuwirth, C. M.; Kaufer, D. S.; Chandhok, R.; aj.: Computer support for distributed collaborative writing: defining parameters of interaction. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, New York, NY, USA: ACM, 1994, ISBN 0-89791-689-1, s. 145–152, [Online; navštíveno 19.1.2011].
URL http://portal.acm.org/ft_gateway.cfm?id=192893&type=pdf&CFID=6215385&CFTOKEN=18093147

[64] Nielsen, M.: The Future of Science. 2008, [Online; navštíveno 2. 7. 2016].
URL http://michaelnielsen.org/blog/the-future-of-science-2/

[65] Niranatlamphong, W.; Choochaiwattana, W.; Spring, M.: A conceptual framework for digital annotation system on WWW. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, Srpen 2009, s. 27–31, [Online; navštíveno 17. 1. 2011].
URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5234462

[66] Ovsiannikov, I. A.; Arbib, M. A.; Mcneill, T. H.: Annotation technology. *Int. J. Hum.-Comput. Stud.*, ročník 50, č. 4, Duben 1999: s. 329–362, ISSN 1071-5819, doi:10.1006/ijhc.1999.0247, [Online; navštíveno 2. 8. 2016].
URL http://dx.doi.org/10.1006/ijhc.1999.0247

[67] Phelps, T.; Wilensky, R.: Multivalent Annotations. In *Research and Advanced Technology for Digital Libraries*, *Lecture Notes in Computer Science*, ročník 1324, editace C. Peters; C. Thanos, Springer Berlin / Heidelberg, 1997, s. 287–303, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/d87v173g7671j20p/fulltext.pdf

[68] Phelps, T. A.; Wilensky, R.: Robust Hyperlinks and Locations. *D-Lib Magazine*, ročník 6, Červenec 2000, ISSN 1082-9873, [Online; navštíveno 19. 1. 2011].
URL http://www.dlib.org/dlib/july00/wilensky/07wilensky.html

[69] Phelps, T. A.; Wilensky, R.: Robust intra-document locations. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., Květen 2000, s. 105–118, [Online; navštíveno 19. 1. 2011].
URL http://www9.org/w9cdrom/312/312.html

[70] Poeschl, U.: Interactive open access publishing and public peer review: The effectiveness of transparency and self-regulation in scientific quality assurance. *IFLA Journal*, ročník 36, č. I, 2010: s. 40–46, doi:10.1177/0340035209359573, [Online; navštíveno 2. 7. 2016].
URL http://www.atmospheric-chemistry-and-physics.net/pr_acp_poschl_ifla_journal_2010_interactive_open_access_publishing.pdf

[71] Pontiki, M.; Galanis, D.; Papageorgiou, H.; aj.: SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Association for Computational Linguistics, Červen 2015, s. 486–495, [Online; navštíveno 2. 7. 2016].
URL http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval082.pdf

[72] Quint, V.: Amaya. W3C, 2010, [Online; navštíveno 19. 1. 2011].
URL http://www.w3.org/Amaya/

[73] Reeve, L.; Han, H.: Survey of Semantic Annotation Platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, New York, NY, USA: ACM, 2005, ISBN 1-58113-964-0, s. 1634–1638, doi:10.1145/1066677.1067049, [Online; navštíveno 2. 8. 2016].
URL http://doi.acm.org/10.1145/1066677.1067049

[74] da Rocha, T.; Willrich, R.; Fileto, R.; aj.: Supporting Collaborative Learning Activities with a Digital Library and Annotations. In *Education and Technology for a Better World*, *IFIP Advances in Information and Communication Technology*, ročník 302, editace A. Tatnall; A. Jones, Springer Boston, 2009, s. 349–358, [Online; navštíveno 19. 1. 2011].
URL http://www.springerlink.com/content/gug76w637102x821/fulltext.pdf

[75] Röder, M.; Usbeck, R.; Ngonga Ngomo, A.-C.: Developing a Sustainable Platform for Entity Annotation Benchmarks. In *ESWC Developers Workshop 2015*, 2015, [Online; navštíveno 2. 7. 2016].
URL http://svn.aksw.org/papers/2015/ESWC_GERBIL_semdev/public.pdf

[76] Russell, A.: Comet: Low Latency Data for the Browser. In *Infrequently Noted*, Březen 2006, [Online; navštíveno 28. 12. 2010].
URL http://infrequently.org/2006/03/comet-low-latency-data-for-the-browser/

[77] Saias, J.: Sentiue: Target and Aspect based Sentiment Analysis in SemEval-2015 Task 12. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Association for Computational Linguistics, Červen 2015, s. 767–771, [Online; navštíveno 2. 7. 2016].
URL http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval130.pdf

[78] Sanderson, R.; Ciccarese, P.; de Sompel, H. V.; aj.: Open Annotation Data Model. Open Annotation Community Group, 2013, [Online; navštíveno 5. 1. 2016].
URL http://www.openannotation.org/spec/core/

[79] Sanderson, R.; Ciccarese, P.; Young, B.; aj.: Web Annotation Data Model. W3C Candidate Recommendation, Červenec 2016, [Online; navštíveno 1. 8. 2016].
URL https://www.w3.org/TR/annotation-model/

[80] Sannomiya, T.; Amagasa, T.; Yoshikawa, M.; aj.: A framework for sharing personal annotations on web resources using XML. In *Proceedings of the workshop on Information technology for virtual enterprises*, ITVE '01, Washington, DC, USA: IEEE Computer Society, 2001, ISBN 0-7695-0960-6, s. 40–48, [Online; navštíveno 19. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=545625&type=pdf&CFID=6206112&CFTOKEN=33710793

[81] Schaffert, S.: IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE '06. 15th IEEE International Workshops on*, Červen 2006, ISSN 1524-4547, s. 388–396, [Online; navštíveno 2. 8. 2016].
URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4092241

[82] Schaffert, S.: Semantic social software – semantically enabled social software or socially enabled semantic web. In *Semantics 2006*, 2006, [Online; navštíveno 17. 1. 2011].
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.8638&rep=rep1&type=pdf

[83] Schaffert, S.; Eder, J.; Grünwald, S.; aj.: KiWi – A Platform for Semantic Social Software (Demonstration). In *The Semantic Web: Research and Applications*, *Lecture Notes in Computer Science*, ročník 5554, editace L. Aroyo; P. Traverso; F. Ciravegna; P. Cimiano; T. Heath; E. Hyvönen; R. Mizoguchi; E. Oren; M. Sabou; E. Simperl, Springer Berlin / Heidelberg, 2009, s. 888–892, [Online; navštíveno 17. 1. 2011].
URL http://www.springerlink.com/content/351683094p8m1862/fulltext.pdf

[84] Schaffert, S.; Gruber, A.; Westenthaler, R.: A Semantic Wiki for Collaborative Knowledge Formation. In *Proceedings of SEMANTICS 2005 Conference*, Vienna, Austria, 2006, [Online; navštíveno 6. 1. 2011].
URL http://wwwold.salzburgresearch.at/research/gfx/SemWikiForCollKnowForm_20060120.pdf

[85] Schickler, M. A.; Mazer, M. S.; Brooks, C.: Pan-browser support for annotations and other meta-information on the World Wide Web. In *Proceedings of the fifth international World Wide Web conference on Computer networks and ISDN systems*, Amsterdam, The Netherlands, The Netherlands: Elsevier Science Publishers B. V., 1996, s. 1063–1074, [Online; navštíveno 2. 8. 2016].
URL http://www.sciencedirect.com/science/article/pii/0169755296000608

[86] Semantic MediaWiki. In *Semantic MediaWiki*, semantic-mediawiki.org, 2010, [Online; navštíveno 28. 12. 2010].
URL http://semantic-mediawiki.org/wiki/Semantic_MediaWiki

[87] SharedCopy. SharedCopy, 2010, [Online; navštíveno 29. 12. 2010].
URL http://sharedcopy.com/

[88] ShiftSpace. ShiftSpace, 2010, [Online; navštíveno 29. 12. 2010].
URL http://www.shiftspace.org/

[89] Smrz, P.; Dytrych, J.: Towards New Scholarly Communication: A Case Study of the 4A Framework. In *SePublica*, *CEUR Workshop Proceedings*, ročník 721, Ruzica Piskac, Redaktion Sun SITE, Informatik V, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany, 2011, ISSN 1613-0073, [Online; navštíveno 2. 8. 2016].
URL http://ceur-ws.org/Vol-721/paper-07.pdf

[90] Smrz, P.; Dytrych, J.: Advanced Features of Collaborative Semantic Annotators – the 4A System. In *Proceedings of the 28th International FLAIRS Conference*, Hollywood, Florida, USA: AAAI Press, 2015, [Online; navštíveno 2. 8. 2016].
URL
http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS15/paper/view/10356

[91] Smrz, P.; Otrusina, L.; Dytrych, J.; aj.: DECIPHER Deliverable D4.2.1: Relationship Mining Component. Technická zpráva, Brno University of Technology, 2012.

[92] Smrz, P.; Otrusina, L.; Dytrych, J.; aj.: DECIPHER Deliverable D4.3.1: DECIPHER Semantic Annotator. Technická zpráva, Brno University of Technology, 2013, [Online; navštíveno 2. 8. 2016].
URL http://cordis.europa.eu/docs/projects/cnect/1/270001/080/deliverables/001-DecipherD431SemanticAnnotatorv01.pdf

[93] Specia, L.; Motta, E.: Integrating Folksonomies with the Semantic Web. In *The Semantic Web: Research and Applications*, *Lecture Notes in Computer Science*, ročník 4519, editace E. Franconi; M. Kifer; W. May, Springer Berlin / Heidelberg, 2007, s. 624–639, [Online; navštíveno 17. 1. 2011].
URL http://www.springerlink.com/content/413285327hj53234/fulltext.pdf

[94] Stenetorp, P.; Pyysalo, S.; Topić, G.; aj.: BRAT: A Web-based Tool for NLP-assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, s. 102–107, [Online; navštíveno 19. 7. 2016].
URL http://dl.acm.org/citation.cfm?id=2380921.2380942

[95] Surdeanu, M.; Heng, J.: Overview of the English Slot Filling Track at the TAC2014 Knowledge Base Population Evaluation. In *Proceedings of the TAC-KBP 2014 Workshop*, 2013=4, [Online; navštíveno 2. 8. 2016].
URL http://nlp.cs.rpi.edu/paper/sf2014overview.pdf

[96] Toh, Z.; Su, J.: NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Association for Computational Linguistics, Červen 2015, s. 496–501, [Online; navštíveno 2. 7. 2016].
URL http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval083.pdf

[97] Vaishar, A.: *Folksonomie*. Diplomová práce, ÚK ÚZ FF MU v Brně, 2007, [Online; navštíveno 28. 12. 2010].
URL http://is.muni.cz/th/4157/ff_b/Folksonomie.pdf

[98] Völkel, M.; Krötzsch, M.; Vrandecic, D.; aj.: Semantic Wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, New York, NY, USA: ACM, 2006, ISBN 1-59593-323-9, s. 585–594, [Online; navštíveno 17. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=1135863&type=pdf&CFID=6442261&CFTOKEN=24665493

[99] Völkel, M.; Haller, H.: Conceptual Data Structures (CDS) – Towards an Ontology for Semi-Formal Articulation of Personal Knowledge. In *14th International Conference on Conceptual Structures, Aalborg University, Denmark, 16.07.06*, 2006, [Online; navštíveno 17. 12. 2010].
URL http://www.xam.de/2006/04-cds.pdf

[100] Annotea Project. W3C, 2010, [Online; navštíveno 29. 12. 2010].
URL http://www.w3.org/2001/Annotea/

[101] XML Schema Tutorial. Refsnes Data, 2011, [Online; navštíveno 2. 1. 2011].
URL http://www.w3schools.com/Schema/default.asp

[102] Waltham, M.: Why does one size not fit all in journal publishing? Červen 2010, prezentace na Society for Scholarly Publishing meeting 2010, slidy dostupné online [navštíveno 2. 7. 2016]: http://www.marywaltham.com/SSP_Seminar_2010.pdf.

[103] Wang, A.; Hoang, C.; Kan, M.-Y.: Perspectives on crowdsourcing annotations for natural language processing. *Language Resources and Evaluation*, ročník 47, č. 1,

2013: s. 9–31, ISSN 1574-020X, doi:10.1007/s10579-012-9176-1, [Online; navštíveno 19. 7. 2016].
URL http://dx.doi.org/10.1007/s10579-012-9176-1

[104] Weng, C.; Gennari, J. H.: Asynchronous collaborative writing through annotations. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, CSCW '04, New York, NY, USA: ACM, 2004, ISBN 1-58113-810-5, s. 578–581, [Online; navštíveno 19. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=1031705&type=pdf&CFID=6206112&CFTOKEN=33710793

[105] Semantic wiki. In *Wikipedia – The Free Encyclopedia*, nadace WIKIMEDIA, 2010, [Online; navštíveno 28. 12. 2010].
URL http://en.wikipedia.org/wiki/Semantic_wiki

[106] Wiki. In *Wikipedie – Otevřená encyklopedie*, nadace WIKIMEDIA, 2010, [Online; navštíveno 28. 12. 2010].
URL http://cs.wikipedia.org/wiki/Wiki

[107] Wojahn, P. G.; Neuwirth, C. M.; Bullock, B.: Effects of interfaces for annotation on communication in a collaborative task. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '98, New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1998, ISBN 0-201-30987-4, s. 456–463, [Online; navštíveno 19. 1. 2011].
URL http://portal.acm.org/ft_gateway.cfm?id=274706&type=pdf&CFID=6215385&CFTOKEN=18093147

[108] Wolfe, J.: Annotation technologies: A software and research review. *Computers and Composition*, ročník 19, č. 4, 2002: s. 471 – 497, ISSN 8755-4615, doi:http://dx.doi.org/10.1016/S8755-4615(02)00144-5.
URL
http://www.sciencedirect.com/science/article/pii/S8755461502001445

[109] Yee, K. P.: CritLink: Advanced Hyperlinks Enable Public Annotation on the Web. 2002, http://zesty.ca/pubs/cscw-2002-crit.pdf.
URL http://zesty.ca/pubs/cscw-2002-crit.pdf

[110] Zheng, Q.: *Structured Annotations to Support Collaborative Writing Workflow*. Diplomová práce, The Faculty of Graduate Studies (Computer Science), The University of British Columbia, Prosinec 2005, [Online; navštíveno 17. 12. 2010].
URL https://circle.ubc.ca/bitstream/handle/2429/17720/ubc_2006-0141.pdf?sequence=1

[111] Zheng, Q.; Booth, K.; McGrenere, J.: Co-Authoring with Structured Annotations. Department of Computer Science, University of British Columbia, 2006, [Online; navštíveno 17. 12. 2010].
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.2558&rep=rep1&type=pdf

[112] Zhou, D.; Bian, J.; Zheng, S.; aj.: Exploring social annotations for information retrieval. In *Proceeding of the 17th international conference on World Wide Web*,

WWW '08, New York, NY, USA: ACM, 2008, ISBN 978-1-60558-085-2, s. 715–724, [Online; navštíveno 2. 8. 2016].
URL http://dl.acm.org/citation.cfm?id=1367594

[113] Zhu, J.; Nie, Z.; Liu, X.; aj.: StatSnowball: A statistical approach to extracting entity relationships. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, New York, NY, USA: ACM, 2009, ISBN 978-1-60558-487-4, s. 101–110, doi:http://doi.acm.org/10.1145/1526709.1526724, [Online; navštíveno 2. 8. 2016].
URL http://doi.acm.org/10.1145/1526709.1526724

Přílohy

Seznam příloh

A Srovnání anotačních nástrojů **116**

B Srovnání pokročilých anotačních nástrojů **131**

C Specifikace formátu anotace 4A **133**

D Specifikace 4A protokolu verze 1.1 **137**
D.1 Správa sezení . . . 138
D.2 Uživatelé a skupiny uživatelů . . . 139
D.2.1 Řízení odběru anotací . . . 140
D.2.2 Synchronizace dokumentu . . . 141
D.2.3 Přenos typů anotací . . . 143
D.3 Přenos anotací . . . 146
D.4 Nabízení anotací . . . 146
D.5 Podpora kontrolovaného slovníku . . . 148
D.6 Přenos nastavení . . . 149
D.6.1 Chyby a varování . . . 150
D.7 Potvrzení bez zaslání dat . . . 155

E Zjednodušený příklad komunikace protokolem v. 1.1 **156**

F Specifikace 4A protokolu verze 2.0 **160**
F.1 Komunikace . . . 160
F.1.1 Synchronní kanál . . . 160
F.1.2 Asynchronní kanál . . . 162
F.2 Sezení a přihlášení uživatele . . . 162
F.2.1 Navázání spojení . . . 162
F.2.2 Ukončení spojení . . . 164
F.2.3 Přihlášení uživatele . . . 164
F.2.4 Odhlášení uživatele . . . 165
F.3 Uživatelé a uživatelské skupiny . . . 165
F.3.1 Seznamy uživatelů . . . 166
F.3.2 Seznamy skupin uživatelů . . . 168
F.3.3 Vstup uživatele do skupiny . . . 169
F.3.4 Odchod uživatele ze skupiny . . . 169
F.4 Odběry anotací . . . 169
F.4.1 Vytvoření odběru . . . 170
F.4.2 Rušení odběru . . . 171
F.4.3 Modifikace odběru . . . 171
F.4.4 Seznamy odběrů . . . 171
F.4.5 Přihlášení se k odběru . . . 172
F.4.6 Odhlášení se z odběru . . . 173
F.5 Synchronizace dokumentu . . . 173
F.5.1 Proces synchronizace . . . 173
F.5.2 Znovuposlání obsahu dokumentu . . . 176
F.5.3 Modifikace dokumentu . . . 177

F.6 Typy . . . 180
F.6.1 Jednoduché typy . . . 180
F.6.2 Formát typu anotace . . . 182
F.6.3 Manipulace s typy anotací . . . 184
F.6.4 Získání typů anotací od serveru . . . 185
F.6.5 Získání atributů z ontologie . . . 185
F.7 Anotace . . . 186
F.7.1 Formát anotace . . . 186
F.7.2 Manipulace s anotacemi klienta . . . 191
F.7.3 Vytvoření anotace klientem . . . 192
F.7.4 Modifikace a rušení anotace klientem . . . 193
F.7.5 Znovuzaslání anotací . . . 193
F.8 Návrhy anotací . . . 193
F.8.1 Manipulace s návrhy . . . 193
F.8.2 Získání návrhů . . . 194
F.8.3 Potvrzení návrhu . . . 195
F.8.4 Odmítnutí návrhu . . . 197
F.9 Kontrolovaný slovník . . . 197
F.9.1 Získání typů entit . . . 198
F.9.2 Získání entit . . . 198
F.10 Nastavení . . . 200
F.10.1 Přenos nastavení ze serveru . . . 200
F.10.2 Změna parametrů . . . 201
F.11 Chyby a varování . . . 201
F.11.1 Chybová zpráva . . . 201
F.11.2 Varovací zpráva . . . 202
F.11.3 Seznam chybových kódů . . . 202
F.11.4 Seznam kódů varování . . . 213

G Integrace v systému z projektu Decipher **215**

Příloha A

Srovnání anotačních nástrojů

V této příloze je uvedena tabulka se srovnáním anotačních nástrojů. Tato tabulka je dostupná i ve veřejně sdíleném Google dokumentu na adrese https://docs.google.com/spreadsheets/d/14ionbRVYBQuDOcNLazKfRWYzrkax3qFCspm9SiaG5Aw/edit?usp=sharing

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
(4A)	Doplňky do TinyMCE, FF, Opery a IE	Dokument či fragment	Prostý text	Dynamický strom	Jednoduché i strukturované	Pomocí atributů	XML (4A)	Open Annotation nebo 4A	XPath, offset, délka a obsah	Apache License 2.0
RDFaCE	Doplněk do TinyMCE	Fragment	Data v RDF	Lineární seznam	Jednoduché i strukturované	Pomocí atributů	Offline	RDFa	RDFa přímo v dokumentu	LGPL
Pundit	Bookmarklet	Fragment	Vazba na ontologii, tagy a prostý text	Hledání v ontologii	Strukturované a text	Pomocí trojic	HTTP	? (ve zdrojových textech)	? (ve zdrojových textech, asi XPointer)	Klient AGPL, server ne
Domeo	Bookmarklet	Fragment	Prostý text	Hledání v ontologii	Dle zásuvných modulů	Dle zásuvných modulů	HTTP ?	Open Annotation	Open Annotation	Apache
MAT	Webová aplikace	Fragment	Ne	Lineární seznam	Jednoduché i strukturované	Pomocí atributů	HTTP?	JSON	? (ve zdrojových textech)	BSD
GATE Developer	Desktopová aplikace	Fragment	Ne	Lineární seznam	Pouze textové (páry název – hodnota)	Ne	Pracuje lokálně	XML	uzly přímo v textu	LGPLv3

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
GATE Teamware	Webová aplikace	Fragment	Ne	Lineární seznam	Pouze textové (páry název – hodnota)	Ne	HTTP?	? (ve zdrojových textech)	? (ve zdrojových textech)	AGPLv3
Annotator	Aplikace v JavaScriptu	Fragment	Prostý text, se zásuvným modulem formátovaný	Volné manuálně zadávané tagy	Ne	Ne	JSON	JSON	XPath a offsety	GPLv3 / MIT
A.nnotate	Webová aplikace	Kopie dokumentu – fragment, region obrázku, ...	Prostý text	Volné manuálně zadávané tagy	Ne	Vztahy mezi anotacemi a vztahy s ontologií	URI	JSON	stránka, souřadnice v textu, kontext a obsah	Ne
WebAnno	Webová aplikace	Fragment	Ne	Výběr z předdefinovaného seznamu (změna v administraci)	Přidávají se v administraci a hodnoty se vybírají ze seznamu, který je nutné vytvořit	Vztahy mezi anotacemi (lze zadat typy vztahů a jejich vlastnosti)	HTTP?	binární, export do tsv	?	Apache License 2.0

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
brat	Webová aplikace	Fragment	Prostý text	Výběr typu entity z několika možností	Ne	Vztahy mezi anotacemi (několik typů vztahu s poznámkou)	HTTP?	brat stand-off format (plain text TSV)	offsety	Ano (without restriction)
TextAE	Webová aplikace	Fragment	Ne	Volné manuálně zadávané tagy (entity label)	Ne	Pojmenované vztahy mezi entitami označenými tagy	HTTP (REST)	JSON	Offsety	Ano (Open source)
mae-annotation	Desktopová aplikace	Fragment	Prostý text	Lineární seznam	Ne	Vztahy mezi anotacemi	Pracuje lokálně	XML	Offsety	GPLv3
BioQRator	Webová aplikace	Fragment	Reprezentativní název	Lineární seznam	Pouze textové (páry název – hodnota)	Vztahy mezi anotacemi – vytvářeny odděleně	HTTP ?	BioC nebo CSV	BioC	Ne

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
Annotation Studio	Webová aplikace	Fragment	Prostý text a obrázky	Volné manuálně zadávané tagy	Ne	Ne	HTTP, REST API	? (ve zdrojových textech)	? (ve zdrojových textech)	GPLv2
Hypothes.is	Bookmarklet	Fragment	Formátovaný text	Volné manuálně zadávané tagy (autocomplete pro existující)	Ne	Ne	HTTP ?	Open Annotation	Open Annotation	Zjednodušená BSD
Annozilla	Doplněk do FF	Fragment	Formátovaný text	Seznam předdefinovaných	Ne	Ne	HTTP a XML	RDF Annotea	XPointer	MPL
Amaya	Desktopová aplikace – editor HTML	Dokument či fragment	HTML	Ne	Ne	Ne	HTTP a XML	RDF Annotea	XPointer	Ano (GPL compatible)
AnnotateIt	Aplikace v JavaScriptu, bookmarklet	Fragment	Prostý text	Ne	Ne	Ne	URI	JSON	? (ve zdrojových textech)	GPLv3

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
Marginalia	Webová aplikace v JavaScriptu	Fragment	Prostý text	Ne	Ne	Ne	? (ve zdrojových textech)	RFC 4287	Cesta k blokovému elementu a offset	GPLv3
Knowtator	Aplikace v Javě – doplněk do Protégé	Fragment	Vazba na ontologii	Vazba na ontologii	V ontologii	V ontologii	Pracuje lokálně	lze exportovat XML	offsety	ano (licence nenalezena)
Semantator	Aplikace v Javě – doplněk do Protégé	Fragment	Vazba na ontologii	Vazba na ontologii	V ontologii	V ontologii	Pracuje lokálně	lze exportovat XML	offsety	Ne (pouze binární distribuce)
loomp / OCA	Doplněk do TinyMCE	Fragment	RDF	Vazba na ontologii	V ontologii (+ vztahy v RDF)	V ontologii (+ vztahy v RDF)	HTTP a XML ?	RDFa	RDFa	LGPLv3
AKTive Media	Aplikace v Javě	Fragment nebo region v obrázku	Vazba na ontologii	Vazba na ontologii	V ontologii	V ontologii	Pracuje lokálně	RDF	? (ve zdrojových textech)	ano (licence nenalezena)
GoNTogle	Aplikace v Javě závislá na OpenOffice	Dokument či fragment	Vazba na ontologii	Vazba na ontologii	Ne	V ontologii	viz server	Instance tříd OWL	?	Ne (pouze bin. distribuce)
Magpie	Doplněk do IE a FF	Fragment	Vazba na ontologii	Vazba na ontologii	Ne	V ontologii	Pracuje lokálně	Uzavřený	?	? (nenalezeno)

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
COAT	Webová aplikace v Javě	Fragment dokumentu v korpusu	Vazba na ontologii	Vazba na ontologii	Ne	V ontologii	Webová aplikace – pracuje na serveru	RDF	? (ve zdrojových textech)	LGPL (součást NeOn)
rstWeb	Webová aplikace	Fragment	Ne	Ne	Ne	Pojmenované odkazy mezi fragmenty	HTTP ?	.rs3	? (ve zdrojových textech)	MIT
Egas	Webová aplikace	Fragment	Ne	Lineární seznam	Ne	Vztahy mezi anotacemi	HTTP	Export do XML	offsety?	Ne
Ellogon	Desktopová aplikace	Dokument či fragment (dle úlohy)	Ne	Strom / seznam (dle úlohy)	Jednoduché (v podstatě další tagy)	Pouze omezené při 1 úloze	Pracuje lokálně	? (ve zdrojových textech)	? (ve zdrojových textech)	LGPL
Glozz	Desktopová aplikace	Fragment	Ne	Lineární seznam	Ne	Vztahy mezi anotacemi	Pracuje lokálně	XML	? (ve zdrojových textech)	? (nutná registrace)
Djangology Web Annotator	Webová aplikace	Fragment	Ne	Lineární seznam	Ne	Ne	HTTP	? (ve zdrojových textech)	Offsety	ano (licence nenalezena)

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
Flat	Webová aplikace	Fragment	Ne	Lineární seznam	Ne	Ne	HTTP?	FoLiA	Paragraph Sentence Word	Ano (licence nenalezena)
Anote(2)	Desktopová aplikace	Fragment	Vazba na ontologii	Lineární seznam	Ne	Ne	Offline	? (ve zdrojových textech)	? (ve zdrojových textech)	Ano, freeware (KCCTC and IBB)
PubTator	Webová aplikace	Fragment	Ne	Lineární seznam	Ne	Ne	HTTP ?	PubTator	offsety a obsah	Ne
TagTog	Webová aplikace	Fragment	Ne	Lineární seznam	Ne	Vztahy ?	HTTP ?	Export do TSV, BioC, XML	BioC, ...	Ne
MyMiner	Webová aplikace	Fragment	Ne	Lineární seznam	Ne	Oddělená aplikace	HTTP ?	? (ve zdrojových textech)	? (ve zdrojových textech)	Ne
Argo	Webová aplikace	Fragment	Ne	Strom pro prohlížení, lineární seznam pro výběr	Ne	Ne	HTTP ?	? (ve zdrojových textech)	Offsety	Ne

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
Marky	Webová aplikace	Fragment	Ne	Lineární seznam	Pouze textové (páry otázka – odpověď)	Ne	HTTP ?	? (ve zdrojových textech)	? (ve zdrojových textech)	GPLv3
Wired-Marker	Doplněk do FF	Fragment	Prostý text a obrázky	Lineární seznam	Ne	Ne	? (ve zdrojových textech)	? (ve zdrojových textech)	? (ve zdrojových textech)	Creative Commons
Text Annotation Tool	Modul do Open edX	Fragment	Formátovaný text s odkazy a obrázky	Lineární seznam	Ne	Ne	? (ve zdrojových textech)	? (ve zdrojových textech)	? (ve zdrojových textech)	AGPL
MyStickies	Webová aplikace	Souřadnice v dokumentu	Prostý text	Volné manuálně zadávané tagy	Ne	Ne	Uzavřený	Uzavřený	?	Ne
Diigo	Bookmarklet, doplňky do prohlížečů	Fragment, dokument, screenshot dokumentu	Prostý text, kresba	Volné tagy, doporučené, poslední použité	Ne	Ne	Uzavřený	Uzavřený	?	Ne (komerční)

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
WebNotes	Bookmarklet, doplněk prohlížeče, webová aplikace	Fragment či souřadnice v dokumentu	Prostý text	Výběr ze stromu pro všechny anotace současně	Ne	Ne	Uzavřený	Uzavřený	?	Ne (komerční)
Reframe It	Bookmarklet, doplňky do prohlížečů	Fragment nebo dokument	Prostý text	Volné tagy	Ne	Ne	Uzavřený	Uzavřený	?	Ne (komerční)
Evernote	Desktopová aplikace	Kopie dokumentu	Nadpis a formátovaný text	Volné tagy	Ne	Ne	Uzavřený	Uzavřený	?	Ne (komerční)
Elianto	Webová aplikace	Fragment	Odkaz na Wikipedii a relevance pro dokument	Ne	Ne	Ne	HTTP ?	? (ve zdrojových textech)	? (ve zdrojových textech)	Apache License 2.0
MnM	Aplikace v Javě	Fragment	Vazba na ontologii	Ne	Ne	Vazba na ontologii	Uzavřený	Uzavřený	?	KMI license, nutná registrace

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
Scrible	Bookmarklet	Fragment, dokument	Prostý text, barvy a změny písma	Volné tagy pro celý dokument	Ne	Ne	Uzavřený	Uzavřený	?	Ne (komerční)
iComment (download)	Doplněk do IE a FF	Dokument, fragment či obrázek	Prostý text s omezeným formátováním	Ne	Ne	Ne	Uzavřený	Uzavřený	?	Ne
Yawas	Doplněk do FF a Chrome	Fragment	Ne	Volné tagy pro všechny anotace současně	Ne	Ne	Google Bookmarks	Google Bookmarks	? (pravděpodobně pouze obsah)	Ne
GENIUS	Webová aplikace	Fragment	Text se základním formátováním, odkazy a obrázky	Ne	Ne	Ne	HTTP?	?	?	Ne
Annotary	Webová aplikace	Fragment	Prostý text	Ne	Ne	Ne	HTTP?	?	?	Ne
Authorea	Webová aplikace	Fragment	Formátovaný text	Ne	Ne	Ne	HTTP?	?	?	Ne

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
FloatNotes	Doplněk do FF	Pozice na stránce	Prostý text s podporou syntaxe Markdown	Ne	Ne	Ne	Pracuje lokálně, lze použít Firefox Sync	? (ve zdrojových textech)	? (ve zdrojových textech)	Ano, protože je napsán ve skriptovacích jazycích
Note Anywhere	Rozšíření do Chrome	Pozice na stránce	Prostý text	Ne	Ne	Ne	Pracuje lokálně	?	?	Ne
Notely	Webová aplikace	Pozice na stránce	Prostý text	Ne	Ne	Ne	Pracuje lokálně	? (ve zdrojových textech)	? (ve zdrojových textech)	Ano, ale licence nenalezena
Annotation tool	Webová aplikace	Kopie dokumentu – oblast	Nadpis a prostý text s odkazy	2 předdefinované	Ne	Odpovědi	Uzavřený	Uzavřený	?	Ne
Crocodoc	Webová aplikace	Kopie dokumentu – fragment, bod nebo oblast	Prostý text, kresba, zvýraznění	Ne	Ne	Ne	Uzavřený	Uzavřený	?	Ne (komerční)

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
Notable	Bookmarklet, doplněk do FF	Kopie dokumentu – oblast, fragment zdrojového textu	Ne	Ne	Ne	Ne	Uzavřený	Uzavřený	?	Ne (komerční)
Bounce	Webová aplikace	Screenshot dokumentu – oblast	Prostý text	Ne	Ne	Ne	Uzavřený	Uzavřený	?	Ne
ShiftSpace (on GitHub)	Doplněk do FF	Modifikovaná kopie dokumentu	Prostý text a modifikace dokumentu	Ne	Ne	Ne	? (ve zdrojových textech)	JSON	? (ve zdrojových textech)	MPL
Webklipper	Webová aplikace, doplněk pro FF, Chrome, IE8 (podpora ukončena)	Kopie dokumentu – dokument, fragment či oblast	Prostý text k dokumentu	Ne	Ne	Ne	Uzavřený	Uzavřený	?	Ne
WissKI	Modul pro Drupal 6 (PHP $<$ 5.3)	Fragment	?	Lineární seznam	Jednoduché	?	?	?	?	GPLv2

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
CritLink	Webová aplikace (od 2003 mimo provoz) (ale lze stáhnout)	Dokument či fragment	Titulek, prostý text, rozšířené možnosti	Seznam předdefinovaných	Ne	Pomocí odkazů	HTTP	HTML	Text-Search Fragment Identifier	Ano

Tabulka A.1: Srovnání anotačních nástrojů

Nástroj	Typ klienta	Anotovaný objekt	Obsah anotace	Tagy	Atributy	Strukturování	Protokol	Formát anotace	Popis fragmentu	Dostupnost zdrojového kódu
DM Tools	Webová aplikace	Fragment, oblast v obrázku	Formátovaný text či odkaz	Ne	Ne	Pomocí odkazů	HTTP	Export do RDF Open Annotation	Open Annotation	? (nenalezeno)
Ontomat	Aplikace v Javě	Fragment	Vazba na ontologii	Vazba na ontologii	V ontologii	V ontologii	Pracuje lokálně, v 2. režimu nějak uloží výsledek na server	RDF uložené na konec HTML nebo na server	XPointer	? (Service Temporarily Unavailable)
Awesome Highlighter	Bookmarklet, doplněk do FF	Fragment	Prostý text	Ne	Ne	Ne	Uzavřený	Uzavřený	?	? (načítání stránky neskončí)
Snip.it	Bookmarklet a webová aplikace	Dokument	Prostý text	Seznam předdefinovaných kategorií	Ne	Ne	Uzavřený	Uzavřený	?	Ne
SharedCopy	Bookmarklet	Kopie dokumentu – fragment, oblast	Prostý text, kresba	Ne	Ne	Ne	Uzavřený	Uzavřený	?	? (stránky aktuálně nedostupné)

Tabulka A.2: Srovnání anotačních nástrojů, které již nejsou dostupné

Příloha B

Srovnání pokročilých anotačních nástrojů

V této příloze je uvedena tabulka se srovnáním pokročilých anotačních nástrojů. Tato tabulka je dostupná i ve veřejně sdíleném Google dokumentu na adrese https://docs.google.com/spreadsheets/d/14ionbRVYBQuD0cNLazKfRWYzrkax3qFCspm9SiaG5Aw/edit?usp=sharing

Nástroj	Přidávání atributů	Atributy s literály	Vnořené anotace	Výběr vnoření	Odkazy mezi anotacemi	Výběr odkazu	Vnořování fragmentů	Způsob vnořování	Překrývání fragmentů	Spolupráce na anotování	Nabízení anotací	Odlišení nabídek
4A	Ano, dynamické	Ano, typy dle XSD	Ano	Vložení atributu, výběr textu	Ano	Kliknutím na zvýrazněné, výběrem textu	Ano	Libovolné	Ano	V reálném čase	Ano	Ohraničením + rozdílná manipulace
RDFaCE	Ontologie	Ano, String, URL, Date, ?	Ano, ale musí být i fragmenty	Výběr textu uvnitř fragmentu	Ne	Nelze	Ano	Pouze shora dolů (výběr části)	Pouze úplné	Offline	Ano	Ne, přímo vytvoří anotace
Pundit	Jako trojice	Pouze text	Ne	Nelze	Ano	Sestavení trojice tažením myší	?	?	?	?	Ne	Ne
Domeo	Moduly	Ano, dle modulů	?	?	Ne	Nelze	?	?	?	?	Ano	Ano
MAT	Ano, dynamické?	Ano, String, Int, Float, Boolean	Ne	Nelze	Ano	Kliknutím na zvýrazněné, výběrem textu	?	?	?	Offline	Ano	Ne
GATE Developer	Ano, dynamické?	Pouze text	Ne	Nelze	Ne	Nelze	Ano	Libovolné	Ano	Offline	?	Ne

Tabulka B.1: Srovnání pokročilých anotačních nástrojů

Příloha C

Specifikace formátu anotace 4A

Anglická verze specifikace je na stránkách Výzkumné skupiny znalostních technologií[1].

Formát anotace 4A je založen na RDF/XML, kde podmětem je vždy anotace. Pro zjednodušení implementace klientů i serveru a urychlení zpracování jsou zde však určitá zjednodušení, takže se nejedná o validní RDF/XML. Nicméně transformace (export) do validního RDF je jednoduchá (pouze se převedou N-tice na trojice).

Součástí anotace jsou:

- typ anotace,
- datum a čas vytvoření,
- autor,
- URI anotovaného dokumentu (přesněji kopie tohoto dokumentu na serveru),
- anotované fragmenty,
- textový obsah anotace,
- atributy.

Anotace jsou strukturovány s využitím atributů. Atributy vytvářejí nejen interní strukturu anotace, ale mohou být využity i k vytváření komplexních struktur z více anotací. Každý atribut má název, typ a hodnotu. Atributy mohou mít hodnoty jednoduchých typů (viz atributy typů anotací v příloze D) nebo strukturovaných typů (vnořené anotace nebo odkazy na anotace; typem atributu je typ anotace). Odkazy na anotace umožňují propojování anotací vztahy, seskupování anotací apod. Každá anotace může mít libovolný počet atributů a je-li hodnotou atributu vnořená anotace, tato anotace může mít opět libovolný počet atributů. Úroveň vnoření není omezená.

Typy anotací jsou hierarchicky organizované do stromové struktury. Od verze protokolu 1.1 je zde k dispozici i vícenásobná dědičnost, která strom rozšiřuje na obecný acyklický graf. Nicméně na pozadí je stále základní stromová struktura a další hrany grafu jsou pouze virtuální (jedná se o atributy typu anotace, které se neukládají přímo do anotací). Základními typy mohou být běžné typy anotací (např. poznámka, popis, komentář apod.), ale také základní typy entit (např. věc (thing), osoba, zvíře apod.). Uživatelé pak mohou

[1] http://knot.fit.vutbr.cz/annotations/4A_protocol_1_1_en.html#annotationFormat

vytvářet nové podtypy a vytvořit tak komplexní strom typů. Typy mohou být využity jako tagy a uživatel pak může využít anotace pouze k tagování.

Každá skupina uživatelů sdílí svůj strom typů anotací. Globálně je zde jeden velký strom, ve kterém jednotlivé skupiny uživatelů vytvářejí větve. Typy jsou pak identifikovány skupinou uživatelů, pozicí (cestou) ve stromě a svým názvem.

Když je typ vyplněný v atributu či v textovém poli, je identifikován svým linearizovaným názvem nebo URI. Obě alternativy jsou v jedné skupině uživatelů vzájemně převoditelné. URI končí skupinou uživatelů následovanou cestou ve stromě typů, kde jednotlivé typy jsou odděleny lomítky. Linearizovaný název je cesta ve stromě typů, kde jsou jednotlivé typy odděleny sekvencí znaků „ -> “. V linearizovaném názvu není specifikovaná skupina uživatelů, protože je zde předpoklad, že bude nastavena v jiném formulářovém poli nebo v nastavení (parametr DefaultGroup – viz příloha D). Linearizovaný název je uživatelsky přívětivější a uživateli by měl být zobrazen vždy když je to možné. URI je jednoznačný identifikátor obsahující i skupinu uživatelů, který by měl být využit pro komunikaci mezi klientem a serverem a pro ukládání anotací.

Autor anotace je identifikován systémově jednoznačným URI. URI uživateli přiřazuje server. Zobrazované jméno (celé jméno uživatele nebo přezdívka) bude umístěno v atributu `name` a v dalších atributech mohou být uvedeny další doplňující informace (e-mail, URI obrázku či fotografie apod.).

URI anotovaného dokumentu (atribut `rdf:resource` elementu `a:source`) identifikuje kopii anotovaného dokumentu uloženou na serveru. Před zahájením procesu anotování klient vždy zašle kopii anotovaného dokumentu na server s URI originálního dokumentu. Server pak provede synchronizaci a klientovi vrátí URI pro využití v anotacích. Tímto je umožněno odstranění identifikátoru sezení a dalších nerelevantních informací z URI a jednoznačné určení, o který dokument se jedná.

Pro robustní pozicování anotovaných fragmentů jsou jednotlivé fragmenty popsány pomocí XPath, offsetu a délky anotovaného textu a pomocí samotného anotovaného textového obsahu fragmentu. Když pracujeme s linearizovaným dokumentem, XPath není uveden a pozice je popsána pouze pomocí offsetu, délky a textového obsahu fragmentu. Cesta k anotovanému fragmentu jednoznačně určuje konkrétní oblast dokumentu (např. nadpis, položku seznamu, buňku tabulky apod.), ve které je anotovaný fragment obsažen. Offset a délka určují konkrétní část textu. Textový obsah musí být obsažen také a to pro případ, že dojde ke změně anotovaného textu. Když fragment po změně dokumentu nebude nalezen, bude označen atributem `valid` s hodnotou `false`, ale stále bude obsažen v anotaci, aby byl zachován kompletní původní anotovaný text i v případě, že se anotace přesunula na úroveň celého dokumentu. Anotace může být přiřazena k více částem textu, ale v návrhu uživatelského rozhraní klienta z pohledu uživatele tato možnost nebude obsažena, protože to obvykle není výhodné (fragmenty mohou být anotovány samostatnými anotacemi a shromážděny pomocí atributů v jiné anotaci, která může popsat vztahy mezi významy jednotlivých fragmentů a ne jen mezi jejich výběrem).

Vzhledem k tomu, že je často anotována celá věta nebo její část, výběr textu přes více uzlů DOM není častý. Nicméně pokud tato situace nastane (např. pokud je část textu zvýrazněna tučným písmem), vybraný text bude automaticky rozdělen do více fragmentů jedné anotace (pro jednotlivé uzly DOM). Toto je v mnoha případech nejrobustnější popis takto vybraného textu (nejtolerantnější vzhledem ke změnám dokumentu).

Přesun anotace na úroveň celého dokumentu se nazývá osiření. Tato situace nastane, když se v aktualizovaném dokumentu nepodaří nalézt žádný z fragmentů anotace. Když má anotace více fragmentů, některé fragmenty mohou zůstat platné a k osiření nedojde.

Textový obsah anotace obsahuje textové informace zadané uživatelem. Význam těchto informací může být dán typem anotace, ale ve většině případů se jedná o textovou poznámku, komentář, popis a jiné doplňující informace.

Atributy jsou detailněji popsány v příloze D.

Příklad strukturované anotace

```
<rdf:Description rdf:about="http://example.com/annotations/123456">
  <rdf:type rdf:resource="http://example.com/types/g01/annotation/task"/>
  <a:dateTime rdf:value="2011-01-01T20:00:00Z" />
  <a:author id="http://example.com/authors/123456"
            name="Jaroslav Dytrych"
            address="idytrych@fit.vutbr.cz"/>
  <a:source rdf:resource="http://example.com/documents/getDoc?id=123456"/>
  <a:fragment>
    <a:path>/html/body/div[@id='container']/div[@id='main']/div
    [@id='post1']/DIV[2]/p[1]</a:path>
    <a:offset>22</a:offset>
    <a:length>33</a:length>
    <a:annotatedText>Faculty of Information Technology</a:annotatedText>
  </a:fragment>
  <a:content>
    <![CDATA[
        ...
      ]]>
  </a:content>
  <a:attribute name="place" type="geoPoint">
    <geo:Point>
      <geo:lat>55.701</geo:lat>
      <geo:long>12.552</geo:long>
    </geo:Point>
  </a:attribute>
  <a:attribute name="date" type="nestedAnnotation">
    <rdf:Description
      rdf:about="http://example.com/annotations/123457">
      <rdf:type
      rdf:resource="http://example.com/types/g01/annotation/description"/>
      <a:dateTime rdf:value="2011-01-01T20:00:00Z" />
      <a:author id="http://example.com/authors/123456"
                name="Jaroslav Dytrych"
                address="idytrych@fit.vutbr.cz"/>
      <a:source
      rdf:resource="http://example.com/documents/getDoc?id=123456"/>
      <a:fragment>
        <a:path>/html/body/div[@id='container']/div[@id='main']
        /div[@id='post1']/p[1]</a:path>
        <a:offset>92</a:offset>
        <a:length>17</a:length>
```

```
        <a:annotatedText>14th January 2011</a:annotatedText>
      </a:fragment>
      <a:content>
      <![CDATA[
            ...
          ]]>
      </a:content>
      <a:attribute name="date" type="DateTime"
      rdf:value="2011-01-14T00:00:00Z"/>
    </rdf:Description>
  </a:attribute>
  <a:attribute name="File1" type="Binary"
  rdf:value="TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNlY3RldHVlciBhZGlwa
  XNjaW5nIGVsaXQsIHNlZCBkaWFtIG5vbnVtbXkgbmliaCBldWlzbW9kIHRpbmNpZHVudCB1d
  CBsYW9yZWV0IGRvbG9yZSBtYWduYSBhbGlxdWFtIGVyYXQgdm9sdXRwYXQuIA0KVXQg"/>
  <a:attribute name="txt" type="Text">
    <a:Content>
      <![CDATA[Lorem ipsum dolor sit amet, consectetuer adipiscing elit,
      sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna
      aliquam erat volutpat.]]>
    </a:Content>
  </a:attribute>
  <a:attribute name="reason" type="annotationLink"
               uri="http://example.com/annotations/1234567"/>
</rdf:Description>
```

Příloha D

Specifikace 4A protokolu verze 1.1

V této příloze je popis protokolu pro přenos anotací verze 1.1. I když je tento protokol v době psaní textu této práce stále podporován serverem, již byl nahrazen protokolem verze 2.0 uvedeným v příloze F.

Anglická verze protokolu je na stránkách Výzkumné skupiny znalostních technologií[1].

Protokol pro komunikaci mezi klientem a serverem má 10 částí, jejichž popis je uveden níže. Každá část obsahuje určitý typ zpráv, které mohou být společně kombinovány do jednoho balíku, který se následně zasílá na server či klientovi. Všechny zprávy v balíku jsou obsaženy v elementu `<messages>`, který je současně kořenovým elementem dokumentu se zprávami ve formátu XML. Celý XML dokument pak lze zasílat protokolem na nižší úrovni (např. HTTP).

V případě transportu protokolem HTTP je komunikace ustavena dvěma kanály: AJAX kanál a Comet kanál.

Spojení je nejprve ustaveno přes první (AJAX) kanál. Klient následně automaticky ustaví spojení přes druhý (Comet) kanál. Následně klient zasílá data na server přes AJAX a server odpovídá přes oba kanály (zprávy jsou zaslány přes kanál, který je vhodný pro daný typ informace). Comet se využívá pouze k zasílání asynchronních zpráv ze serveru na klienta. Odpovědi na žádosti o informace, chybové zprávy a další informace generované v odpovědi na zprávu od klienta se zasílají jako odpovědi na zprávu přes AJAX kanál. Informace, které se rozesílají všem zainteresovaným klientům (např. nové anotace, typy anotací a změny v dokumentu), jsou zasílány přes Comet kanál.

Pro zjednodušení klientů se nové informace nezasílají zpět přes AJAX kanál, ale přes Comet (klient je tedy může očekávat pouze na 1 kanálu a zobrazovat všechny změny jednotným způsobem). Server přijímá oba typy požadavků na jedné adrese (je-li server napsaný v jazyce Java, bude příjem zpráv implementován v 1 servletu) a rozlišuje mezi nimi dle obsahu zprávy. Když je v balíku zpráv pouze informace o sezení, jedná se o požadavek Comet kanálu a odpověď bude pozdržena dokud nebudou k dispozici data k odeslání. Pokud je v balíku i jiný obsah, jedná se o požadavek na AJAX a odpověď bude zaslána okamžitě.

Verze protokolu 1.1 oproti 1.0 navíc obsahuje:

- vícenásobnou dědičnost typů anotací,
- komentáře k typům anotací a jejich atributům
- a další drobná vylepšení a opravy chyb.

[1] http://knot.fit.vutbr.cz/annotations/4A_protocol_1_1_en.html

Specifikace protokolu 1.0 již není k dispozici, neboť není nikde využita a její využití by nebylo vhodné.

D.1 Správa sezení

Správa sezení zahrnuje dohodu na verzi protokolu, přihlášení uživatele a odhlášení uživatele.

Klient nejprve zahájí spojení zasláním zprávy `<connect>`, kde uvede nejvyšší podporovanou verzi protokolu:

```
<connect protocolVersion="1.1"/>
```

Server odpoví chybovou zprávou nebo následovně:

```
<connected protocolVersion="1.1" sessionID=""/>
```

Server zašle využitou verzi protokolu v atributu `protocolVersion`. Měl by využít stejnou verzi jako zaslal klient, nebo nejnižší verzi, která je kompatibilní s tou, kterou zaslal klient. Pokud klient nabídne novější verzi než server podporuje, server využije nejvyšší dostupnou verzi. Když klient následně detekuje, že jeho verze není zpětně kompatibilní s verzí od serveru, musí přepnout na verzi serveru nebo jinou zpětně kompatibilní verzi. Pokud kompatibilní verzi nepodporuje, musí se odpojit. Když verze nabídnutá klientem není podporována ze strany serveru, server odpoví chybovým hlášením a klient se může pokusit o připojení nižší verzí protokolu.

Vzhledem k tomu, že novější verze protokolu může být zpětně kompatibilní, klient a server mohou implementovat různé verze. Když má klient nebo server novou funkcionalitu, která druhou stranou není podporovaná, druhá strana jednoduše ignoruje neznámé elementy a atributy. Pokud nová verze nebude zpětně kompatibilní, server nebo klient, který ji implementuje, musí odmítnout spojení s nekompatibilní kombinací verzí.

Ukončení spojení je signalizováno následující zprávou:

```
<disconnect/>
```

Server zašle id sezaní v atributu `sessionID` zprávy `connected`. Od dané chvíle klient musí zasílat id sezení ve všech balících zpráv v atributu `id` elementu `session`:

```
<session id=""/>
```

Přihlášení uživatele je realizováno následující zprávou:

```
<login user="" password=""/>,
```

kde atribut `user` obsahuje uživatelské jméno nebo e-mail a atribut `password` heslo uživatele. Alternativně lze využít externí autentizaci, k čemuž se využívá následující zpráva:

```
<login user="" token="" system=""/>,
```

kde `user` obsahuje uživatelské jméno nebo e-mail a `token` obsahuje autentizační token z externího systému, který uživatele autentizoval, jehož URL je v atributu `system`.

V případě úspěšného přihlášení server odpoví seznamem parametrů nastavení uživatele a následující zprávou:

```
<logged id="" name=""/>,
```

kde `id` je identifikátor uživatele a `name` je jeho zobrazované jméno. V případě, že přihlášení selže, server odpoví chybovou zprávou.

Odhlášení je realizováno následující zprávou:

```
<logout/>
```

Zahájení komunikace s dohodnutím verze protokolu a přihlášením může být provedeno současně. Obdobně lze současně odhlásit uživatele a ukončit sezení.

D.2 Uživatelé a skupiny uživatelů

Pro získání informací o profilech uživatelů může klient zaslat následující zprávu:

```
<queryPersons filter="" withGroups=""/>
```

Pro volbu pole ve filtru může klient využít klíčové slovo `id`, `email` nebo `name` následované dvojtečkou. V případě filtrování dle více polí musí být jednotlivá pole oddělena středníky. Je-li uveden volitelný atribut `withGroups` s hodnotou `true`, v informacích o profilech uživatelů budou zahrnuty i informace o tom, ve kterých skupinách tito uživatelé jsou. Odpověď od serveru bude následující:

```
<persons>
  <person id="" login="" name="" email="" photoURI=""/>
</persons>
```

Atribut `name` obsahuje celé jméno uživatele, atribut `photoURI` může obsahovat URI fotografie uživatele, která může být při zobrazení umístěna u jeho profilu. Pokud byly požadovány informace o skupinách uživatelů, každá značka `person` bude obsahovat značku `userGroups` (viz níže). Značka `person` může obsahovat i další informace o profilu daného uživatele v dalších značkách a textovém obsahu značky.

Pro získání informací o skupinách uživatelů klient zašle následující zprávu:

```
<queryUserGroups filter="" withPersons=""/>
```

Ve filtru lze využít URI skupiny nebo její název. Pokud bude uveden i volitelný atribut `withPersons` s hodnotou `true`, budou v informacích o skupinách zahrnuty i informace o jejich členech. V názvu lze využít zástupné symboly „*“ (libovolný počet libovolných znaků).

Server na tuto zprávu odpoví:

```
<userGroups>
  <group uri="" name=""/>
</userGroups>
```

Pokud byly požadovány informace o členech skupin, v každé značce `group` bude obsažena i značka `persons` (viz výše).

Pro přihlášení ke skupině uživatelů klient zašle následující zprávu:

```
<join group=""/>
```

kde atribut group obsahuje URI skupiny.

K odhlášení uživatele ze skupiny slouží zpráva:

```
<leave group=""/>
```

D.2.1 Řízení odběru anotací

Klient může přijímat pouze anotace zvolených typů ze zvolených zdrojů. Zdrojem může být jiný uživatel nebo URI, který identifikuje anotační server, skupinu uživatelů či jiný obecný zdroj.
Klient se k odběru anotací přihlašuje následující zprávou:

```
<subscribe>
  <source type="" user=""/>
  <source type="" uri=""/>
  <source type=""/>
  <source user=""/>
  <source uri=""/>
</subscribe>
```

Elementů `source` může být libovolné nenulové množství a mohou mít kombinace parametrů, které jsou uvedeny výše. Parametr `user` identifikuje uživatele, `uri` obecný zdroj anotací. Parametr `type` udává typ anotací, přičemž s typem jsou automaticky vybrány i všechny podtypy. V typu může být využit i zástupný symbol „*“, který nahrazuje libovolný počet libovolných znaků.

Pokud není uveden typ, budou přijímány všechny typy anotací (dle skupin, ve kterých se uživatel nachází) z daného zdroje. Pokud není uveden zdroj, budou přijímány všechny anotace daného typu.

K odhlášení může klient využít zprávu `unsubscribe`:

```
<unsubscribe>
  <source type="" user=""/>
  <source type="" uri=""/>
  <source type=""/>
  <source user=""/>
  <source uri=""/>
</unsubscribe>
```

Pro odhlášení platí stejná pravidla jako pro přihlášení (libovolný počet elementů `source`, atd.). Klient automaticky odebírá všechny anotace od svého přihlášeného uživatele a ze všech skupin uživatelů, ve kterých je, pokud se od jejich odběru explicitně neodhlásí.

D.2.2 Synchronizace dokumentu

Synchronizace dokumentu je proces, při kterém server získá kopii aktuální verze anotovaného dokumentu. Pokud server tento dokument získá poprvé, uloží si jej a vrátí klientovi adresu uložené verze, kterou bude klient využívat v anotacích. Pokud má server dokument uložený, porovná novou verzi s uloženou verzí, a pokud se shodují, zašle klientovi adresu uložené verze. Pokud se dokumenty shodují částečně, server vyhodnotí změny. Pokud změny neovlivní žádné anotace, dokument se transparentně aktualizuje (pro klienta stejné jako shoda dokumentů). Pokud by změny ovlivnily některé anotace, server zašle klientovi varování, že některé anotace musely být aktualizovány, a synchronizaci dokončí. V případě zásadnějších změn či zneplatnění anotací dojde k chybě synchronizace a uživatel se musí rozhodnout, zda synchronizaci dokončí a zneplatní tak některé či všechny anotace (vymaže či přesune na úroveň celého dokumentu), nebo nedokončí a bude anotovat uloženou (starší) verzi dokumentu či jiný dokument.

Když se stejným dokumentem pracuje více uživatelů, je možné, že mají novější verzi dokumentu než je uložena v systému, ve kterém je zobrazen anotační klient (je zde verze, která byla využita připojenými uživateli ve chvíli zahájení jejich práce). V tomto případě server také vrátí chybu synchronizace. Z tohoto důvodu chyba synchronizace obsahuje celý obsah dokumentu, který je uložen na anotačním serveru. Uživatel pak může porovnat svoji verzi s tou na serveru a rozhodnout se pro jednu z nich, nebo vytvořit zcela novou verzi jejich kombinací. Po provedení synchronizace bude aktualizována verze všech ostatních připojených klientů se stejným dokumentem.

Protože klient může být i jednoduchý textový editor, který nepracuje se strukturovaným textem, server by měl podporovat i linearizaci dokumentu. V tomto případě klient dokument linearizuje na prostý text a zašle jej serveru v linearizované podobě. Pokud má server strukturovanou podobu, linearizuje ji a porovná se zaslanou. Pokud se linearizované verze neshodují, dojde k chybě synchronizace. Pokud se shodují, synchronizace bude dokončena a server bude každou následně zaslanou anotaci upravovat pro strukturovanou verzi dokumentu.

Syntaxe:

```
<synchronize resource="http://example.com/documents/doc1.txt"
             linearize="false" overwrite="false">
  <content>
    <![CDATA[
      ...
    ]]>
  </content>
</synchronize>
```

Klient pošle serveru kopii anotovaného dokumentu a adresu, ze které pochází. Parametr `resource` udává umístění zdroje (např. URI anotované webové stránky). Element content obsahuje obsah daného dokumentu (pravděpodobně v sekci `CDATA`).

Nepovinný parametr `linearized` udává, zda klient pracuje s linearizovanou verzí dokumentu. Výchozí hodnota je false.

Nepovinný parametr `overwrite` umožňuje vynutit synchronizaci v případě, kdy je dokument s daným URI na serveru uložen, ale jeho obsah se neshoduje. Server v tomto případě

musí nahradit uložený dokument a upravit (přesunout na úroveň celého dokumentu a doplnit textový obsah o informaci o změně) či vymazat všechny anotace, které byly ovlivněny. Klient by tento atribut neměl využít při prvním pokusu o synchronizaci. Jeho použití musí být schváleno uživatelem. Pokud se obsah textu shoduje, server atribut ignoruje. Výchozí hodnota je false.

Při úspěšné synchronizaci server odpoví zprávou:

```
<synchronized resource=""/>
```

Atribut `resource` obsahuje URI kopie anotovaného dokumentu, která je uložena na serveru. Tento URI musí klient využívat v anotacích.

Pokud v průběhu práce dojde k situaci, kdy se obsah anotovaného fragmentu neshoduje s obsahem, který server nalezne na dané pozici v dokumentu, dojde k tzv. rozsynchronizování. V tomto případě server zašle chybovou zprávu a zprávu `<resynchronize/>` (element bez obsahu a atributů), čímž požádá o resynchronizaci.

Klient provede resynchronizaci zasláním následující zprávy:

```
<resynchronize>
  <content>
    <![CDATA[
      ...
    ]]>
  </content>
</resynchronize>
```

Po resynchronizaci je vždy nutné znovu načíst všechny anotace. Server je tedy automaticky zašle v odpovědi.

Pokud klient provádí modifikace dokumentu, musí každou změnu zaslat na server:

```
<textModification path="" offset="" length="">
  <![CDATA[
    Nový obsah vybraného fragmentu ...
  ]]>
</textModification>
```

Atribut `path` udává XPath uzlu DOM dokumentu, ve kterém byla změna provedena. Atributy `offset` a `length` udávají offset a délku změněného fragmentu. Když jsou offset i délka prázdné, dojde k výběru celého uzlu DOM.

V těle elementu `<textModification/>` je potom uveden nový obsah fragmentu (včetně značek HTML). Pokud je vložen nový fragment, délka původního fragmentu je nulová. Pokud je fragment vymazán, element s novým obsahem je prázdný. Pokud pracujeme s linearizovanou verzí dokumentu, XPath je prázdný.

Pokud zvolíme jiný typ uzlu DOM než textový uzel (typicky element node), modifikace bude provedena nad hodnotou uzlu. Např. pro `"<p>content</p>"` má offset řetězce `"content"` hodnotu 3. Když zvolíme uzel atributu (attribute node), offset hodnoty je dán názvem atributu a řetězcem `'="'` – tedy např. pro `"<a href="http://">"` bude mít offset řetězce `"http://"` hodnotu 6.

Server následně zprávu `textModification` rozešle všem klientům pracujícím se stejným dokumentem.

D.2.3 Přenos typů anotací

Přenos typů anotací probíhá obousměrně. Pokud je přidán, upraven či vymazán typ, klient tuto změnu okamžitě zašle serveru a ten ji rozešle všem ostatním klientům, kterých se týká. Klient však nemusí udržovat kompletní strom typů, ale může mít načtenou pouze jeho část. V tomto případě může server buď zasílat všechny změny, nebo může udržovat informaci o tom, které části stromu má klient načtené, a zasílat pouze informace o změnách v těchto částech. Server musí vždy udržovat kompletní strom typů dané skupiny uživatelů (všechny změny jsou mu zasílány).

Klient o část stromu typů žádá zprávou:

```
<queryTypes filter=""/>
```

Atribut `filter` umožňuje získání určitého podstromu typů. Jedná se o linearizovaný název (cestu ve stromě typů, kde jednotlivé typy jsou odděleny řetězcem „ -> “) či URI typu se zástupnými symboly „*“ (libovolný počet libovolných znaků). Filtr tedy umožňuje vybrat podstrom nebo množinu typů, jejichž název či URI obsahuje daný text.

Pokud klient zažádá o strom typů, server na to odpoví zprávou s přidáním typů (viz níže). Pokud filtru nevyhovuje žádný typ, server zašle prázdný seznam typů.

Přenos typů anotací je prováděn následující zprávou:

```
<types>
  <add>
    <type name="" ancestor="" uri="" group="" restrictedAttributes="">
      <attribute name="" type="" required=""/>
      <attribute name="" type="" required="">
        <comment></comment>
      </attribute>
      <ancestor uri=""/>
      <comment></comment>
    </type>
  </add>
  <change/>
  <remove/>
</types>
```

Element `types` obsahuje:

- element `add`, pokud byl přidán typ,
- element `change`, pokud byl upraven typ,
- element `remove`, pokud byl vymazán typ.

V každém ze tří výše uvedených elementů může být obsažen libovolný počet elementů `type`. Atributy tohoto elementu jsou `name` (název typu), `ancestor` (URI primárního rodičovského typu), `uri` (URI typu) a `group` (URI skupiny uživatelů, které typ patří). Pokud je URI primárního rodičovského typu prázdný, jedná se o základní typ (potomek neviditelného kořene stromu typů). URI jednoznačně identifikuje typy. Pokud není uvedena skupina,

určí se z URI typu, podle rodičovského typu nebo podle výchozí skupiny uživatele, který typ přidal.

U každého typu mohou být definovány i výchozí atributy. Tyto jsou potom obsaženy v elementech `attribute`. Každý atribut má název (`name`) a typ (`type`). Pokud se jedná o jednoduchý datový typ, je uveden název tohoto typu. Pokud se jedná o vnořenou anotaci či odkaz na anotaci, je uveden očekávaný typ této anotace (informace o vnoření či odkazování zde není potřebná, protože obě varianty jsou zde významově ekvivalentní). Pokud je atribut povinný, má element `attribute` i atribut `required` s hodnotou `true`.

Jednoduché datové typy atributů jsou:

- `String` – libovolný řetězec
- `URI` – URI
- `DateTime` – datum a čas dle RFC 3339 (iso-date-time with datespec-full)
- `Date` – datum dle RFC 3339 [34] (datespec-full)
- `Time` – čas dle RFC 3339 (time)
- `Integer` – celé číslo
- `Decimal` – desetinné číslo
- `Boolean` – pravdivostní hodnota (`true` nebo `false`)
- `GeoPoint` – geografický bod dle Basic Geo [17] (WGS84 lat/long) Vocabulary (basic)
- `AnyAnnotation` – libovolná vnořená anotace nebo odkaz na anotaci (nemůže být typem atributu anotace – při vytváření anotace zde musí být zvolen konkrétní typ)
- `Person` – uživatel 4A serveru identifikovaný URI nebo e-mailem. Tento typ je zastaralý a v budoucích verzích protokolu může být odstraněn.
- `Duration` – doba trvání dle RFC 3339 (duration)
- `Binary` – binární data (pro ukládání souborů k anotacím, např. ve formátu OpenDocument) – data budou kodována v base64 (server může limitovat velikost souboru, typicky na 2 MB).
- `Text` – dlouhá textová data
- `Image` – URI obrázku (obdoba URI, ale zobrazuje se jinak – náhled s odkazem na obrázek v plné velikosti)
- `Entity` – entita z kontrolovaného slovníku

URI typu obsahuje cestu ke kořeni stromu typů anotací. Jednotlivé prvky této cesty se skládají z názvů primárních předků. Linearizované jméno typu je rovněž složené z názvů primárních předků, ale typ anotace může mít více předků. Všechny předky typu lze určit ze seznamu předků, který je specifikován elementy `ancestor` v těle elementu `type`. Pro vyhledání typu anotace může uživatel využít všechny předky (typ bude zobrazen na více místech stromu pod jednotlivými předky, automatické doplňování jej doplní do linearizovaného

jména za libovolného předka apod.). Nicméně pouze primární předek bude využit v URI a server nemá možnost určit, kterou cestou uživatel typ vybral. Pouze takto sestavená URI bude uložena v anotaci a odpovídající linearizované jméno bude následně u anotace zobrazováno (ostatní předci mohou být zobrazeni společně s primárním, ale tato funkcionalita je volitelná). Atribut `uri` elementu `ancestor` obsahuje URI rodičovského typu anotace.

Typ anotace může mít komentář pro objasnění jeho významu. Obdobně mohou mít komentáře i jeho atributy. V obou případech je komentář obsažen v těle elementu `comment`, který je obsažen v elementu, ke kterému komentář patří. V jednom elementu `type` nebo `attribute` může být maximálně 1 element `comment`. Je doporučeno, aby byl obsah elementu s komentářem obalen v sekci `CDATA`.

Typ anotace může mít takzvaně omezené atributy (hodnota `true` volitelného atributu `restrictedAttributes`). V tomto případě uživatel danému typu nemůže přidávat další atributy a nelze měnit typy existujících atributů. Nicméně pokud je typ atributu současně typem anotace, stále zde lze využít vnořenou anotaci i odkaz na anotaci.

Název typu nelze upravit jinak, než smazáním starého typu a přidáním nového.

Když uživatel k typu anotace přidává nový atribut, musí vyplnit jeho název a zvolit typ hodnoty. Pokud byly typy anotací importovány z ontologie, budou zde atributy, které mají název a typ hodnoty, ale jejich přiřazení k typu není definované. Z tohoto důvodu jsou zde tzv. „atributy z ontologie“, které poskytují alternativní způsob volby názvu a typu hodnoty atributu. Uživatel si pak může zvolit, zda zadá název a typ hodnoty, nebo zda si je zvolí ze seznamu atributů importovaných z ontologie.

O atributy z ontologie klient žádá následující zprávou:

```
<queryAttrFromOnto group=""/>
```

Kde atribut `group` obsahuje URI skupiny uživatelů, do které požadované atributy náleží, nebo hvězdičku pro získání atributů z ontologie ze všech skupin uživatelů.

Server odpoví následující zprávou:

```
<attrsFromOntology>
  <attribute name="" type="" group=""/>
  <attribute name="" type="" group="">
    <comment></comment>
  </attribute>
</attrsFromOntology>
```

Element `attrsFromOntology` může obsahovat libovolné množství elementů `attribute` s jednotlivými atributy. Každý atribut má název (`name`), typ (`type`) a URI skupiny uživatelů, do které náleží (`group`). Jedná-li se o atribut jednoduchého datového typu, v atributu `type` bude název tohoto typu. Jedná-li se o atribut strukturovaného typu (vnořená anotace nebo odkaz na anotaci), využije se URI tohoto typu (informace o vnoření či odkazování zde není potřebná, protože obě varianty jsou zde významově ekvivalentní).

D.3 Přenos anotací

Přenos anotací je prováděn obdobně jako přenos typů:

```
<annotations>
  <add>
    <annotation/>
  </add>
  <change/>
  <remove/>
</annotations>
```

Každé přidání, úprava či vymazání anotace jsou ihned zaslány na server. V elementu `annotations` jsou dle provedené operace obsaženy elementy `add` (přidané) `change` (upravené) a `remove` (vymazané). V každém z těchto elementů může být obsažen libovolný počet elementů `annotation`.

Server každou změnu ihned zašle klientům, kterých se týká (u kterých dotčená anotace patří mezi odebírané). Pokud je přidána anotace, je tato anotace vždy zaslána i klientovi, který ji přidal (skrze Comet kanál), aby získal přidělený identifikátor anotace.

Po synchronizaci či resynchronizaci dokumentu jsou klientovi jako přidané automaticky zaslány všechny anotace, které patří k tomuto dokumentu a vyhovují klientem definovaným požadavkům na odběr anotací (zdroje a typy).

Pokud klient potřebuje znovu načíst některou anotaci (např. po neúspěšném pokusu o editaci) či všechny anotace, může serveru zaslat jednu z následujících dvou variant zprávy:

```
<reload uri="http://example.com/annotations/123456"/>
<reload all="true"/>
```

Atribut `uri` u první varianty zprávy udává URI požadované anotace, atribut `all` u druhé varianty udává, že mají být znovu zaslány všechny anotace.

D.4 Nabízení anotací

Server může klientovi nabídnout automaticky vygenerované anotace k danému dokumentu či jeho části. Klient o nabídku anotací zažádá následovně:

```
<suggestAnnotations path="" offset="" length="" type=""/>
```

Atributy `path`, `offset` a `length` obsahují cestu v dokumentu (XPath), offset a délku fragmentu, ke kterému by měly být nabídnuty anotace. Pokud je uvedena pouze cesta, budou nabídnuty anotace k celému uzlu DOM dokumentu. Pokud není uveden žádný z těchto atributů, budou nabídnuty anotace k celému dokumentu. Volitelný atribut type udává požadovaný typ nabízených anotací. S tímto typem budou současně nabízeny i všechny jeho podtypy.

Jedna kombinace cesty, offsetu a délky může ukazovat pouze na jeden uzel DOM nebo část jeho obsahu. Když klient potřebuje nabídnout anotace pro více uzlů (např. 3 odstavce), musí zaslat více elementů `suggestAnnotations`. Je-li zasláno více elementů, server vezme typ anotace z náhodného z nich (všechny tyto elementy by měly mít stejnou hodnotu

atributu `type`). Když klient potřebuje změnit seznam uzlů, ke kterým mají být anotace nabízeny, musí znovu zaslat celý seznam elementů `suggestAnnotations`. Je-li zaslán nový element či více elementů, aktuální seznam elementů na serveru bude nahrazen právě zaslaným. Zaslaný seznam elementů `suggestAnnotations` tedy nemůže být požadavkem na více typů anotací (v budoucí verzi toto bude změněno, ale v aktuální verzi by to vedlo k neúměrnému zvýšení komplexity zpracování na serveru).

Server odpoví zprávou s nabídkami anotací:

```
<suggestions>
  <annotation tmpId="" confidence=""/>
</suggestions>
```

V elementu `suggestions` může být libovolný počet anotací (elementů `annotation`). Atribut `confidence` udává odhadnutou míru jistoty anotace v procentech. Hodnota může být využita klientem při automatickém přijímání a odmítání anotací. Anotace v nabídce nemají trvalý identifikátor, ale pouze dočasný (`tmpId`). Dočasný identifikátor slouží k informování serveru o manipulaci s nabídkami a vyřazení anotace z nabídky při její aktualizaci. Může být využit i k zaslání informace o odstranění nabídky z klienta v případě aktualizace nabídek po změně textu (klient tedy nepotřebuje možnost aktualizace nabídky – postačuje vložení a odstranění).

Hodnotu `tmpId` lze využít i pro vytváření odkazů mezi nabídkami anotací. V tomto případě bude atribut nabízené anotace obsahovat např.:

```
<a:attribute name="reason" type="annotationLink" tmpId="1234567"/>
```

Všechny odkazované nabídky musí být potvrzeny společně s nabídkou, ve které jsou využity. Toto pravidlo platí rekurzivně – je tedy nutné potvrdit celou strukturu nabídky. Nabídka může odkazovat i na existující anotaci. Při editaci anotace lze rovněž odkazovat na nabídku, ale při uložení změn musí být tato nabídka potvrzena. Když klient potřebuje potvrdit nabídnutou anotaci (v reakci na akci uživatele nebo po automatickém rozhodnutí na základě uživatelského nastavení), pošle ji na server stejně jako jakoukoliv přidanou anotaci (ve zprávě `annotations`), ale element `annotation` bude obsahovat atributy `confirmed` a `tmpId`. Atribut `tmpId` obsahuje dočasný identifikátor nabídky a atribut `confirmed` obsahuje způsob potvrzení dané nabídky. Metody potvrzení jsou:

- `manually` – anotace potvrzená uživatelem,
- `manuallyEdited` – uživatel editoval nabídnutou anotaci a následně ji uložil,
- `automatically` – automaticky potvrzená nabídka (dle konfigurace klienta),
- `automaticallyEdited` – automaticky potvrzená nabídka, která byla automaticky editována (přizpůsobena).

Pokud dojde ke změně dokumentu, může být potřeba upravit nabídku anotací. V tomto případě server okamžitě zašle aktualizaci nabídky anotací. Pro jednodušší implementaci klienta není podporována úprava nabídnutých anotací. Pokud se některá anotace změní, je odstraněna a server nabídne novou verzi. V elementu `suggestions` tedy může být i libovolný počet elementů `delete`:

```
<suggestions>
  <annotation tmpId="" confidence=""/>
  <delete tmpId=""/>
</suggestions>
```

Pokud klient nemá anotaci s daným dočasným identifikátorem, element `delete` ignoruje. Pokud uživatel (či klient, dle nastavení) některou nabídku odmítne, klient zašle na server zprávu:

```
<refusedSuggestions>
  <suggestion tmpId="" method=""/>
</refusedSuggestions>
```

Atribut `method` udává způsob odmítnutí a může mít jednu z následujících hodnot:

- `manually` – odmítnuta uživatelem,
- `automatically` – automaticky odmítnuta dle nastavení klienta.

Pokud klient nechce přijímat další aktualizace nabídek anotací, zažádá server o nabídky anotací k fragmentu dokumentu s nulovou délkou. Tento fragment by měl být na začátku obsahu dokumentu.

D.5 Podpora kontrolovaného slovníku

Kontrolovaný slovník klientovi umožňuje vyhledávání pojmenovaných entit ve slovníku. Server zpracuje odpovědi na dotazy od klienta a zašle je klientovi zpět v příslušné zprávě.

Anotační klient zasílá požadavky na položky ve slovníku ve zprávě `queryEntities`. Tato zpráva má 2 parametry, z nichž jeden je povinný (`filter`) a jeden volitelný. Prvním parametrem je `type` – tento parametr určuje typ (skupinu) entit, ve kterých se má vyhledávat (např.: `Artwork`, `Location` nebo `Artist`). Když je parametr `type` prázdný, server vyhledává ve všech typech entit. Klient může výsledky dále omezit pomocí atributu `filter`.

Příklad požadavku:

```
<queryEntities type="Artwork" filter="Sunflowers"/>
```

nebo

```
<queryEntities type="" filter="Sunflowers"/>
```

Server odpoví asynchronní zprávou `<entities>`, přičemž nemusí garantovat, že odpovědi přijdou ve stejném pořadí, v jakém byly zaslány požadavky. Tato zpráva bude obsahovat všechny entity, které server nalezl. Syntaxe jednotlivých entit je následující:

```
<entity name="" uri= "" type=""  visualRepresentation= "">
     <![CDATA[ ... ]]>
</entity>
```

Kde parametr:

- `name` obsahuje název entity,
- `uri` obsahuje URI entity,
- `type` obsahuje typ (skupinu) entit, který je ekvivalentní s typem v požadavku (byl-li vyplněn),
- `visualRepresentation` obsahuje URI obrázku, který reprezentuje danou entitu.

Značka `entity` obsahuje popis entity v sekci CDATA. Název a URI jsou povinné, ostatní parametry jsou volitelné. Následuje příklad odpovědi na dotaz výše:

```
<entities>
    <entity name="Sunflowers from Arles"
            uri="http://tinyurl.com/sunflowers-from-arles"
            type="Artwork"
       visualRepresentation="http://tinyurl.com/sunflowers-from-arles-img">
     <![CDATA[Sunflowers from Arles by Sergei Chepik is available at
     Catto Gallery.]]>
    </entity>
    <entity name="Sunflowers in a vase with doll, book and box beside them"
            uri="http://artsalesindex.artinfo.com/asi/lots/1515253"
            type="Artwork"/>
</entities>
```

Neodpovídá-li kritériím v dotazu žádná entita, server zašle prázdnou zprávu `<entities>`:

```
<entities>
</entities>
```

Po tom, co uživatel zvolí entitu, je tato klientem přidána jako atribut anotace typu `entity`. Parametry tohoto atributu mají stejný význam jako parametry výsledků dotazu. Atribut typu `entity` má následující syntaxi:

```
<attribute type="entity" name="">
    <entity name="" uri="" type="" visualRepresentation="">
         <![CDATA[ ... ]]>
    </entity>
</attribute>
```

D.6 Přenos nastavení

Nastavení je seznam položek, které mají název a řetězcovou hodnotu. Nastavení lze rozdělit na nastavení serveru a nastavení klienta s tím, že nastavení klienta budou mít prefix „`Client`“ (např.: „`ClientAnnotationTypeColor:Animal->People->Employee`“ s hodnotou „`green`“). Při zobrazení uživateli se potom některá (známá) nastavení zpracují a zobrazí ve formulářích a ostatní se vypíšou v tabulce pro ostatní nastavení, kde je uživatel může měnit.

Konkrétní položky nastavení závisí na implementaci serveru a klienta. Aby nedocházelo k problémům při využití více různých klientů jedním uživatelem, měly by být názvy položek nastavení prefigovány i typem a názvem klienta (např. „`ClientFFExtAnnotFox`“ bude prefix pro rozšíření Firefoxu nazvané AnnotFox) nebo by měly být takové, aby byl jejich význam zcela zřejmý (např. „`ClientDefaultAnnotationType`“ pro výchozí typ anotace).

Vzhledem k tomu, že je předpokládán malý počet položek a malá frekvence přenášení, vždy je přenášen kompletní seznam položek nastavení. Neuvedení položky tedy povede k jejímu odstranění (je-li možné). Pokud položku není možné odstranit (např. nastavení serveru), bude nastavena na výchozí hodnotu.

Nastavení se přenáší zprávou:

```
<settings>
  <param name="" value=""/>
</settings>
```

Elementů `param`, které tvoří jednotlivé položky, může být libovolný (i nulový) počet. Atribut `name` obsahuje název položky, atribut `value` řetězcovou hodnotu položky.

D.6.1 Chyby a varování

Chybové zprávy slouží k informování klienta o chybě. Chybová zpráva obsahuje číslo chyby (atribut `number`) a její textový obsah (v elementu `message`). Přímo v elementu `error` může obsahovat i doplňující informace, které se týkají konkrétní chyby. Při chybě oprávnění při přístupu k anotacím bude obsažena informace o tom, ke kterým zdrojům byl odepřen přístup. Při nezdařené operaci s existující anotací (úprava či mazání) musí chybová zpráva obsahovat informaci o tom, které anotace se týká (kterou anotaci je třeba znovu načíst). Při problému s atributy musí obsahovat i informaci o tom, kterých atributů se týká. Obdobně je to pro další chybové zprávy.

Textový obsah chybových zpráv bude lokalizován do jazyka nastaveného parametrem „`ServerLanguage`“, který bude mít hodnoty dle ISO 639-2 [51] (varianty pro bibliografické účely). Pokud tento parametr nebude nastaven, zprávy budou v angličtině.

Syntaxe:

```
<error number="1">
  <message>
    <![CDATA[
      Chybné přihlašovací jméno nebo heslo.
    ]]>
  </message>
</error>
<error number="2">
  <accessDenied user=""/>
  <accessDenied uri=""/>
  <accessDenied type=""/>
  <accessDenied type="" user=""/>
  <accessDenied type="" uri=""/>
  <message>
```

```
    <![CDATA[
      Nemáte oprávnění ke zvolené anotaci.
    ]]>
  </message>
</error>
<error number="3">
  <message>
    <![CDATA[
      Přístup pouze pro čtení - anotace nebyla uložena.
    ]]>
  </message>
</error>
<error number="4">
  <reload uri="http://example.com/annotations/123456"/>
  <message>
    <![CDATA[
      Editace není povolena.
    ]]>
  </message>
</error>
<error number="5">
  <reload uri="http://example.com/annotations/123456"/>
  <message>
    <![CDATA[
      Mazání není povolené.
    ]]>
  </message>
</error>
<error number="6">
  <reload uri="http://example.com/annotations/123456"/>
  <message>
    <![CDATA[
      Chybí povinné atributy.
    ]]>
  </message>
  <attribute name="" type=""/>
  <attribute name="" type=""/>
</error>
<error number="7">
  <reload uri="http://example.com/annotations/123456"/>
  <message>
    <![CDATA[
      Chybná hodnota atributu.
    ]]>
  </message>
  <attribute name="" type=""/>
</error>
<error number="62">
```

```
  <message>
    <![CDATA[
      Jiný klient pracuje s jinou verzí tohoto dokumentu.
    ]]>
  </message>
  <content>
    <![CDATA[<html><head></head><body><p>John Millington Synge
    travelled to the Aran Islands.</p></body></html>]]>
  </content>
</error>
```

Čísla chyb jsou následující

0. Nepodporovaná verze protokolu.
1. Chybné přihlašovací jméno nebo heslo.
2. Nemáte oprávnění ke zvolené anotaci.
3. Přístup pouze pro čtení - anotace nebyla uložena.
4. Editace není povolena.
5. Mazání není povolené.
6. Chybí povinné atributy.
7. Chybná hodnota atributu.
8. Chybná volba fragmentu - nabízení není možné.
9. Synchronizace selhala - pro danou URI je již uložen jiný obsah dokumentu.
10. Synchronizace není možná.
11. Chyba synchronizace.
12. Editace tohoto typu anotace není povolená.
13. Mazání tohoto typu anotace není povolené.
14. Přidávání typů anotací není povolené.
15. Typ atributu neexistuje.
16. Přidaný typ anotace je chybný.
17. Atributy přidaného typu anotace jsou chybné - některé atributy byly vynechány.
18. Upravovaný typ anotace je chybný - změny nebyly uloženy.
19. Typ anotace neexistuje.
20. Změna názvu, předka či skupiny typu anotace není možná.
21. Chyba v nastavení - nastavení nelze uložit.

22. Synchronizační zpráva bez adresy zdrojového dokumentu.
23. Synchronizační zpráva bez obsahu dokumentu.
24. Chybný zdrojový dokument v anotaci.
25. Chybný anotovaný fragment. Fragment byl zahozen.
26. Chybný atribut v anotaci. Atribut byl zahozen.
27. Neznámá osoba v atributu.
28. Chybná metoda potvrzení nebo dočasný identifikátor nabídky.
29. Editovaná anotace nebyla nalezena. Změny nelze uložit.
30. Mazaná anotace nebyla nalezena. Anotaci nelze vymazat.
31. Identifikátor sezení je neplatný - sezení pravděpodobně expirovalo.
32. Chybný požadavek. Chyba klienta či nekompatibilní verze protokolu.
33. Chyba v modulu serveru.
34. Požadovaná anotace nebyla nalezena.
35. Chybný XPath výraz ve fragmentu anotace. Fragment byl zahozen.
36. Chyba v anotovaném dokumentu.
37. Chybný offset nebo délka v anotovaném fragmentu. Fragment byl zahozen.
38. Chyba v editované anotaci. Změny nelze uložit.
39. Modifikaci textu nelze aplikovat.
40. Neznámý typ anotace - nabízení není možné.
41. Chybný formát data v anotaci. Datum bylo nastaveno na aktuální.
42. Chybný formát data v atributu. Atribut byl zahozen.
43. Dokument nebyl synchronizován. Manipulace s anotacemi není možná.
44. Chybná informace o autorovi anotace.
45. Neznámá skupina uživatelů.
46. Neznámá skupina uživatelů v typu anotace. Skupina bude nastavena alternativním způsobem.
47. Mazání využitých typů anotací není dovoleno. Nejprve je nutné smazat všechny anotace daného typu.
48. Data nebylo možné uložit kvůli interní chybě serveru.
49. Duplicitní URI typu anotace.

50. Popis modifikace textu je chybný.
51. Mazání typů anotací s podtypy není dovoleno. Nejprve je nutné vymazat všechny podtypy.
52. Nejednoznačný fragment (lze jej nalézt na více místech v dokumentu).
53. Chybný URI anotovaného dokumentu.
54. Chybný popis anotace.
55. Seznam předků přidávaného typu anotace je chybný - některé předky bylo nutno vynechat.
56. URI přidávaného typu anotace je chybný nebo není dostatek informací k jeho vytvoření.
57. Nelze se přidat do skupiny administrátorů. Tuto operaci smí provést pouze administrátor.
58. Nemůžete opustit skupinu administrátorů, protože jste jejím posledním členem.
59. Některé anotace by měly být aktualizovány, ale tyto změny nebylo možné uložit.
60. Nelze se připojit k SEC serveru.
61. Část dat nebyla uložena kvůli interní chybě serveru.
62. Jiný klient pracuje s jinou verzí tohoto dokumentu.

100. Neznámá chyba.

Pokud dojde k zneplatnění anotace či k jejímu přesunu na globální úroveň, aniž by došlo k chybě, může být potřeba varovat uživatele. V tomto případě server zašle zprávu s varováním. Klient by měl varování zobrazit uživateli a to buď přímo u dotčené anotace, nebo pomocí nějakého postranního panelu či dialogového okna.

Zpráva s varováním má následující syntaxi:

```
<warning number="1" annotation="http://example.com/annotations/123456">
  <![CDATA[
    Anotace odstraněna.
  ]]>
</warning>
```

Každé varování má číslo (atribut `number`) a textový obsah pro zobrazení uživateli. Pokud se varování týká konkrétní anotace, element `warning` má i atribut `annotation`, který obsahuje URI dané anotace.

Čísla varování jsou:

1. Anotace odstraněna.
2. Anotace osiřela (fragmenty byly zneplatněny).

3. Anotace byla automaticky aktualizována.
4. Anotace částečně osiřela (některé fragmenty byly zneplatněny).
5. Nejste přihlášen. Můžete se pouze přihlásit nebo odpojit (ostatní zprávy budou ignorovány).
6. Fragmenty anotace byly aktualizovány.

100. Jiné varování (interní chyba serveru).

D.7 Potvrzení bez zaslání dat

Pokud je zaslána zpráva, která vyžaduje provedení nějaké operace, je třeba na ni odpovědět, aby druhá komunikující strana měla potvrzeno přijetí zprávy, případně úspěšné provedení operace, kdy nejsou vrácena data. Pokud není zaslána chybová zpráva, předpokládá se úspěšné provedení operace (implicitní potvrzování). Odpověď tedy může obsahovat pouze informace, které jsou důsledkem provedené operace, či jiné užitečné informace. V některých případech však nejsou k dispozici žádné užitečné informace k zaslání (např. po úspěšném vymazání anotace) a je potřeba, aby server či klient potvrdil úspěšnost operace. V tomto případě pošle následující zprávu:

```
<ok/>
```

Tato zpráva může být využita serverem i jako prevence vypršení časového limitu na Comet kanálu.

Příloha E

Zjednodušený příklad komunikace protokolem v. 1.1

V této příloze je popsán zjednodušený příklad komunikace mezi anotačním klientem a serverem s využitím 4A protokolu verze 1.1. Anglická verze je na stránce výzkumné skupiny KNOT[1].

Klient (zahájení komunikace, specifikace verze protokolu a autentizace):

```
<connect/>
<login/>
```

Server (potvrzení, specifikace verze protokolu a nastavení parametrů):

```
<connected/>
<logged/>
<settings/>
```

Klient (přihlášení k odběrům anotací, synchronizace anotovaného dokumentu a žádost o strom typů anotací):

```
<session/>
<subscribe/>
<synchronize/>
<queryTypes/>
```

Server (URI synchronizovaného dokumentu (kopie na serveru), požadované anotace a typy anotací):

```
<synchronized/>
<annotations/>
<types><add/></types>
```

Klient (žádost o přidání nového typu anotace):

```
<session/>
<types><add/></types>
```

[1] http://knot.fit.vutbr.cz/annotations/4A_protocol_1_1_en.html#simplifiedCommunication

Server (vrátí právě přidané typy anotací pro zjednodušení jejich spárování s těmi, co budou zaslány přes Comet kanál (server doplní URI)):

```
<types><add/></types>
```

Server (zašle právě přidané typy anotací) – přes Comet kanál:

```
<types><add/></types>
```

Klient (zašle přidanou anotaci):

```
<session/>
<annotations/>
```

Server:

```
<ok/>
```

Server (zašle zpět právě přidanou anotaci s nově přiřazeným identifikátorem):

```
<annotations/>
```

Klient (zašle změněnou (aktualizovanou) anotaci):

```
<session/>
<annotations/>
```

Server (vrátí chybovou zprávu):

```
<errors/>
```

Klient (požádá o původní obsah dané anotace):

```
<session/>
<reload/>
```

Server (vrátí požadovanou anotaci):

```
<annotations/>
```

Klient (přidání anotace):

```
<session/>
<annotations/>
```

Server (vrátí chybu synchronizace):

```
<errors/>
<resynchronize/>
```

Klient (žádost o resynchronizaci):

```
<session/>
<resynchronize/>
```

Server (vrátí všechny anotace daného textu ze zvolených zdrojů (dle odběrů)):

```
<annotations/>
```

Klient (žádost o změnu nastavení):

```
<session/>
<settings/>
```

Server (vrátí aktuální nastavení):

```
<settings/>
```

Klient (žádost o nabídky anotací):

```
<session/>
<suggestAnnotations/>
```

Server (nabídky anotací):

```
<suggestions/>
```

Klient (žádost o synchronizaci jiného anotovaného dokumentu (změnu dokumentu)):

```
<session/>
<synchronize/>
```

Server (URI synchronizovaného dokumentu (kopie na serveru) a požadované anotace):

```
<synchronized/>
<annotations/>
```

Klient (odhlášení a odpojení):

```
<session/>
<logout/>
<disconnect/>
```

Server:

```
<ok/>
```

Příklad kompletní zprávy z klienta na server

```
<?xml version="1.0" encoding="utf-8" ?>
<messages xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:a="http://nlp.fit.vutbr.cz/annotations/AnnotXMLSchema">
  <session id="0"/>
  <annotations>
    <add>
      <annotation>
        <rdf:Description>
          <rdf:type rdf:resource="http://example.com/types/g1/Person"/>
          <a:dateTime rdf:value="2011-02-16T22:00:00Z"/>
          <a:author id="http://example.com/authors/1"
                    name="Jaroslav Dytrych"
                    address="dytrych@fit.vutbr.cz"/>
          <a:source rdf:resource="http://example.com/documents/getDoc?id=1"/>
          <a:fragment>
            <a:path>/html/body/p[6]/text()</a:path>
            <a:offset>8</a:offset>
            <a:length>8</a:length>
            <a:annotatedText>idytrych</a:annotatedText>
          </a:fragment>
          <a:content>
          <![CDATA[Test annotation type Person with attributes.]]>
          </a:content>
          <a:attribute name="Name" type="String"
                       rdf:value="Jaroslav Dytrych"/>
          <a:attribute name="Email" type="String"
                       rdf:value="dytrych@fit.vutbr.cz"/>
          <a:attribute name="Age" type="Integer" rdf:value="26"/>
          <a:attribute name="Work" type="geoPoint">
            <geo:Point>
              <geo:lat>49.2269</geo:lat>
              <geo:long>16.5956</geo:long>
            </geo:Point>
          </a:attribute>
        </rdf:Description>
      </annotation>
    </add>
  </annotations>
</messages>
```

Příloha F

Specifikace 4A protokolu verze 2.0

V této příloze je popis protokolu pro přenos anotací verze 2.0. První verzi formální specifikace této verze protokolu vytvořil na základě mnou vytvořené verze 1.1 a mých podkladů (požadavky na změny, návrhy konkrétních změn apod.) J. Macháček v rámci své diplomové práce [52]. Anglická verze protokolu je k dispozici na stránkách Výzkumné skupiny znalostních technologií[1].

V definicích zpráv se používají makra, která se uvozují složenými závorkami: `{makro}`.

U některých atributů může klient použít zástupný znak *. Tento znak se interpretuje stejně, jako výraz .* v regulárních výrazech – libovolný počet libovolných znaků (včetně prázdného řetězce).

V popisu protokolu se u odpovědí uvažuje pouze kladná odpověď (nenastala žádná chyba). Pokud při komunikaci dojde k chybě, server posílá klientovi chybovou zprávu.

V protokolu budou postupně představeny typy objektů, které obě strany protokolu používají. Některé z nich mají URI, který jednoznačně identifikuje daný objekt v rámci Internetu. Schémata pro vytváření URI jsou popsána v tabulce F.

F.1 Komunikace

Veškerá komunikace probíhá v jazyku XML. Používají se dva různé kanály:

- *Synchronní komunikace (AJAX)*:
  Používá se pro komunikaci typu dotaz-odpověď. Komunikaci zahájí klient vysláním požadavku. Server požadavek zpracuje a klientovi vrátí odpověď.

- *Asynchronní komunikace (Comet)*:
  Používá se v případech, kdy server potřebuje zaslat klientovi zprávu bez předchozího požadavku. Přes tento kanál server distribuuje například nové anotace a jejich typy nebo změny v dokumentu. Tento kanál se také využívá v situacích, kdy data ze serveru mohou přijít za relativně dlouhou dobu a není žádoucí, aby se na tuto dobu zablokoval synchronní kanál.

F.1.1 Synchronní kanál

U synchronního kanálu je důležité, aby v každém okamžiku bylo otevřeno maximálně jedno spojení se serverem. Pokud klient vygeneruje požadavek v momentě, kdy se čeká na odpověď

[1] http://knot.fit.vutbr.cz/annotations/protocol_2_0.html

Typ objektu	Schéma URI
Anotace	{URI serveru}/Annotations/{serv|temp}/{číslo anotace}
Návrhy anotací	{URI serveru}/Annotations/sugg/{číslo návrhu anotace}
Odběry anotací	{URI serveru}/Annotations/subscriptions/{číslo odběru anotace}
Typy anotací	{URI serveru}/Annotations/types/g{číslo skupiny}/{cesta typu}
Typy atributů	— nemají předepsané formáty —
Uživatelé	{URI serveru}/Annotations/users/{číslo uživatele}
Uživatelské skupiny	{URI serveru}/Annotations/groups/{číslo skupiny}
Dokumenty	{URI serveru}/Annotations/documents/getDoc?id={číslo dokumentu}
Entity kontrolovaného slovníku	— nemají předepsané formáty —
Atributy z ontologie	— nemají předepsané formáty —

Tabulka F.1: Schémata pro vytváření URI objektů využitých při komunikaci

na předchozí požadavek, pak nový požadavek vstupuje do fronty a z ní je vytažen a odeslán serveru až po té, co klient obdrží odpověď na předchozí požadavek.

Dotaz

```
<?xml version="1.0"?>
<messages sessionID="{ID sezení}">
  {seznam zpráv}
</messages>
```

Atribut *sessionID* slouží k identifikaci sezení (viz F.2). Tento atribut je povinný s výjimkou situace, kdy klient není připojen k serveru a žádné sezení tedy nemá vytvořené.

Odpověď

```
<?xml version="1.0"?>
<messages>
  {seznam zpráv}
</messages>
```

Pokud server obdrží požadavek, u kterého se neočekává žádná užitečná odpověď a zároveň nedojde k žádné chybě, server i tak musí nějak odpovědět. V tomto případě odpovídá jednoduchou zprávou *ok*:

```
<?xml version="1.0"?>
<messages>
  <ok/>
</messages>
```

F.1.2 Asynchronní kanál

Dotaz

Odpovědi na požadavky na Comet kanálu typicky přicházejí po relativně dlouhé době. Počet aktivních spojení se serverem na jednu webovou stránku je přitom omezen prohlížeči. Z tohoto důvodu není možné, aby si více klientů vytvořilo vlastní Comet kanál. Na straně klienta tudíž musí existovat mechanismus, který spravuje jediný Comet kanál pro všechny připojené anotační editory. Klient musí v požadavku uvést ID sezení daných editorů.

```
<?xml version="1.0"?>
<messages>
  {seznam ID sezení formátu:
    <session id="{ID sezení}"/>
  }
  <comet/>
</messages>
```

Odpověď

Server posílá odpověď určenou vždy jen jednomu připojenému editoru. Pokud se v jednom okamžiku objeví více zpráv, které je potřeba poslat různým editorům v rámci jednoho Comet kanálu, pak server tyto odpovědi posílá jednotlivě jako odpovědi na nové požadavky.

```
<?xml version="1.0"?>
<messages sessionID="{ID sezení}">
  {seznam zpráv}
</messages>
```

Podle atributu *sessionID* klient pozná, kterému editoru na stránce má zprávu předat.

V případech, kdy vyprší časový limit pro poslání odpovědi a server nemá žádná data, která by klientovi poslal, server odpovídá jednoduchou zprávou *ok*, přičemž v atributu *sessionID* uvede ID sezení libovolného připojeného editoru:

```
<?xml version="1.0"?>
<messages sessionID="{ID sezení}">
  <ok/>
</messages>
```

Po obdržení odpovědi od serveru musí klient poslat další Comet požadavek. Ve zprávě znovu specifikuje ID sezení všech připojených editorů.

F.2 Sezení a přihlášení uživatele

F.2.1 Navázání spojení

Dotaz

Klient naváže spojení se serverem zprávou *connect*. V elementu *messages* neuvádí *sessionID*, protože jej ještě nezná. Zpráva *connect* má povinný atribut *protocolVersion* – klient uvede číslo nejvyšší podporované verze.

Pomocí volitelného atributu *attachCometTo* je možné donutit server přidat nově vytvořené sezení k již existujícímu Comet kanálu. To se hodí v případech, kdy se na jedné webové stránce spustí více anotačních editorů. Klient do tohoto atributu uvede číslo sezení libovolného editoru, který už je k Comet kanálu přiřazen.

```
<connect protocolVersion="{verze protokolu}"
         {volitelně: attachCometTo="{ID sezení}"}/>
```

Příklad:

```
<connect protocolVersion="2.0"
         attachCometTo="34"/>
```

Odpověď

Pokud nenastane žádný problém, server odpoví zprávou *connected*:

```
<connected protocolVersion="{verze protokolu}"
           sessionID="{id sezení}"/>
```

Atributy:

- *protocolVersion* – verze protokolu, kterým se bude komunikovat
- *sessionID* – identifikátor sezení, který klient musí od tohoto okamžiku dávat jako atribut do každého balíčku zpráv *messages*

Pokud server nepodporuje verzi protokolu, jakou klient požaduje, server vrátí nejvyšší možnou verzi, kterou podporuje a která je zároveň zpětně kompatibilní s verzí, jakou navrhl klient. Klient by měl touto verzí protokolu komunikovat bez problémů. Pokud ale server žádnou takovou verzi nepodporuje, vrátí chybovou zprávu. Pokud je klient schopen komunikovat ještě jinými verzemi protokolu, může postupně zkoušet připojit se k serveru s těmito verzemi.

Příklad:

```
<connected protocolVersion="2.0"
           sessionID="17"/>
```

Vytvoření Comet kanálu

Klient po obdržení odpovědi vytvoří *Comet* kanál, přes který bude probíhat asynchronní komunikace. Zpráva, kterou se kanál otevře, vypadá následovně:

```
<comet/>
```

Vždy, když server vrátí klientovi odpověď, klient musí kanál znovu otevřít zprávou *comet*. Kanál může být otevřený jen omezenou dobu, protože webový prohlížeč nastavuje časový limit na obdržení odpovědi. Pokud má server už dlouho otevřený kanál a nepotřebuje klientovi zaslat žádné zprávy, odpoví alespoň zprávou *ok*. Tím se kanál uzavře a klient jej následně znovu otevře.

F.2.2 Ukončení spojení

Klient ukončí spojení zprávou *disconnect*:

```
<disconnect/>
```

V případě neúspěšného připojení k serveru klient tuto zprávu neposílá (není totiž připojen).

Server na tuto zprávu neodpovídá.

F.2.3 Přihlášení uživatele

Dotaz

Uživatel se přihlásí zprávou *login*. Existují 2 alternativy:

- *Běžné přihlášení*

  ```
  <login login="{uživatelské jméno}"
         password="{hashovaná podoba hesla}"/>
  ```

  Příklad:

  ```
  <login login="admin"
         password="MD5(mysecretpassword)"/>
  ```

- *Externí přihlášení s tokenem*
  Token generuje třetí strana.

  ```
  <login login="{uživatelské jméno}"
         token="{token}"
         system="{URI systému}"/>
  ```

  Povinné parametry jsou:

  - *login*: uživatelské jméno,
  - *token*: token vygenerovaný systémem,
  - *system*: URI systému, který token vygeneroval.

  Příklad:

  ```
  <login login="dytrych"
         token="we9fg2n2j9230vdvjla095"
         system="http://example.com/system"/>
  ```

Odpověď

Pokud vše proběhne v pořádku, server uživatele přihlásí a klientovi pošle zprávy *logged* a *settings*:

```
<logged uri="{URI uživatele}"
        login="{login uživatele}"
        name="{celé jméno uživatele}"
        email="{e-mail uživatele}"
        image="{URI obrázku uživatele}"/>
{zpráva settings}
```

Atributy elementu *logged* odpovídají povinným atributům uživatele. Zpráva *settings* bude vysvětlena později.

Poznámka: V případě použití *OpenID* se *login* nepoužívá, a proto to ani není povinný údaj.

Příklad:

```
<logged uri="http://example.com/Annotations/users/89"
        login="dytrych"
        name="Jaroslav Dytrych"
        email="dytrych@fit.vutbr.cz"
        image="http://example.com/images/photo.png"/>
```

F.2.4 Odhlášení uživatele

Klient odhlásí uživatele zprávou *logout*:

```
<logout/>
```

Server na tuto zprávu neodpovídá.

F.3 Uživatelé a uživatelské skupiny

Uživatel je osoba, které je umožněno přihlásit se k danému serveru skrze uživatelský účet. Skupina uživatelů je entita, která má schopnost sdružovat více uživatelů do jediného celku.

Uživatel je na serveru reprezentován následující množinou povinných údajů:

- *uri* – URI uživatele – jeho jednoznačný identifikátor,
- *login* – přihlašovací jméno uživatele,
- *name* – celé jméno uživatele,
- *email* – e-mail uživatele,
- *image* – URI obrázku uživatele.

Poznámka: V případě použití *OpenID* se *login* nepoužívá, a není to proto ani povinný údaj.

F.3.1 Seznamy uživatelů

Dotaz

Klient může požádat server o seznam uživatelů zasláním zprávy *getUsers*:

```
<getUsers {filtrovací atributy}/>
```

Je možné specifikovat až čtyři různé atributy, jež filtrují seznam uživatelů, o který má klient zájem:

- *uri* – URI uživatele,
- *email* – e-mailová adresa uživatele,
- *name* - celé jméno uživatele; je možné použít zástupný znak *; server zadaný výraz pokládá za prefix jména a v seznamu uživatelů ho porovnává s každým slovem ve jméně (stejně jako u entit z kontrolovaného slovníku),
- *groupUri* - URI skupiny; vyhledávají se jen uživatelé z dané skupiny.

První dva atributy by samy o sobě měly vést k seznamu o jediném uživateli.

Následující příklad je dotazem na seznam uživatelů, jejichž jméno nebo příjmení začíná na *Adam*:

```
<getUsers name="Adam"/>
```

Dále je možné použít element *includeOnly* a do něj vkládat elementy, které mají za cíl omezit odpověď jen na některé uživatelské údaje:

```
<getUsers {filtrovací atributy}>
  <includeOnly>
    {seznam projekčních elementů}
  </includeOnly>
</getUsers>
```

Makro `{seznam projekčních elementů}` je tvořeno následujícími elementy, přičemž každý může být použit maximálně jednou:

- <login/>
- <name/>
- <email/>
- <image/>
- <groups/>

Pokud není uveden element *includeOnly*, server vrací všechny informace o uživateli, které má k dispozici. Pokud je uveden, tak server vrací pouze ty údaje, které jsou explicitně uvedeny. Omezení vrácených údajů může být výhodné v případech mohutných přenosů uživatelů, kde nás zajímají jen některé informace – například v našeptávačích. Atribut *uri* vrací server vždy.

Následující příklad je požadavkem na seznam všech uživatelů. Kromě *uri* bude server vracet jméno a e-mail uživatele, žádné jiné údaje:

```
<getUsers>
  <includeOnly>
    <name/>
    <email/>
  </includeOnly>
</getUsers>
```

Odpověď

Odpověď od serveru vypadá následovně:

```
<users>
  {seznam uživatelů}
</users>
```

Makro `{seznam uživatelů}` tvoří elementy *user*:

```
<user {uživatelské údaje}/>
```

Případně:

```
<user {uživatelské údaje}>
  <groups>
    {seznam skupin ve formátu
      <group uri="{URI skupiny}"/>
    }
  </groups>
</user>
```

Příklad odpovědi na dotaz výše:

```
<users>
  <user uri="http://example.com/Annotations/users/89"
        name="Adam Nový"
        email="adam.novy@example.com"/>
  <user uri="http://example.com/Annotations/users/17"
        name="Adam Novák"
        email="adam.novak@example.com"/>
</users>
```

F.3.2 Seznamy skupin uživatelů

Dotaz

Seznam skupin může klient od serveru získat zasláním zprávy *getUserGroups*:

```
<getUserGroups {selekční atributy}
               {volitelně: withUsers="{true|false}"}/>
```

Makro `{selekční atributy}` tvoří tyto volitelné atributy:

- *uri* – URI skupiny; je-li uveden, server vrátí konkrétní skupinu,
- *name* – jméno skupiny; je možné použít zástupný znak *.

Atribut *withUsers* udává, zda se u každé skupiny má také zahrnout seznam jejích uživatelů, výchozí je *false*.

Příklad:

```
<getUserGroups uri="http://example.com/Annotations/groups/27"
               withUsers="true"/>
```

Pokud je *withUsers* nastaveno na *true*, tak jako u seznamu uživatelů je možné použít element *includeOnly*, který má shodnou syntaxi i sémantiku:

```
<getUserGroups {atributy}>
  <includeOnly>
    {seznam projekčních elementů}
  </includeOnly>
</getUserGroups>
```

Odpověď

Server odpovídá následující zprávou:

```
<userGroups>
  {seznam skupin}
</userGroups>
```

Kde jednotlivé skupiny mají formát

```
<group name="{jméno skupiny}"
       uri="{URI skupiny}"/>
```

nebo

```
<group name="{jméno skupiny}"
       uri="{URI skupiny}">
  {seznam uživatelů}
</group>
```

Příklad celé zprávy:

```
<userGroups>
  <group name="Administrators"
         uri="http://example.com/Annotations/groups/27">
    <user uri="http://example.com/Annotations/users/17"/>
    <user uri="http://example.com/Annotations/users/89"/>
  </group>
</userGroups>
```

F.3.3 Vstup uživatele do skupiny

Přihlášený uživatel může vstoupit do skupiny zasláním zprávy *joinUserGroup*:

```
<joinUserGroup uri="{URI skupiny}"/>
```

Příklad:

```
<joinUserGroup uri="http://example.com/Annotations/groups/27"/>
```

Server na tuto zprávu neodpovídá.

F.3.4 Odchod uživatele ze skupiny

Přihlášený uživatel může vystoupit ze skupiny zasláním zprávy *leaveUserGroup*:

```
<leaveUserGroup uri="{URI skupiny}"/>
```

Příklad:

```
<leaveUserGroup uri="http://example.com/Annotations/groups/27"/>
```

Server na tuto zprávu neodpovídá.

F.4 Odběry anotací

Odběr anotací je objekt, který má specifikované zdroje anotací. Zdroje anotací vybírají anotace podle jejich typu, autora a skupiny autora. Každý odběr je tak v podstatě množina pravidel pro selekci určité množiny anotací. Každý uživatel je oprávněn se k jakémukoliv odběru přihlásit a získávat tak anotace, o které má zájem, bez toho, aby tyto anotace musel ručně vytvářet. Přihlášení k odběru je platné jen v rámci sezení. Pokud klient potřebuje nastavit odběry v každém sezení, musí o ně vždy znovu žádat, jinak zůstane u výchozího nastavení, kdy odebírá anotace od svého uživatele a skupin, do kterých náleží.

F.4.1 Vytvoření odběru

Dotaz

Klient u odběru vytvoří dočasný identifikátor *tmpId*. Od serveru přijde mapování dočasného identifikátoru na trvalý identifikátor (URI). Server tedy klientovi neposílá zpátky celý odběr. Klient u odběru smaže *tmpId* a přiřadí správný URI. Zpráva pro vytvoření odběru vypadá takto:

```
<createSubscription tmpId="{dočasné id odběru}"
                    name="{jméno odběru}">
  {seznam zdrojů formátu
    <source subscribe="{true|false}"
            {selekční atributy zdroje}/>
  }
</createSubscription>
```

Zdroj (*source*) má atribut *subscribe*, který udává, zda se anotace z tohoto zdroje mají odebírat nebo ne. Lze totiž pomocí pozitivního zdroje (*subscribe="true"*) nastavit odebírání jisté množiny anotací a následně tuto množinu filtrovat dalším negativním zdrojem (*subscribe="false"*).

Makro `{selekční atributy zdroje}` tvoří volitelné atributy:

- *typeUri* – URI typu, jehož anotace se budou odebírat,
- *authorUri* – URI autora, jehož anotace se budou odebírat,
- *groupUri* – URI skupiny; budou se odebírat anotace všech jejích uživatelů.

Příklad:

```
<createSubscription tmpId="138"
                    name="My Favorite Art Movements">
  <source subscribe="true"
          typeUri="http://example.com/Annotations/types/g17/Movement"
          authorUri="http://example.com/Annotations/users/123456"
          groupUri="http://example.com/Annotations/groups/27"/>
  <source
    subscribe="false"
    typeUri="http://example.com/Annotations/types/g17/Movement/Expressionism"/>
</createSubscription>
```

Odpověď

```
<subscriptionCreated tmpId="{dočasné id odběru}"
                     uri="{URI odběru}"/>
```

Příklad:

```
<subscriptionCreated tmpId="138"
                     uri="http://example.com/Annotations/subscriptions/65"/>
```

F.4.2 Rušení odběru

Klient může zrušit (vymazat) odběr zasláním zprávy:

```
<removeSubscription uri="{URI odběru}"/>
```

Příklad:

```
<removeSubscription uri="http://example.com/Annotations/subscriptions/438"/>
```

Server na tuto zprávu neodpovídá.

F.4.3 Modifikace odběru

Klient může modifikovat odběr zasláním následující zprávy:

```
<modifySubscription uri="{URI odběru}"
                    name="{jméno odběru}">
  {seznam zdrojů formátu
    <source subscribe="{true|false}"
            {selekční atributy zdroje}/>
  }
</modifySubscription>
```

Příklad:

```
<modifySubscription uri="http://example.com/Annotations/subscriptions/71"
                    name="My Favorite Art Movements (Modified)">
  <source subscribe="true"
          typeUri="http://example.com/Annotations/types/g17/Movement"
          authorUri="http://example.com/Annotations/users/123456"
          groupUri="http://example.com/Annotations/groups/27"/>
  <source
    subscribe="false"
    typeUri="http://example.com/Annotations/types/g17/Movement/Expressionism"/>
  <source
    subscribe="false"
    typeUri="http://example.com/Annotations/types/g17/Movement/Impressionism"/>
</modifySubscription>
```

Server na tuto zprávu neodpovídá.

F.4.4 Seznamy odběrů

Dotaz

Klient může od serveru požadovat seznam odběrů filtrovaný různými atributy. Pokud žádné atributy nejsou zadány, server vrací seznam všech odběrů od všech uživatelů.

```
<getSubscriptions {filtrovací atributy}/>
```

Makro `{filtrovací atributy}` tvoří volitelné atributy:

- *subscriptionUri* – vrací se pouze odběr s daným URI,
- *authorUri* – vrací se pouze odběry, které vytvořil autor s daným URI,
- *groupUri* – URI skupiny; vrací se pouze odběry, které vytvořili uživatelé dané skupiny.

Příklad:

```
<getSubscriptions authorUri="http://example.com/Annotations/users/84"/>
```

Odpověď

```
<subscriptions>
  {seznam odběrů formátu
    <subscription uri="{URI odběru}"
                  name="{jméno odběru}"
                  authorUri="{URI autora odběru}">
      {seznam zdrojů formátu
        <source subscribe="{true|false}"
                {filtrovací atributy}/>
      }
    </subscription>
  }
</subscriptions>
```

Příklad:

```
<subscriptions>
  <subscription uri="http://example.com/Annotations/subscriptions/71"
                name="My Favorite Art Movements"
                authorUri="http://example.com/Annotations/users/8394">
    <source subscribe="true"
            typeUri="http://example.com/Annotations/types/g17/Movement"
            authorUri="http://example.com/Annotations/users/67"
            groupUri="http://example.com/Annotations/groups/27"/>
    <source
      subscribe="false"
      typeUri="http://example.com/Annotations/types/g17/Movement/Expressionism"/>
  </subscription>
</subscriptions>
```

F.4.5 Přihlášení se k odběru

Uživatel se přihlásí k již existujícímu odběru zasláním následující zprávy:

```
<subscribe subscriptionUri="{URI odběru}"/>
```

Příklad:

```
<subscribe
  subscriptionUri="http://example.com/Annotations/subscriptions/4"/>
```

Server na tuto zprávu neodpovídá.

F.4.6 Odhlášení se z odběru

Uživatel se může odhlásit z odběru, ke kterému je přihlášen, zasláním následující zprávy:

```
<unsubscribe subscriptionUri="{URI odběru}"/>
```

Příklad:

```
<unsubscribe
  subscriptionUri="http://example.com/Annotations/subscriptions/4"/>
```

Server na tuto zprávu neodpovídá.

F.5 Synchronizace dokumentu

Všechny dokumenty, které klienti otevřou v anotačním editoru, si server ukládá a klientům vrací URI těchto kopií. Všechny anotace, které dokument anotují, jako cíl anotace uvádějí právě tento URI.

F.5.1 Proces synchronizace

Klienti při procesu *synchronizace* předávají serveru URI a obsah dokumentu, který mají načtený v editoru. Server se podívá, zda už má daný URI v databázi. Pokud ne, tak jen daný dokument uloží. V případech, kdy se obsahy dokumentů liší, server může přepsat svůj dokument tím, který mu klient poslal. Pak ale také musí vyhodnotit, jak se jednotlivé změny v obsahu dotýkají anotací, které dokument anotují, a případně odpovídajícím způsobem aktualizovat jejich fragmenty. V nejhorším případě nebude možné nové fragmenty určit a novým cílem anotace se tak stane celý dokument. Uživatel pak má možnost ručně přenastavit cíle dotyčných anotací.

Dotaz

Zpráva na provedení synchronizace vypadá takto:

```
<synchronize uri="{URI načteného dokumentu}"
             {volitelné atributy}>
  <![CDATA[{obsah dokumentu}]]>
</synchronize>
```

Poznámka: Oproti verzi 1.1 chybí obalující element *content*.

Poznámka: Atribut *uri* se v protokolu 1.1 jmenuje *resource*. K přejmenování tady došlo proto, aby se snadněji rozlišovalo mezi:

- *uri* – URI dokumentu, který má klient načtený před synchronizací. Tuto hodnotu píše klient pouze do zprávy *synchronize* a
- *resource* – URI dokumentu na serveru. Jedná se o kopii původního dokumentu.

Makro `{volitelné atributy}` tvoří volitelné atributy:

- *linearized* – udává, zda klient pracuje s linearizovanou verzí dokumentu (prostý text). Pokud klient nemá WYSIWYG editor a umí pracovat pouze s nestrukturovaným textem, pak může s dokumentem pracoval v jeho linearizované formě. Výchozí hodnotou je *false*.
- *overwrite* – udává, zda serverová verze dokumentu má být nuceně přepsána tou, kterou uživatel posílá. Klient tuto hodnotu nenastavuje na *true* při počáteční synchronizaci. Nastavení atributu na *true* musí být schváleno uživatelem. Výchozí hodnotou je *false*. Pokud se serverová verze od té klientské neliší, pak na hodnotě tohoto atributu nezáleží.

Příklad synchronizační zprávy:

```
<synchronize
  uri="http://example.com/Annotations/documents/getDoc?id=1234doc1.txt"
  linearized="false"
  overwrite="false">
  <![CDATA[<html><head></head><body><p>Hello World!</p></body></html>]]>
</synchronize>
```

Odpověď

Server porovná svou verzi dokumentu s tou, kterou posílá klient.

- Pokud se shodují, synchronizace je úspěšná.
- Pokud se neshodují, server analyzuje rozdíly a zjišťuje, jak by se změnily cíle anotací dokumentu, pokud by se serverová verze dokumentu měla nahradit klientskou. Pak záleží na tom, jestli by bylo hodně zásahů do fragmentů anotací.
  - Pokud by zásahů nebylo mnoho, server nahradí svou verzi dokumentu klientskou a klientovi pošle varovací zprávu se seznamem URI všech jeho anotací, které mají změněné fragmenty (pokud takové jsou). Synchronizace je úspěšná.
  - Pokud by zásah byl veliký, server pošle klientovi chybovou zprávu, ve které uvede obsah svého dokumentu. Synchronizace je neúspěšná. Uživatel se pak rozhodne, se kterou verzí chce pracovat:
    - s klientskou verzí: Klient znovu pošle zprávu *synchronize*. Jako obsah dokumentu posílá opět svoji verzi a atribut *overwrite* nastaví na *true*. Synchronizace bude nyní úspěšná.
    - se serverovou verzí: Klient znovu pošle zprávu *synchronize*. Jako obsah dokumentu posílá verzi, kterou mu server poslal v chybové zprávě. Synchronizace bude nyní úspěšná.

Zpráva *synchronized*

Tuto zprávu posílá server při úspěšné synchronizaci.

```
<synchronized resource="{URI dokumentu na serveru}"
              lastModification="{ID poslední modifikace}"/>
```

Atribut *resource* obsahuje URI dokumentu na serveru – kopie toho dokumentu, jež klient poslal ve zprávě *synchronize*. Tento URI klient používá u anotací.

Atribut *lastModification* obsahuje identifikátor poslední modifikace provedené na dokumentu.

Příklad:

```
<synchronized
  resource="http://example.com/Annotations/documents/getDoc?id=1234"
  lastModification="72"/>
```

Spolu se zprávou *synchronized* posílá server klientovi anotace a všechny typy, které jsou potřeba pro interpretaci daných anotací.

Varovací zpráva

Pokud budou změněny fragmenty některých anotací k danému dokumentu, pak server spolu se zprávou *synchronized* pošle klientovi varovací zprávu, ve které uvede URI daných anotací:

```
<warning code="fragments updated">
  <message>
    <![CDATA[Fragmenty anotace byly aktualizovány.]]>
  </message>
  <annotations>
    {seznam anotací formátu
      <annotation uri="{URI anotace}"/>
    }
  </annotations>
</warning>
```

Příklad:

```
<warning code="fragments updated">
  <message>
    <![CDATA[Fragmenty anotace byly aktualizovány.]]>
  </message>
  <annotations>
    <annotation uri="http://example.com/Annotations/serv/3985"/>
    <annotation uri="http://example.com/Annotations/serv/1545"/>
    <annotation uri="http://example.com/Annotations/serv/148868"/>
    <annotation uri="http://example.com/Annotations/serv/88916"/>
    <annotation uri="http://example.com/Annotations/serv/35486"/>
    <annotation uri="http://example.com/Annotations/serv/99"/>
  </annotations>
</warning>
```

Chybová zpráva

Při neúspěšné synchronizaci se klientovi pošle chybová zpráva:

```
<error code="sync error different">
  <message>
    <![CDATA[Synchronizace selhala - pro danou URI je již uložen
    jiný obsah dokumentu.]]>
  </message>
  <serverVersion>
    <![CDATA[{obsah serverové verze dokumentu}]]>
  </serverVersion>
</error>
```

Příklad:

```
<error code="sync error different">
  <message>
    <![CDATA[Synchronizace selhala - pro danou URI je již uložen
    jiný obsah dokumentu.]]>
  </message>
  <serverVersion>
    <![CDATA[<html><body><p>Hello world!</p></body></html>]]>
  </serverVersion>
</error>
```

Případně může být zaslán i jiný typ chybové zprávy, který však neobsahuje obsah kopie dokumentu na serveru, neboť zde byla jiná příčina chyby (viz popis chybových zpráv níže).

F.5.2 Znovuposlání obsahu dokumentu

Zpráva od serveru

Pokud server detekuje neshodu již jednou synchronizovaných verzí dokumentu, ať již k tomuto dojde jakkoliv, server pošle klientovi příkaz, aby mu vrátil momentální obsah dokumentu. Pokud se neshoda týká požadavku klienta (např. požadavek na vytvoření anotace s chybným obsahem fragmentu), server tuto zprávu posílá přes *AJAX* kanál, jinak ji posílá přes *Comet* kanál.

```
<resynchronize resource="{URI zdroje}"
               method="{soft|hard}"/>
```

Atributy:

- *resource* – URI kopie dokumentu na serveru. Podle této hodnoty klient pozná, který dokument má serveru poslat.
- *method* – Říká klientovi, o jaký typ resynchronizace se jedná:
  - *soft*: Měkká resynchronizace
  - *hard*: Tvrdá resynchronizace

Příklad:

```
<resynchronize
  resource="http://example.com/Annotations/documents/getDoc?id=1234"
  method="soft"/>
```

Klientova reakce

- *Měkká resynchronizace*

  Klient serveru pošle pouze obsah svého dokumentu:

  ```
  <resynchronization resource="{URI zdroje}">
    <![CDATA[{obsah dokumentu}]]>
  </resynchronization>
  ```

  Příklad:

  ```
  <resynchronization
    resource="http://example.com/Annotations/documents/getDoc?id=1234">
    <![CDATA[<p>Hello World!</p>]]>
  </resynchronization>
  ```

  Server poté pošle klientovi změny v anotacích a návrzích.

- *Tvrdá resynchronizace*

  Celý proces synchronizace se musí udělat znovu, tzn. klient znovu posílá zprávu *synchronize* (popsána výše).

F.5.3 Modifikace dokumentu

Klient může být rozšířením pro WYSIWYG editor. Uživatel tedy může text libovolně měnit. Klient musí serveru posílat modifikace textu, které uživatel provedl, dostatečně často, aby docházelo k minimu konfliktů a ostatní uživatelé měli aktuální verzi dokumentu – tedy nejlépe s každou změnou v dokumentu (změna jednoho či několika znaků, případně napsané slovo).

Dotaz

```
<modification lastApplied="{číslo poslední modifikace}">
  {seznam modifikačních zpráv}
</modification>
```

Makro `{číslo poslední modifikace}` zastupuje identifikátor poslední modifikace, kterou klient provedl na svém dokumentu, než vytvořil novou změnu (viz níže).

Makro `{seznam modifikačních zpráv}` tvoří seznam modifikací. Existují 3 různé typy modifikací:

- *replace*

  Modifikace obsahu uvnitř daného uzlu DOM

```
<replace path="{XPath fragmentu}"
         {volitelně:
           offset="{offset}"
           length="{délka}"
         }>
  {volitelně
    <![CDATA[{nová data}]]>
  }
</replace>
```

  Atributy elementu *replace* jsou:

  – *path* – XPath fragmentu v rámci dokumentu,
  – *offset* – index počátku výběru v rámci uzlu,
  – *length* – délka výběru obsahu uzlu.

  Pokud se neuvedou atributy *offset* a *length*, bude se nahrazovat celý uzel.

  Poznámka: Sekce *CDATA* nemůže být uvedena, pokud by její obsah byl prázdný.

  Příklad:

```
<replace path="html[1]/body[1]/p[2]/em[3]/text()[1]"
         offset="16"
         length="4">
  <![CDATA[van Gogh]]>
</replace>
```

- *insertAfter*:

  Přidání obsahu za daný uzel DOM

  Předchozí varianta modifikace obecně neumožňuje jednoduše přidat uzel do jiného uzlu. Vytvoření nového uzlu by se muselo udělat náhradou jednoho vnitřního uzlu za dva (původní+nový). Toto řešení je nepraktické a přináší vyšší riziko chyby v konkurentnosti uživatelů modifikujících daný uzel. Druhou variantou je přidání obsahu za určitý uzel:

```
<insertAfter path="{XPath fragmentu}">
  <![CDATA[{nová data}]]>
</insertAfter>
```

Příklad:

```
<insertAfter path="html[1]/body[1]/p[2]">
  <![CDATA[<p>Nový odstavec</p>]]>
</insertAfter>
```

- *insertBefore*:

  Přidání obsahu před daný uzel DOM

  Obdoba *insertAfter*. Liší se tím, že obsah vkládá před adresovaný uzel.

```
<insertBefore path="{XPath fragmentu}">
  <![CDATA[{nová data}]]>
</insertBefore>
```

  Příklad:

```
<insertBefore path="html[1]/body[1]/p[2]/em[3]">
  <![CDATA[<br/>]]>
</insertBefore>
```

Odpověď

Aktuální verze anotovaného dokumentu je uložena na serveru. Modifikování dokumentu je omezeno výlučným přístupem. Když server obdrží nový požadavek na modifikaci ve chvíli, kdy ještě není dokončená obsluha předchozího požadavku, nový požadavek je zařazen do fronty typu FIFO, kde čeká na uvolnění kritické sekce. Server kromě obsahu aktuální verze dokumentu uchovává také čítač jeho změn a určitý počet naposledy aplikovaných změn. Když od klienta přijde požadavek na aplikaci změn, server vyhodnotí, zda je možné provést požadované změny na serverové verzi dokumentu tak, aby výsledek byl korektní z pohledu klienta.

Pokud se verze dokumentu, ze které klient vychází, shoduje s aktuální verzí, kterou má server, tak server bez dalších kontrol dané změny provede, zvýší čítač modifikací a spolu s modifikační zprávou odešle zbývajícím klientům, kteří pracují s dokumentem.

Pokud se verze dokumentů neshodují, je dalším kritériem počet požadavků na změny, které serveru poslali ostatní klienti od doby poslední změny, jakou daný klient zaregistroval. Modifikace z těchto požadavků klient neměl k dispozici a proto je ve své vlastní modifikaci nezohlednil. Pokud je těchto požadavků více než stanovená hodnota dle nastavení serveru (min. 3), server klientovi posílá chybovou zprávu se sdělením, že klientova verze dokumentu je příliš neaktualní na to, aby bylo možné na ní provádět změny.

Pokud je čekajících požadavků na změny méně než daná konstanta, server vyhodnotí, zda kterákoliv ze změn může nějak ovlivnit nové změny. Pokud ano, server posílá klientovi chybovou zprávu, ve které mu oznámí, že změny nemohly být aplikovány kvůli konfliktu s jinými změnami. V opačném případě je změny možné provést – server zvýší čítač modifikací a spolu s modifikační zprávou odešle zbývajícím klientům, kteří pracují se stejným dokumentem.

Klientovi, který o modifikaci žádá, server v případě úspěchu posílá následující zprávu:

```
<modificationApplied id="{číslo modifikace}"/>
```

Číslo modifikace se klient musí dozvědět proto, aby věděl, že toto číslo nemá očekávat u příchozích modifikací od jiných klientů (viz níže).

Ostatním klientům server distribuuje modifikační zprávy v tomto formátu:

```
<modification id="{číslo modifikace}">
  {seznam modifikačních zpráv}
</modification>
```

Také klient si vede čítač modifikací provedených na svém dokumentu. Příchozí modifikace vytvářejí frontu, ze které jsou odebrány až v momentě, kdy v čítači modifikací je hodnota o 1 menší, než číslo dané modifikace. Pak je změna provedena a čítač inkrementován.

V době, kdy klient čeká na odpověď na svůj modifikační požadavek, klient neaplikuje žádné příchozí modifikace a všechny případné modifikace provedené uživatelem vkládá do fronty. Pokud server odpovídá na požadavek modifikace chybou, klient má následující možnosti:

- zobrazit uživateli chybu a všechny serverem zamítnuté změny v dokumentu revertovat,
- provést ty modifikace z fronty příchozích modifikací, které navazují na verzi dokumentu, ze které klient vycházel při generování požadavku na změny. Při provádění modifikací je nutné odpovídajícím způsobem upravovat ty modifikace, které chce klient provést. Pokud je tato operace úspěšná, klient se může pokusit znovu vyslat požadavek na modifikace.

F.6 Typy

Anotace může nabývat pouze typu anotace (strukturovaný typ). Strukturovaný typ může obsahovat atributy, které mohou nabývat jednoduchých, ale i strukturovaných typů.

Poznámka: Pokud není uvedeno jinak, v rámci všech typů zpráv v tomto protokolu platí, že pokud se po klientovi žádá uvedení nějakého typu anotace, server uvažuje i všechny podtypy daného typu. Pokud tedy klient požádá například o návrhy anotací typu *Person*, server vrací i typy jako *Person->Employee* nebo *Person->Artist*.

F.6.1 Jednoduché typy

Jednoduché datové typy atributů v systému *4A* jsou definovány v tabulce F.2.

Název	String
URI	http://www.w3.org/2001/XMLSchema#string
Popis	ASCII řetězec
Příklad hodnoty	John Doe
Název	URI
URI	http://www.w3.org/2001/XMLSchema#anyUri
Popis	URI
Příklad hodnoty	http://example.com

Název	DateTime
URI	http://www.w3.org/2001/XMLSchema#dateTime
Popis	datum a čas dle RFC 3339 [34] (iso-date-time with datespec-full)
Příklad hodnoty	2002-10-10T12:00:00-05:00
Název	Date
URI	http://www.w3.org/2001/XMLSchema#date
Popis	datum a čas dle RFC 3339 [34] (datespec-full)
Příklad hodnoty	2002-10-10+05:00
Název	Time
URI	http://www.w3.org/2001/XMLSchema#time
Popis	čas dle RFC 3339 [34] (time)
Příklad hodnoty	13:20:00-05:00
Název	Integer
URI	http://www.w3.org/2001/XMLSchema#integer
Popis	celé číslo
Příklad hodnoty	-518
Název	Decimal
URI	http://www.w3.org/2001/XMLSchema#decimal
Popis	desetinné číslo
Příklad hodnoty	1.23
Název	Boolean
URI	http://www.w3.org/2001/XMLSchema#boolean
Popis	pravdivostní hodnota
Příklad hodnoty	true
Název	GeoPoint
URI	http://www.w3.org/2003/01/geo/wgs84_pos#Point
Popis	geografický bod dle Basic Geo (WGS84 lat/long) Vocabulary [17] (basic)

Příklad hodnoty	`<geo:Point>`
	`  <geo:lat>55.701</geo:lat>`
	`  <geo:long>12.552</geo:long>`
	`</geo:Point>`

Název	AnyAnnotation
URI	http://knot.fit.vutbr.cz/annotations/ knotOAExtension#anyAnnotation
Popis	vnořená anotace nebo odkaz na anotaci
Příklad hodnoty	nenabývá žádné hodnoty
Název	AnyEntity
URI	http://knot.fit.vutbr.cz/annotations/ knotOAExtension#anyEntity
Popis	entita
Příklad hodnoty	nenabývá žádné hodnoty
Název	Duration
URI	http://www.w3.org/2001/XMLSchema#duration
Popis	trvání dle RFC 3339 [34] (duration)
Příklad hodnoty	P1Y2MT2H

Název	Binary
URI	http://www.w3.org/2001/XMLSchema#base64binary
Popis	binární data (soubory, např. v OpenDocument formátu); data se kódují do *base64*; velikost souboru může být limitovaná serverem
Příklad hodnoty	0FB8
Název	Text
URI	http://knot.fit.vutbr.cz/annotations/knotOAExtension#text
Popis	dlouhý řetězec
Příklad hodnoty	Nějaký text
Název	Image
URI	http://knot.fit.vutbr.cz/annotations/knotOAExtension#imageUri
Popis	URI obrázku
Příklad hodnoty	http://upload.wikimedia.org/wikipedia/commons/a/a0/ Bruegge_View_from_Rozenhoedkaai.jpg
Název	Entity
URI	http://knot.fit.vutbr.cz/annotations/knotOAExtension#entity
Popis	entita z kontrolovaného slovníku
Příklad hodnoty	Serializace atributu toto typu je odlišná od serializace atributů všech ostatních jednoduchých typů – viz F.7.1.

Tabulka F.2: Jednoduché datové typy atributů

F.6.2 Formát typu anotace

```
<type name="{jméno typu}"
      uri="{URI typu}"
      groupUri="{URI skupiny}"
      restrictedAttributes="{omezení atributů}"
      {volitelně: ontologyUri="{URI z ontologie}"}>
  <directAncestors primary="{URI primárního předka}">
    {seznam přímých předků ve formátu
      <ancestor uri="{URI typu}"/>
    }
  </directAncestors>
  <attributes>
    {seznam atributů}
  </attributes>
  {volitelný komentář formátu
    <comment>
      <![CDATA[{komentář}]]>
    </comment>
  }
</type>
```

Povinné atributy elementu *type* jsou:

- *name* – jméno typu, resp. jméno uzlu v hierarchii typů anotací,

- *uri* – URI typu,
- *groupUri* – URI skupiny uživatelů, do které typ patří,
- *restrictedAttributes* – pokud je nastaveno na *true*, tak není dovoleno přidávat další atributy daného typu anotace a nelze měnit typy existujících atributů.

Atribut *ontologyUri* se uvádí v případě, že se daný typ nachází v ontologii. Hodnotou atributu je URI typu v dané ontologii.

Element *directAncestors* má atribut *primary*, který obsahuje primárního přímého předka, který, je-li prázdný, reprezentuje kořen stromu typů. Podelementy typu *ancestor* obsahují ostatní přímé předky typu.

Makro `{seznam atributů}` tvoří prvky *attribute*:

```
<attribute valueType="{simple|linked|nested}"
           name="{jméno atributu}"
           typeUri="{URI typu}"
           required="{povinnost atributu}"
           {volitelně: ontologyUri="{URI atributu z ontologie}"}/>
  {volitelně: <comment><![CDATA[{komentář atributu}]]></comment>}
</attribute>
```

Parametr *valueType* určuje typ hodnoty:

- *simple* – jednoduchý typ,
- *linked* – odkaz na anotaci,
- *nested* – vnořená anotace.

Parametr *ontologyUri* je uveden pouze v případě, že atribut patří k nějaké ontologii. V takovém případě se uvede URI atributu z ontologie.

Atributy mohou nabývat jednoduchých i strukturovaných typů.

Příklad serializovaného typu:

```
<type name="Picture"
      uri="http://example.com/Annotations/types/g17/Art/Artwork/Picture"
      groupUri="http://example.com/Annotations/groups/27"
      restrictedAttributes="true">
  <directAncestors
    primary="http://example.com/Annotations/types/g17/Art/Artwork">
    <ancestor uri="http://example.com/Annotations/types/g17/Art"/>
    <ancestor
      uri="http://example.com/Annotations/types/g17/Art/Artist/Work"/>
  </directAncestors>
  <attributes>
    <attribute name="Created"
               typeUri="http://www.w3.org/2001/XMLSchema#date"
               required="true">
      <comment>
```

```
      <![CDATA[Date of creation]]>
    </comment>
  </attribute>
  <attribute name="Price"
             typeUri="http://www.w3.org/2001/XMLSchema#integer"
             required="true">
    <comment>
      <![CDATA[Current price of the picture]]>
    </comment>
  </attribute>
</attributes>
<comment>
  <![CDATA[Picture is a kind of an artwork.]]>
</comment>
</type>
```

F.6.3 Manipulace s typy anotací

Server může ovládat množinou typů anotací, které klient obsahuje. Když klient požaduje typy, server posílá zpět příkaz na přidání typů, které mu blíže specifikuje. Když klient vytvoří nový typ, server přes *Comet* kanál posílá příkaz na přidání typů ostatním klientům, kteří pracují s daným dokumentem. Také klient může přidávat, měnit nebo odebírat typy, které si server uchovává. Komunikaci tvoří tři typy zpráv:

- *addTypes*: přidá nové typy

  ```
  <addTypes>
    {seznam typů anotací}
  </addTypes>
  ```

- *removeTypes*: odebere některé typy

  ```
  <removeTypes>
    {seznam typů anotací ve formátu
      <type uri="{URI typu}"/>
    }
  </removeTypes>
  ```

  Příklad:

  ```
  <removeTypes>
    <type uri="http://example.com/Annotations/types/g17/Art"/>
    <type uri="http://example.com/Annotations/types/g17/Person/Employee"/>
    <type uri="http://example.com/Annotations/types/g17/City"/>
  </removeTypes>
  ```

- *modifyTypes*: změní některé typy

  ```
  <modifyTypes>
    {seznam typů anotací}
  </modifyTypes>
  ```

Poznámka: Jméno typu nemůže být změněno. Pokud uživatel chce změnit název typu, musí vytvořit nový typ s novým jménem a původní typ odstranit.

F.6.4 Získání typů anotací od serveru

Dotaz

Klient nemusí vždy potřebovat všechny typy. Celkový počet typů totiž může být vysoký a mohlo by dojít k zahlcení klienta. Klient si sám žádá o typy a server mu je vrací. Při tom si server požadavky ukládá a když dojde ke změně v nějakém typu, o který klient dříve požádal (např. modifikace jiným klientem), nebo v jeho podtypu, server mu tyto změny okamžitě pošle i bez požádání.

Klient si může vyžádat typy anotací zasláním následující zprávy:

```
<getTypes {volitelně: uri="{URI typu}"}/>
```

Příklad:

```
<getTypes uri="http://example.com/Annotations/types/g17/Person/Artist"/>
```

Parametrem *uri* se vybírá typ a celý jeho podstrom. Je možné použít zástupný znak *. Pokud atribut není uveden, server vrátí všechny typy anotací ve skupinách daného uživatele.

Odpovědí od serveru je zpráva *addTypes*.

F.6.5 Získání atributů z ontologie

Když uživatel k typu anotace přidává nový atribut, musí vyplnit jeho název a zvolit typ hodnoty. Pokud byly typy anotací importovány z ontologie, budou zde atributy, které mají název a typ hodnoty, ale jejich přiřazení k typu není definované. Z tohoto důvodu jsou zde tzv. „atributy z ontologie“, které poskytují alternativní způsob volby názvu a typu hodnoty atributu. Uživatel si pak může zvolit, zda zadá název atributu a typ hodnoty, nebo zda si je zvolí ze seznamu atributů importovaných z ontologie. Klient může požádat o atributy z ontologie následující zprávou.

Dotaz

```
<getOntologyAttributes {volitelně: groupUri="{URI skupiny}"}/>
```

Atribut *groupUri* vyjadřuje URI skupiny uživatelů, na kterou se omezí seznam atributů. Pokud není uveden, server vrací atributy ze všech skupin.

Poznámka: Tento element byl přejmenován z *queryAttrFromOnto*, jak jej definoval protokol 1.1.

Příklad:

```
<getOntologyAttributes groupUri="http://example.com/Annotations/groups/27"/>
```

Odpověď

```
<ontologyAttributes>
  {seznam atributů}
</ontologyAttributes>
```

Poznámka: Tento element byl přejmenován z *attrsFromOntology*, jak jej definoval protokol 1.1.

Seznam atributů je tvořen elementy typu *attribute*:

```
<attribute name="{jméno atributu}"
           uri="{URI atributu}"
           typeUri="{URI typu}"
           groupUri="{URI skupiny}">
  {volitelně: <comment><![CDATA[{komentář atributu}]]></comment>}
</attribute>
```

F.7 Anotace

Anotace je strukturovaný komentář k úseku textu v dokumentu. Má typ a atributy. Atributy mohou obsahovat vnořené anotace či odkazy na další anotace a vytvářet tak komplexnější hierarchii.

I když je tento protokol navržen obecně, pro sjednocení dokumentace a lepší vysvětlení správného využití je zde popsán i zvolený konkrétní formát anotace, který lze využít. Tímto formátem je Open Annotation [78]. Využití jiného formátu by bylo analogické.

F.7.1 Formát anotace

Anotace se serializují dle specifikace *Open Annotation Data Model* [78].

```
<oa:Annotation rdf:about="{URI anotace}">
  <oa:hasBody>
    <oa:SemanticTag rdf:about="{URI typu anotace}"/>
  </oa:hasBody>
  <oa:hasBody>
    <cnt:ContentAsText rdf:about="{URI anotace}#body">
      <rdf:type rdf:resource="http://purl.org/dc/dcmitype/Text"/>
      <cnt:chars>
        <![CDATA[{obsah anotace}]]>
      </cnt:chars>
      <dc:format>text/plain</dc:format>
    </cnt:ContentAsText>
  </oa:hasBody>
    {TARGET}
  <oa:hasBody>
    <cnt:ContentAsText rdf:about="{URI dokumentu}">
      <rdf:type rdf:resource="http://www.w3.org/2004/03/trix/rdfg-1/Graph"/>
      <trix:TriX>
        <trix:graph>
        {SEZNAM ATRIBUTŮ ANOTACE}
        </trix:graph>
      </trix:TriX>
      <dc:format>text/xml</dc:format>
    </cnt:ContentAsText>
```

```
  </oa:hasBody>
  <oa:annotatedBy>
    <foaf:Person rdf:about="{URI autora}">
      <foaf:name>{jméno autora}</foaf:name>
      <foaf:mbox>mailto:{e-mailová adresa autora}</foaf:mbox>
    </foaf:Person>
  </oa:annotatedBy>
  <oa:annotatedAt>{čas vytvoření anotace}</oa:annotatedAt>
  <oa:serializedAt>{čas vytvoření anotace}</oa:serializedAt>
</oa:Annotation>
```

Poznámka: elementy *oa:annotatedAt* a *oa:serializedAt* obvykle nesou stejný časový údaj, neboť čas serializace lze zanedbat.

Popis maker

- *{URI anotace}*: `{URI serveru}/Annotations/{prefix}/{id anotace}`
  příklad: `http://example.com/Annotations/serv/12356`
- *{URI typu anotace}*: `{URI serveru}/Annotations/types/g{číslo skupiny}/annotation/{serializovaný název typu}`
  příklad: `http://example.com/Annotations/types/g01/annotation/person`
- *{obsah anotace}*: jakýkoliv řetězec ASCII
  příklad: `Tohle je anotace.`
- *{URI dokumentu}*: `{URI serveru}/Annotations/documents/getDoc?id={číslo dokumentu}`
  příklad: `http://example.com/Annotations/documents/getDoc?id=123456`
- *{xpointer}*: `xpointer(string-range({XPath fragmentu}, '{anotovaný text}', {offset}, {délka}))`
  příklad: `xpointer(string-range(/html/body/div/p[1]/text(), 'anotovaný text' , 5, 14))`
- *{seznam fragmentů}*: konkatenace `#{xpointer}`
  příklad:

```
#xpointer(string-range(
  /html/body/div/p[1]/text()[1], 'anotovaný text', 5, 14))
#xpointer(string-range(
  /html/body/div/p[1]/text()[2], 'anotovaný text', 24, 14))
#xpointer(string-range(
  /html/body/div/p[1]/text()[3], 'anotovaný text' , 86, 14))
```

- *{URI autora}*: `{URI serveru}/Annotations/users/{číslo uživatele}`
  příklad: `http://example.com/Annotations/users/123456`
- *{jméno autora}*: jméno autora anotace
  příklad: `Jaroslav Dytrych`

- *{e-mailová adresa autora}*: e-mailová adresa autora anotace
  příklad: `dytrych@fit.vutbr.cz`
- *{čas vytvoření anotace}*: časový okamžit vytvoření anotace
  příklad: `2013-01-28T12:00:00Z`
- *{jméno atributu nebo jeho URI z ontologie}*
  příklad: `http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.2_english_label.rdfs#name`
- *{URI typu}*
  příklad: `http://www.w3.org/2001/XMLSchema#string`
- *{TARGET}*: Anotovat je možné tři typy cílů:
  - Celý dokument:

```
<oa:hasTarget>
  <dctypes:Text rdf:about="{URI dokumentu}">
    <dc:format>text/xml</dc:format>
  </dctypes:Text>
</oa:hasTarget>
```

  - Jeden fragment anotace:

```
<oa:hasTarget>
  <oa:SpecificResource rdf:about="{URI dokumentu}#{xpointer}">
    <oa:hasSelector>
      <oa:FragmentSelector
        rdf:about="{URI dokumentu}#{xpointer}#selector1">
        <dcterms:conformsTo
          rdf:resource="http://tools.ietf.org/rfc/rfc3023"/>
        <rdf:value>{xpointer}</rdf:value>
      </oa:FragmentSelector>
    </oa:hasSelector>
    <oa:hasSource>
      <dctypes:Text rdf:about="{URI dokumentu}">
        <dc:format>text/xml</dc:format>
      </dctypes:Text>
    </oa:hasSource>
  </oa:SpecificResource>
</oa:hasTarget>
```

  - Více fragmentů anotace:

```
<oa:hasTarget>
  <oa:Composite rdf:about="{URI dokumentu}{seznam fragmentů}">
  {SEZNAM FRAGMENTŮ ANOTACE TVARU
    <oa:item>
      <oa:SpecificResource rdf:about="{URI dokumentu}#{xpointer}">
        <oa:hasSelector>
          <oa:FragmentSelector
```

```
            rdf:about="{URI dokumentu}#{xpointer}#selector">
            <dcterms:conformsTo
              rdf:resource="http://tools.ietf.org/rfc/rfc3023"/>
            <rdf:value>{xpointer}</rdf:value>
          </oa:FragmentSelector>
        </oa:hasSelector>
        <oa:hasSource>
          <dctypes:Text rdf:about="{URI dokumentu}">
            <dc:format>text/xml</dc:format>
          </dctypes:Text>
        </oa:hasSource>
      </oa:SpecificResource>
    </oa:item>
  }
  </oa:Composite>
</oa:hasTarget>
```

- Makro `{SEZNAM ATRIBUTŮ ANOTACE}` tvoří atributy formátu:
  - *Atribut je typu entity*:
    - V případě, že atribut nemá hodnotu, je serializován následovně:

      ```
      <trix:triple>
        <trix:uri>{URI cíle}</trix:uri>
        {označení atributu}
        <trix:uri>{URI typu atributu}</trix:uri>
      </trix:triple>
      ```

      `{URI typu atributu}` je v tomto případě `http://knot.fit.vutbr.cz/annotations/knotOAExtension#anyEntity`.
    - V případě, že má atribut alespoň jednu hodnotu, jsou všechny jeho hodnoty serializovány následovně:

      ```
      <trix:triple>
        <trix:uri>{URI cíle}</trix:uri>
        {označení atributu}
        <trix:uri>{URI entity}</trix:uri>
      </trix:triple>
      <trix:triple>
        <trix:uri>{URI entity}</trix:uri>
        <trix:uri>rdf:type</trix:uri>
        <trix:name>{název typu entity}</trix:name>
      </trix:triple>
      {pro všechny ostatní atributy dané entity:
        <trix:triple>
          <trix:uri>{URI entity}</trix:uri>
          <trix:name>{jméno atributu entity}</trix:name>
          <trix:typedLiteral datatype="{URI typu hodnoty atributu}">
            {hodnota atributu}
          </trix:typedLiteral>
        </trix:triple>}
      ```

– *Atribut je jednoduchého typu, ale přitom ne entity*:
Atribut je tvořen konkatenací serializací všech jeho hodnot. Každá hodnota má následující formát:

```
<trix:triple>
  <trix:uri>{URI cíle}</trix:uri>
  {označení atributu}
  <trix:typedLiteral datatype="{URI typu atributu}">
    {hodnota atributu}
  </trix:typedLiteral>
</trix:triple>
```

`hodnota atributu` je serializována podle tabulky jednoduchých typů v kapitole F.6.1.

Poznámka: V případě, že atribut nemá žádnou hodnotu, je serializován, jako by měl právě jednu hodnotu, přičemž `{hodnota atributu}` tvoří prázdný řetězec.

– *Atribut je složeného typu.* Serializace se liší podle toho, jestli je znám typ atributu, tedy typ anotací, které je možné vybírat jako hodnoty daného atributu:

  * Typ vybíratelných anotací není znám – použije se pseudotyp *anyAnnotation*:

    ```
    <trix:triple>
      <trix:uri>{URI cíle}</trix:uri>
      {označení atributu}
      <trix:typedLiteral datatype="{URI typu atributu}"/>
    </trix:triple>
    ```

    `{URI typu atributu}` je v tomto případě `http://knot.fit.vutbr.cz/annotations/knotOAExtension#anyAnnotation`.
    Poznámka: Typ *anyAnnotation* nemůže nabývat žádné hodnoty.

  * Typ vybíratelných anotací je znám. Ten je nejdřív nutné specifikovat:

    ```
    <trix:triple>
      <trix:uri>{URI cíle}</trix:uri>
      {označení atributu}
      <trix:uri>{URI typu atributu}</trix:uri>
    </trix:triple>
    ```

  Dále je potřeba definovat, zda anotace jakožto hodnoty atributu mají být vnořené nebo odkazované:

```
<trix:triple>
  <trix:uri>{URI cíle}</trix:uri>
  {označení atributu}
  <trix:uri>{koae:linkedAnnotation|koae:nestedAnnotation}</trix:uri>
</trix:triple>
```

Jednotlivé hodnoty atributu se poté serializují v závislosti na tom, zda jde o odkazované nebo vnořené anotace.

* Odkazovaná anotace:

```
<trix:triple>
  <trix:uri>{URI cíle}</trix:uri>
  {označení atributu}
  <trix:uri>{URI odkazované anotace}</trix:uri>
</trix:triple>
```

* Vnořená anotace:

```
<trix:triple>
  <trix:uri>{URI cíle}</trix:uri>
  {označení atributu}
  <trix:uri>{URI vnořené anotace}</trix:uri>
</trix:triple>
<trix:triple>
  <trix:uri>{URI vnořené anotace}</trix:uri>
  <trix:uri>koae:nestedIn</trix:uri>
  <trix:uri>{URI rodičovské anotace}</trix:uri>
</trix:triple>
```

Makro `{URI cíle}` tvoří `{URI dokumentu}{seznam fragmentů}`.

Makro `{označení atributu}` tvoří 2 možnosti:

* Atribut je obsažen v ontologii, označuje se pomocí URI:

```
<trix:uri>{URI atributu z ontologie}</trix:uri>
```

* Atribut není obsažen v ontologii, označuje se jménem:

```
<trix:name>{jméno atributu}</trix:name>
```

F.7.2 Manipulace s anotacemi klienta

Server může ovlivňovat množinou anotací, které má klient v paměti. Využívá k tomu následující zprávy:

- *addAnnotations*: přidá klientovi nové anotace

```
<addAnnotations>
  {seznam anotací}
</addAnnotations>
```

Makro `{seznam anotací}` tvoří prvky *oa:Annotation*.

- *removeAnnotations*: odebere klientovi některé anotace

```
<removeAnnotations>
  {seznam anotací ve formátu
    <annotation uri="{URI anotace}"/>
  }
</removeAnnotations>
```

Příklad:

```
<removeAnnotations>
  <annotation uri="http://example.com/Annotations/serv/5555"/>
  <annotation uri="http://example.com/Annotations/serv/4444"/>
</removeAnnotations>
```

- *modifyAnnotations*: změní některé klientovy anotace

```
<modifyAnnotations>
  {seznam anotací}
</modifyAnnotations>
```

Makro `{seznam anotací}` tvoří prvky *oa:Annotation*.

F.7.3 Vytvoření anotace klientem

Dotaz

Klient může poslat serveru nově vytvořené anotace zasláním této zprávy:

```
<createAnnotations>
  {seznam anotací}
</createAnnotations>
```

Makro `{seznam anotací}` tvoří prvky *oa:Annotation*. Trvalé URI (prefix *serv*) musí vytvořit server. Klient vygeneruje dočasnou URI s prefixem *temp*.

Odpověď

```
<annotationsCreated>
  {seznam anotací ve formátu
    <annotation tempUri="{dočasné URI anotace}"
                servUri="{trvalé URI anotace}"/>
  }
</annotationsCreated>
```

Příklad:

```
<annotationsCreated>
  <annotation tempUri="http://example.com/Annotations/temp/268"
              servUri="http://example.com/Annotations/serv/13"/>
  <annotation tempUri="http://example.com/Annotations/temp/42"
              servUri="http://example.com/Annotations/serv/711"/>
</annotationsCreated>
```

Jde o mapování URI s prefixem *temp* na URI s prefixem *serv*. Klient musí všechny výskyty URI anotace přepsat trvalým URI.

F.7.4 Modifikace a rušení anotace klientem

Klient může modifikovat a rušit anotace zprávami *modifyAnnotations* a *removeAnnotations*. Tyto zprávy jsou shodné s těmi, které posílá server klientovi. Server na ně neodpovídá.

F.7.5 Znovuzaslání anotací

Klient může požádat o znovuzaslání jedné anotace nebo všech anotací, které jsou v dokumentech, jež má klient otevřené.

```
<reloadAnnotation {volitelně: uri="{URI anotace}}"/>
```

Příklad:

```
<reloadAnnotation uri="http://example.com/Annotations/serv/5556"/>
```

Atribut *uri* udává URI anotace, která má být znovu poslána. Pokud není tento atribut uveden, server znovu pošle všechny anotace k otevřeným dokumentům. Odpovědí serveru je zpráva *addAnnotations*.

F.8 Návrhy anotací

Návrh anotace je anotace, kterou vytvořily automatické nástroje pro extrakci informací z textu. Vzhledem k tomu, že výstup z těchto nástrojů může obsahovat řadu chyb, je nutné, aby tyto anotace uživatel zkontroloval a potvrdil jejich správnost (nebo je odmítl jako nesprávné). Návrh anotace se liší od anotace prefixem URI *sugg*.

F.8.1 Manipulace s návrhy

Server může ovládat množinu návrhů anotací, které má klient v paměti. Může tak činit těmito dvěma zprávami:

- *addSuggestions*: přidá klientovi nové návrhy

  ```
  <addSuggestions>
    {seznam návrhů anotací ve formátu
      <suggestion confidence="{důvěryhodnost}">
        {anotace}
      </suggestion>
    }
  </addSuggestions>
  ```

  Atribut *confidence* udává úroveň důvěryhodnosti správného rozpoznání anotace. Vyjadřuje se v procentech.

  Makro `{anotace}` je formátu *oa:Annotation*.

- *removeSuggestions*: odebere klientovi některé návrhy

```
<removeSuggestions>
  {seznam návrhů anotací ve formátu
    <suggestion uri="{URI návrhu anotace}"/>
  }
</removeSuggestions>
```

Příklad zprávy:

```
<removeSuggestions>
  <suggestion uri="http://example.com/Annotations/sugg/458784"/>
  <suggestion uri="http://example.com/Annotations/sugg/547"/>
  <suggestion uri="http://example.com/Annotations/sugg/35"/>
  <suggestion uri="http://example.com/Annotations/sugg/42"/>
  <suggestion uri="http://example.com/Annotations/sugg/68586"/>
</removeSuggestions>
```

F.8.2 Získání návrhů

Klient může požádat o návrhy anotací zasláním následující zprávy:

```
<suggestAnnotations minConfidence="{minimální důvěryhodnost}"
                    {volitelně: autoConfirm="{automatické potvrzení}"}>
  {volitelně:
    <types>
      {seznam typů ve formátu
        <type uri="{URI typu anotace}"/>
      }
    </types>
  }
  {volitelně:
    <fragments>
      {seznam fragmentů ve formátu
        <fragment path="{XPath fragmentu}"
                  {volitelně:
                    offset="{offset}"
                    length="{délka}"
                  }/>
      }
    </fragments>
  }
</suggestAnnotations>
```

Atributy elementu *suggestAnnotations* jsou:

- *minConfidence* – minimální úroveň *confidence*, se kterou se návrh zahrnuje do odpovědi. Návrhy s nižší úrovní se ignorují,
- *autoConfirm* – maximální úroveň *confidence*, se kterou se návrh zahrnuje do odpovědi. Návrhy s vyšší úrovní server přímo potvrdí a klientovi vrátí jako anotace, které může uživatel upravit či vymazat. Pokud tento atribut není uveden, server žádné návrhy nepotvrzuje.

Pokud není uveden element *types*, navrhují se anotace všech typů. Pokud uveden je, navrhují se pouze anotace typů daných elementy *type*.

Pokud není uveden element *fragments*, návrhy budou vráceny pro celý dokument. Pokud uveden je, server vrátí pouze návrhy pro fragmenty, které jsou specifikovány elementy *fragment*.

Atributy elementu *fragment* jsou:

- *path* – XPath uzlu, ke kterému se budou navrhovat anotace,
- *offset* – pozice v uzlu, od které se budou navrhovat anotace,
- *length* – délka obsahu, pro který se mají anotace navrhovat.

Pokud ve fragmentu nejsou uvedeny atributy *offset* a *length*, server vrátí návrhy anotací pro celý uzel.

Příklad:

```
<suggestAnnotations>
  <types>
    <type uri="http://example.com/Annotations/types/g17/City"/>
    <type uri="http://example.com/Annotations/types/g17/Person"/>
  </types>
  <fragments>
    <fragment path="html[1]/body[1]/p[3]"
              offset="268"
              length="354"/>
    <fragment path="html[1]/body[1]/p[4]"/>
  </fragments>
</suggestAnnotations>
```

Odpovědí od serveru je zpráva *addSuggestions*.

F.8.3 Potvrzení návrhu

Potvrzení návrhu anotace je proces, který iniciuje uživatel. Cílem operace je vytvořit běžnou anotaci z návrhu anotace, který uživatel potvrzuje.

Poznámka: Při potvrzování návrhů si musí klient být vědom toho, že má-li návrh vnořené nebo odkazované návrhy anotací, pak je potřeba také potvrdit i všechny tyto návrhy. Pro ně platí stejná pravidla, což ve výsledku znamená, že uživatel může potvrzením jedné anotace potvrdit složitější hierarchii návrhů.

Dotaz

Klient může potvrdit návrhy zasláním následující zprávy:

```
<confirmSuggestions>
  {seznam návrhů anotací ve formátu
    <suggestion modified="false"
                uri="{URI návrhu anotace}"/>
  }
</confirmSuggestions>
```

Případně:

```
<confirmSuggestions>
  {seznam návrhů anotací ve formátu
    <suggestion modified="true">
      {anotace ve formátu oa:Annotation}
    </suggestion>
  }
</confirmSuggestions>
```

Atribut *modified* serveru říká, jestli byl návrh anotace před potvrzením upraven. Možné hodnoty jsou:

- *true* – návrh byl upraven, do elementu *suggestion* se musí přidat anotace ve formátu *oa:Annotation*. Naopak se neuvádí URI návrhu, protože to je součástí anotace.
- *false* – návrh nebyl upraven, součástí elementu *suggestion* musí být atribut *uri*, který udává URI návrhu.

Příklad celé zprávy:

```
<confirmSuggestions>
  <suggestion uri="http://example.com/Annotations/sugg/111684"
              modified="false"/>
  <suggestion uri="http://example.com/Annotations/sugg/58568"
              modified="false"/>
</confirmSuggestions>
```

Odpověď

Server vrátí mapování *suggUri* na *servUri*:

```
<suggestionsConfirmed>
  {seznam mapování URI návrhů ve formátu
    <suggestion suggUri="{URI návrhu anotace}"
                servUri="{URI anotace}"/>
  }
<suggestionsConfirmed>
```

Příklad:

```
<suggestionsConfirmed>
  <suggestion suggUri="http://example.com/Annotations/sugg/111684"
              servUri="http://example.com/Annotations/serv/5557"/>
  <suggestion suggUri="http://example.com/Annotations/sugg/481"
              servUri="http://example.com/Annotations/serv/1"/>
</suggestionsConfirmed>
```

Klient pak musí nahradit všechny výskyty *suggUri* za *servUri* a transformovat návrhy na anotace.

F.8.4 Odmítnutí návrhu

Klient může odmítnout návrh zasláním následující zprávy:

```
<refuseSuggestions>
  {seznam URI odmítnutých návrhů anotací ve formátu
    <suggestion uri="{URI návrhu anotace}"/>
  }
</refuseSuggestions>
```

Příklad zprávy:

```
<refuseSuggestions>
  <suggestion uri="http://example.com/Annotations/sugg/111684"/>
  <suggestion uri="http://example.com/Annotations/sugg/1878"/>
</refuseSuggestions>
```

Na tuto zprávu server neodpovídá.

F.9 Kontrolovaný slovník

Kontrolovaný slovník je databáze entit, které obvykle představují reálně existující objekty (osoby, místa, organizace apod.), ale mohou představovat i jiné prvky, které chceme při anotování jednoznačně identifikovat. Tyto entity pak lze používat jako hodnoty atributů anotací. Entity mají své typy, které jsou dané znalostní bází – například:

- person,
- artist,
- location,
- artwork,
- museum,
- event,
- visual art form,

- visual art medium,
- visual art movement,
- visual art genre,
- nationality.

F.9.1 Získání typů entit

Klient může získat seznam typů entit následujícím požadavkem:

Dotaz

```
<getEntityTypes/>
```

Odpověď

```
<entityTypes>
  {seznam typů ve formátu
    <type name="{název typu}"
          description="{popis typu}"/>
  }
</entityTypes>
```

Příklad odpovědi:

```
<entityTypes>
  <type name="artwork"
        description="A work of an artist, considered to has both spiritual
          and monetary value."/>
</entityTypes>
```

F.9.2 Získání entit

Dotaz

Klient může požádat server o entity ze slovníku. Komunikace v tomto případě probíhá asynchronně. Požadavek vypadá následovně:

```
<getEntities name="{jméno entity}"
             {volitelně: type="{typ entity}"}
             {volitelně: maxResults="{maximální počet výsledků}"}/>
```

Atributy elementu *getEntities* jsou:

- *name* – jméno entity; serveru postačuje prefix libovolného slova ve jméně,
- *type* – typ entity; pokud není zadáno, server vrací entity všech typů,
- *maxResults* – maximální počet vyhledaných entit; není-li zadáno, server vyhledá všechny entity.

Poznámka: Neuvedení limitu počtu vyhledávaných entit nebo jeho vysoká hodnota může způsobit výkonnostní problémy. Server proto může aplikovat vlastní limit na počet vrácených entit.

Příklad požadavku:

```
<getEntities type="artwork"
             name="Sunflowers"/>
```

Odpověď

```
<entities name="{jméno entity z~požadavku}"
          {volitelně: type="{typ entity z~požadavku}"}>
  {seznam entit formátu
    <entity name="{jméno entity}"
            uri="{URI entity}"
            type="{typ entity}"
            image="{URI obrázku entity}">
      {volitelně:
        <description>
          <![CDATA[{popis entity}]]>
        </description>
      }
      {další volitelné atributy entity}
    </entity>
  }
</entities>
```

Atribut *type* elementu *entities* udává server v odpovedi pouze v případě, že ho klient uvedl v dotazu.

Příklad odpovědi:

```
<entities type="artwork"
          name="Sunflowers">
  <entity name="Four Cut Sunflowers"
          uri="http://www.freebase.com/m/04d7gfr"
          type="artwork"
          image="http://athena3.fit.vutbr.cz/kb/images/freebase/04d8b77.jpg">
    <description>
      <![CDATA[Four Cut Sunflowers (Aug.-Sept. 1887) is one of a~series of
        sunflower-themed paintings by Dutch painter Vincent van Gogh. The
        painting is currently owned by the Krller-Mller Museum in Otterlo.]]>
    </description>
    <artist>Vincent van Gogh</artist>
    <subject>Sunflower</subject>
    <location>Kröller-Müller Museum</location>
    <owner>Kröller-Müller Museum</owner>
  </entity>
```

```
  <entity name="Vase with Five Sunflowers"
          uri="http://www.freebase.com/m/044dnb2"
          type="artwork"
          image="http://athena3.fit.vutbr.cz/kb/images/freebase/044dnjv.jpg"/>
</entities>
```

F.10 Nastavení

Nastavení je množina parametrů konfigurace klienta nebo serveru. Každý uživatel má na serveru uloženou vlastní sadu parametrů, které může skrze klienta číst a měnit dle potřeby.

F.10.1 Přenos nastavení ze serveru

Pro přenos nastavení ze serveru na klienta se používá zpráva *settings*:

```
<settings>
  {seznam parametrů ve formátu
    <param name="{jméno parametru}"
           value="{hodnota parametru}"
           {volitelně: description="{popis parametru}"}/>
  }
</settings>
```

Jméno parametru by mělo mít následující formát:

- `Server{vlastní jméno parametru}` pro parametr nastavení serveru,
- `Client{vlastní jméno parametru}` pro parametr nastavení klienta společný pro všechny implementace klientů,
- `Client{jméno klientské aplikace}{vlastní jméno parametru}` pro parametr nastavení klienta určený pro konkrétní implementaci klienta.

Popis parametru má význam pouze u nestandardních parametrů, i když jeho využití není nijak omezováno.

Příklad zprávy:

```
<settings>
  <param name="ClientAnnotationTypeColor:Animal->Dog"
         value="green"/>
  <param name="ClientAnnotationTypeColor:City"
         value="blue"/>
</settings>
```

Server tuto zprávu posílá při přihlášení uživatele (spolu se zprávou *logged*) a při jakékoliv změně v seznamu parametrů. Klient na tuto zprávu reaguje tak, že zahodí dříve získaný seznam parametrů a nahradí ho novým, včetně všech úkonů s tím souvisejících.

F.10.2 Změna parametrů

Klient může měnit seznam parametrů s nastavením. Do následující zprávy umisťuje pouze ty parametry, které chce změnit:

```
<updateParameters>
  {seznam parametrů ve formátu
    <param name="{jméno parametru}"
           {volitelně: value="{hodnota parametru}"}
           {volitelně: description="{popis parametru}"}/>
  }
</updateParameters>
```

Poznámka: Pokud chce klient odebrat hodnotu parametru, nastaví atribut *value* na prázdný řetězec. Pokud chce celý parametr odstranit, atribut *value* neuvádí.

Příklad:

```
<updateParameters>
  <param name="ClientAnnotationTypeColor:Animal->Dog"
         value="red"/>
  <param name="ClientAnnotationTypeColor:City"
         value="silver"/>
</updateParameters>
```

Server na tuto zprávu neodpovídá, a to ani zprávou *settings*.

F.11 Chyby a varování

F.11.1 Chybová zpráva

```
<error code="{kód chyby}">
  <message>
    <![CDATA[{popis chyby}]]>
  </message>
  {volitelný kontext chyby}
</error>
```

Atribut *code* označuje kód chyby.

Makro `{popis chyby}` tvoří detailnější popis chyby, který může klient zobrazit uživateli. Jazyk zprávy je závislý na parametru *ServerLanguage* v nastavení. Hodnoty tohoto parametru odpovídají ISO 639-2 [51] (pro bibliografické účely). Výchozí hodnotou je *eng* (angličtina).

Makro `{volitelný kontext chyby}` tvoří volitelné elementy, které blíže specifikují vzniklou chybu.

Příklad zprávy:

```
<error code="permission denied">
  <message>
    <![CDATA[Nemáte oprávnění ke zvolené anotaci.]]>
  </message>
  <annotation uri="http://example.com/Annotations/serv/482"/>
</error>
```

F.11.2 Varovací zpráva

Varovací zpráva má shodný formát s chybovou zprávou, jen se element zprávy jmenuje *warning*:

```
<warning code="{kód varování}">
  <message>
    <![CDATA[{popis varování}]]>
  </message>
  {volitelný kontext varování}
</warning>
```

Příklad zprávy:

```
<warning code="fragments updated">
  <message>
    <![CDATA[Fragmenty anotace byly aktualizovány.]]>
  </message>
  <annotations>
    <annotation uri="http://example.com/Annotations/serv/17"/>
    <annotation uri="http://example.com/Annotations/serv/232"/>
    <annotation uri="http://example.com/Annotations/serv/88"/>
  </annotations>
</warning>
```

F.11.3 Seznam chybových kódů

- „`0`“ – Nepodporovaná verze protokolu.
  - Chybějící, špatná nebo nepodporovaná verze protokolu ve zprávě *connect*.
  - Musí mít kód „0“ pro dosažení zpětné kompatibility (v tomto případě server nemůže rozpoznat správnou verzi protokolu).
- „`bad credentials`“ – Chybné přihlašovací jméno nebo heslo.
  - Neznámý uživatel nebo chybějící či špatné heslo nebo token ve zprávě *login*.
- „`permission denied`“ – Nemáte oprávnění ke zvolené anotaci.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`read only`“ – Přístup pouze pro čtení – anotace nebyla uložena.
  - Daný uživatel nemá povoleno anotovat tento dokument.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`change not permitted`“ – Editace není povolena.
  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

  nebo kontext:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"/>
  ```

- „`removing not permitted`“ – Mazání není povolené.
  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

  nebo kontext:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"/>
  ```

- „`attribute required`“ – Chybí povinné atributy.
  - Povinný atribut nebyl vyplněn.

  Kontext:

  ```
  <attribute annotationUri="{URI anotace}"
             uri="{URI atributu}" name="{název atributu}"
             valueType="{typ hodnoty}" type="{typ atributu}"/>
  ```

- „`attribute value`“ – Chybná hodnota atributu.
  - Špatný formát hodnoty jednoduchého typu atributu.
  - Neznámá vnořená či odkazovaná anotace (v odkazu byl využit (dočasný) identifikátor něčeho neznámého).
  - Chybí typ vnořené anotace.

  Kontext:

  ```
  <attribute annotationUri="{URI anotace}"
             uri="{URI atributu}" name="{název atributu}"
             valueType="{typ hodnoty}" type="{typ atributu}"/>
  ```

`{označení atributu}` má 2 varianty, proto jsou zde 2 atributy `uri` a `name` z nichž bude vždy využit pouze jeden.

- „`sug fragment`“ – Chybná volba fragmentu – nabízení není možné.
  - Problém se specifikací fragmentu pro návrhy anotací (např. když daný fragment právě vymazal jiný uživatel).
- „`sync error different`“ – Synchronizace selhala - pro danou URI je již uložen jiný obsah dokumentu.
  - Byly zjištěny zásadní změny oproti uložené verzi dokumentu, což při synchronizaci způsobí zásadní změny v uložených anotacích.

  Kontext:

  ```
  <content><![CDATA[Obsah kopie dokumentu na serveru.]]></content>
  ```

- „`sync error not possible`“ – Synchronizace není možná.
  - Neznámá chyba synchronizace (např. když server nepodporuje danou metodu (např. linearizovaná forma)).
- „`sync error need resync`“ – Chyba synchronizace.
  - Server při nějakém procesu detekoval možnou chybu synchronizace (např. když fragmenty právě vytvořené anotace neodpovídají dokumentu – jsou posunuté).
- „`type add not permitted`“ – Přidávání typů anotací není povolené.
  - Daný uživatel nemá povolené přidávání nových typů anotací.
- „`attribute type unavailable`“ – Typ atributu neexistuje.
  - Typ atributu anotace neexistuje.

  Kontext:

  ```
  <attribute annotationUri="{URI anotace}"
             uri="{URI atributu}" name="{název atributu}"
             valueType="{typ hodnoty}" type="{typ atributu}"/>
  ```

- „`type malformed`“ – Typ anotace je chybný.
  - Některá část specifikace typu anotace je chybná (název je prázdný apod.)

  Kontext:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"
        invalidAttribute="{název chybného atributu}"/>
  ```

Příklad kontextu:

```
<type name="Picture"
      uri="http://example.com/Annotations/types/g17/Art/Artwork/Picture"
      invalidAttribute="groupUri"/>
```

`{název chybného atributu}` může být `name`, `uri`, `groupUri`, `restrictedAttributes`, `ontologyUri`, `directAncestors` nebo `comment`

- „`type attributes malformed`“ – Atributy typu anotace jsou chybné.
  - Některá část specifikace atributu typu anotace je chybná (chybí typ apod.)

  Kontext:

  ```
  <attribute typeName="{název typu anotace}"
             typeUri="{URI typu anotace}" name="{název atributu}"
             ontologyUri="{URI atributu v ontologii}"
             invalidAttribute="{název chybného atributu}"/>
  ```

  Příklad kontextu:

  ```
  <attribute typeName="Picture"
     typeUri="http://example.com/Annotations/types/g17/Art/Artwork/Picture"
             name="E67_Birth"
             ontologyUri="http://www.cidoc-crm.org/cidoc-crm/E67_Birth"
             invalidAttribute="required"/>
  ```

  `{název chybného atributu}` může být `valueType`, `name`, `typeUri`, `required`, `ontologyUri`, `priority` nebo `comment`

- „`type unknown`“ – Typ anotace neexistuje.
  - Typ anotace nebyl nalezen.

  Kontext v případě, že typ má být modifikován:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"/>
  ```

  Kontext v případě, že typ má být vymazán:

  ```
  <type uri="{URI typu anotace}"/>
  ```

  Kontext v případě, že byl typ využit jako typ anotace:

  ```
  <annotation uri="{URI anotace}"/>
  ```

Kontext v případě, že byl typ využit jako typ atributu anotace:

```
<attribute annotationUri="{URI anotace}"
           uri="{URI atributu}" name="{název atributu}"
           valueType="{typ hodnoty}" type="{typ atributu}"/>
```

Kontext v případě, že byl typ využit jako typ atributu typu anotace:

```
<attribute typeName="{název typu anotace}"
           typeUri="{URI typu anotace}" name="{název atributu}"
           uri="{URI atributu}" valueType="{typ hodnoty}"
           type="{typ atributu}"/>
```

Kontext v případě, že byl typ využit jako předek typu anotace:

```
<type name="{název typu anotace}"
      uri="{URI typu anotace}" ancestor="{URI typu předka}"/>
```

- „`type name modify`“ – Změna názvu, předka či skupiny typu anotace není možná.
  - Pokus o modifikaci názvu, předka nebo skupiny typu anotace.

  Kontext:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"
        invalidAttribute="{název chybného atributu}"/>
  ```

- „`settings malformed`“ – Chyba v nastavení.
  - Chybný parametr nastavení.

  Kontext:

  ```
  <param name="{název parametru}"
         invalidAttribute="{název chybného atributu}"/>
  ```

- „`missing document uri`“ – Synchronizační zpráva bez adresy zdrojového dokumentu.
  - V synchronizační zprávě chybí adresa dokumentu.
- „`missing document content`“ – Synchronizační zpráva bez obsahu dokumentu.
  - V synchronizační zprávě chybí obsah dokumentu (i když je dokument prázdný, stále obsahuje základní značky jako `body`, takže není povoleno synchronizovat s prázdným řetězcem).
- „`bad fragment`“ – Chybný anotovaný fragment.
  - Chybný anotovaný fragment.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`attribute malformed`“ – Chybný atribut anotace.
  - Některá část atributu anotace chybí nebo je špatná – tato chyba se netýká hodnoty.

  Kontext:

  ```
  <attribute annotationUri="{URI anotace}" uri="{URI atributu}"
             name="{název atributu}"
             invalidProperty="{název vlastnosti}"/>
  ```

  `{označení atributu}` má 2 varianty, proto jsou zde 2 atributy `uri` a `name` z nichž bude vždy využit pouze jeden.

  `{název vlastnosti}` je `name`, `uri`, `designation`, `type`, `priority`, `nesting` (`linkedAnnotation` nebo `nestedAnnotation`) nebo `nestedIn`

- „`bad confirm`“ – Chybný indikátor změny nebo identifikátor nabídky.
  - Chybný identifikátor potvrzené či odmítnuté nabídky nebo chybný atribut `modified`.

  Kontext:

  ```
  <suggestion uri="{URI návrhu}"/>
  ```

- „`changed annot not found`“ – Editovaná anotace nebyla nalezena. Změny nelze uložit.
  - Anotace, která má být změněna, nebyla nalezena.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`rem annot not found`“ – Mazaná anotace nebyla nalezena. Anotaci nelze vymazat.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`session expired`“ nebo „`31`“ – Identifikátor sezení je neplatný – sezení pravděpodobně expirovalo.
  - Identifikátor sezení je neplatný – sezení pravděpodobně expirovalo.
  - Nelze spolehlivě určit využitou verzi protokolu – může být vrácen chybový kód verze 1.0.

  Kontext:

  ```
  <session id="{ID expirovaného sezení}"/>
  ```

- „`bad request`“ nebo „`32`“ – Chybný požadavek. Chyba klienta či nekompatibilní verze protokolu.
  - Chyba při zpracování XML s požadavkem.
  - Nelze spolehlivě určit využitou verzi protokolu - může být vrácen chybový kód verze 1.0.
- „`module error`“ – Chyba v modulu serveru.
  - Interní chyba v modulu anotačního serveru.

  Kontext:

  ```
  <module name="{název modulu, ve kterém došlo k chybě}"/>
  ```
- „`reload annot not found`“ – Požadovaná anotace nebyla nalezena.
  - Anotace, která by měla být znovu zaslána, nebyla nalezena.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```
- „`bad document`“ – Chyba v anotovaném dokumentu.
  - Chyba při zpracování anotovaného dokumentu.
- „`annot malformed`“ – Chyby v anotaci.
  - Některá část anotace chybí nebo je špatná – toto se netýká fragmentů a atributů – ty se kontrolují zvlášť.

  Kontext:

  ```
  <annotation uri="{URI anotace}" invalidProperty="{název vlastnosti}"/>
  ```

  `{název vlastnosti}` je `uri`, `body` (min. jeden z nich musí být uveden), `type`, `content`, `target`, `trix`, `author`, `annotatedAt` nebo `serializedAt`
- „`modification not applicable`“ – Modifikaci textu nelze aplikovat.
  - Modifikace dokumentu je v konfliktu s jinou modifikací, kterou daný klient dosud neaplikoval.

  Kontext:

  ```
  <modification id="{ID modifikace, která nemohla být aplikována}"
         mustApply="{ID poslední aplikované modifikace dokumentu}"/>
  ```

- „`bad sugg type`“ – Neznámý typ anotace – nabízení není možné.

  – V požadavku o nabídky byl požadován neznámý typ anotace.

  Kontext:

  ```
  <type uri="{URI typu anotace}"/>
  ```

- „`not synchronized`“ – Dokument nebyl synchronizován. Manipulace s anotacemi není možná.

  – Dokument není synchronizován.

- „`unknown group`“ – Neznámá skupina uživatelů.

  – Neznámá skupina uživatelů ve zprávě `joinUserGroup` nebo `leaveUserGroup`

- „`used type del`“ – Mazání využitých typů anotací není dovoleno. Nejprve je nutné smazat všechny anotace daného typu.

  – Došlo k pokusu o odstranění využitého typu anotace.

  Kontext:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"/>
  ```

- „`persistence error`“ – Data nebylo možné uložit kvůli interní chybě serveru.

  – Vnitřní chyba serveru, která mohla být způsobena např. výpadkem databáze.

- „`duplicit type`“ – Duplicitní URI typu anotace.

  – Došlo k pokusu o vytvoření duplicitního typu anotace.

  Kontext:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"/>
  ```

- „`bad modification`“ – Popis modifikace textu je chybný.

  – Modifikaci nelze aplikovat kvůli chybám v jejím popisu.

  Kontext:

  ```
  <modification id="{ID modifikace, která nemohla být aplikována}"/>
  ```

- „`type w subtype del`“ – Mazání typů anotací s podtypy není dovoleno. Nejprve je nutné vymazat všechny podtypy.

  – Došlo k pokusu o odstranění typu anotace s podtypy.

Kontext:

```
<type name="{název typu anotace}" uri="{URI typu anotace}"/>
```

- „`ambiguous fragment`“ – Nejednoznačný fragment (lze jej nalézt na více místech v dokumentu).

  – Nejednoznačný fragment v anotaci.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`bad document uri`“ – Chybný URI anotovaného dokumentu.

  – Chybný URI anotovaného dokumentu při synchronizaci (chybná syntaxe URI).

- „`type ancestors malformed`“ – Seznam předků přidávaného typu anotace je chybný.

  – Seznam předků typu anotace obsahuje něco špatného.

  Kontext:

  ```
  <type name="{název typu anotace}" uri="{URI typu anotace}"/>
  ```

- „`join administrators`“ – Nelze se přidat do skupiny administrátorů. Tuto operaci smí provést pouze administrátor.

  – Pokus o vstup do skupiny administrátorů.

  Kontext:

  ```
  <group name="{název skupiny}" uri="{URI skupiny}"/>
  ```

- „`last admin`“ – Nemůžete opustit skupinu administrátorů, protože jste jejím posledním členem.

  – Když poslední administrátor opustí skupinu administrátorů, nebude existovat žádný administrátor a nikdo nebude schopen vytvořit nového.

- „`auto update failed`“ – Některé anotace by měly být aktualizovány, ale tyto změny nebylo možné uložit.

  – Interní chyba serveru způsobila, že automatická aktualizace existujících anotací pro daný dokument selhala. Když toto nastalo, je možné, že některé anotace budou mít neplatné fragmenty. Ale při příští synchronizaci nebo modifikaci daného dokumentu by se server měl pokusit o opětovnou aktualizaci a vše opravit.

- „`SEC not available`“ – SEC (Semantic Enrichment Component) server není dostupný (nabídky anotací nebudou dostupné).

  - Semantic Enrichment Component není dostupný – nebude možné vytvářet a aktualizovat nabídky anotací.

- „`sync error other different`“ – Jiný klient pracuje s jinou verzí tohoto dokumentu.

  - Jiný klient (či klienti) pracuje s jinou (pravděpodobně aktualizovanou) verzí tohoto dokumentu. Je třeba se dotázat uživatele, zda má být jeho verze aktualizována, nebo zda má jeho verze nahradit verzi ostatních.

  Kontext:

  ```
  <content><![CDATA[Obsah kopie dokumentu uložené na serveru]]></content>
  ```

- „`empty entity filter`“ – Špatný filtr pro název v požadavku na entity.

  - Filtr pro název nesmí být prázdný (není dovoleno číst celou znalostní bázi).

- „`unknown sub uri`“ – Identifikátor odběru nebyl nalezen.

  - Identifikátor odběru nebyl nalezen. Není možné se odhlásit ani přihlásit k odběru ani měnit či mazat daný odběr.

- „`subscription malformed`“ – Odběr je chybný.

  - Některá část popisu odběru chybí nebo je špatná.

  Kontext v případě, že se jedná o nový odběr:

  ```
  <subscription tmpId="{dočasný identifikátor odběru}"
                invalidProperty="{název vlastnosti}"/>
  ```

  Kontext v případě, že se jedná o modifikovaný odběr:

  ```
  <subscription uri="{URI odběru}"
                invalidProperty="{název vlastnosti}"/>
  ```

  `{název vlastnosti}` je `tmpId`, `name`, `uri` nebo `source` (pro popis některého ze zdrojů)

- „`empty composite`“ – Composite fragmentu je prázdný.

  - Klient deklaroval cíl jako kompozitní, ale prázdný, což není dovoleno.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`duplicit attribute of type`“ – Duplicitní atribut typu anotace.

  - Byl nalezen duplicitní atribut typu anotace.

Kontext:

```
<attribute typeName="{název typu anotace}"
           typeUri="{URI typu anotace}" name="{název atributu}"
           ontologyUri="{URI atributu v ontologii}"/>
```

- „`missing confidence`“ – V požadavku o nabídky chybí minimální důvěryhodnost.
  - V požadavku o nabídky chybí minimální důvěryhodnost.
- „`bad max entities`“ – Chybný formát maximálního počtu výsledků v požadavku o entity.
  - Maximální počet výsledků v požadavku o entity není číslo.
- „`not in group`“ – Nejste v žádné skupině uživatelů – nelze manipulovat s typy anotací, anotacemi apod.
  - Aby uživatel mohl anotovat, musí být v nějaké skupině uživatelů.
- „`unsupported operation`“ – Nepodporovaná operace.
  - Po serveru byla požadována nepodporovaná operace.
- „`unknown error`“ – Neznámá chyba.
  - Neznámá chyba zjištěná na serveru.
  - Pro více detailů se prosím podívejte do logu serveru.
- „`suggestions not ready`“ – Nabídky ještě nejsou připraveny, zkuste to prosím později.
  - Klient se pokouší využít automatické potvrzování nabídek anotací, ale nabídky ještě nejsou dostupné.
- „`missing modified sug`“ – Chybí modifikovaný návrh anotace.
  - Bylo deklarováno, že návrh je modifikovaný, ale modifikovaná podoba chybí.

  Kontext:

  ```
  <suggestion uri="{URI návrhu}"/>
  ```

- „`modification specification`“ – Chybně specifikovaná modifikace.
  - Modifikace obsahuje popis neexistujícího fragmentu apod.

  Kontext:

  ```
  <modification id="{ID modifikace, která nemohla být aplikována}"/>
  ```

- „`duplicit subscription`“ – Byl nalezen duplicitní odběr.

  – Byl nalezen duplicitní odběr.

  Kontext:

  ```
  <subscription tmpId="{dočasný identifikátor odběru}"/>
  ```

- „`bad autoconfirm`“ – Chybná míra důvěryhodnosti pro automatické potvrzování návrhů.

  – Chybná míra důvěryhodnosti pro automatické potvrzování návrhů.

F.11.4 Seznam kódů varování

- „`server error`“ – Nekritická interní chyba serveru.

  – Nastala nějaká interní chyba serveru, neměla by však mít vliv na funkcionalitu z vnějšího pohledu.

- „`annot superseded`“ – Anotace odstraněna.

  – Server odstranil nějakou anotaci, protože již nebyla platná.

- „`annot orphaned`“ – Anotace osiřela (fragmenty byly zneplatněny).

  – Fragmenty některé anotace již nebyly nalezeny v textu a bude tedy zobrazena na úrovni celého dokumentu.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`annot updated`“ – Anotace byla automaticky aktualizována.

  – Klient zaslal novou či modifikovanou anotaci s fragmenty, které nebyly na správném místě, ale byly spolehlivě aktualizovány (např. posun o několik znaků způsobený tím, že jiný uživatel právě píše před anotovaný text).

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`annot partially orphaned`“ – Anotace částečně osiřela (některé fragmenty byly zneplatněny).

  – Část fragmentů některé anotace již nebyla nalezena v textu.

  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

- „`not logged`“ – Nejste přihlášen. Můžete se pouze přihlásit nebo odpojit (ostatní zprávy budou ignorovány).
  - Klient je připojený, ale uživatel dosud nebyl přihlášen.
- „`fragments updated`“ – Fragmenty anotace byly aktualizovány.
  - Fragmenty některé z již uložených anotací byly aktualizovány kvůli modifikaci dokumentu.

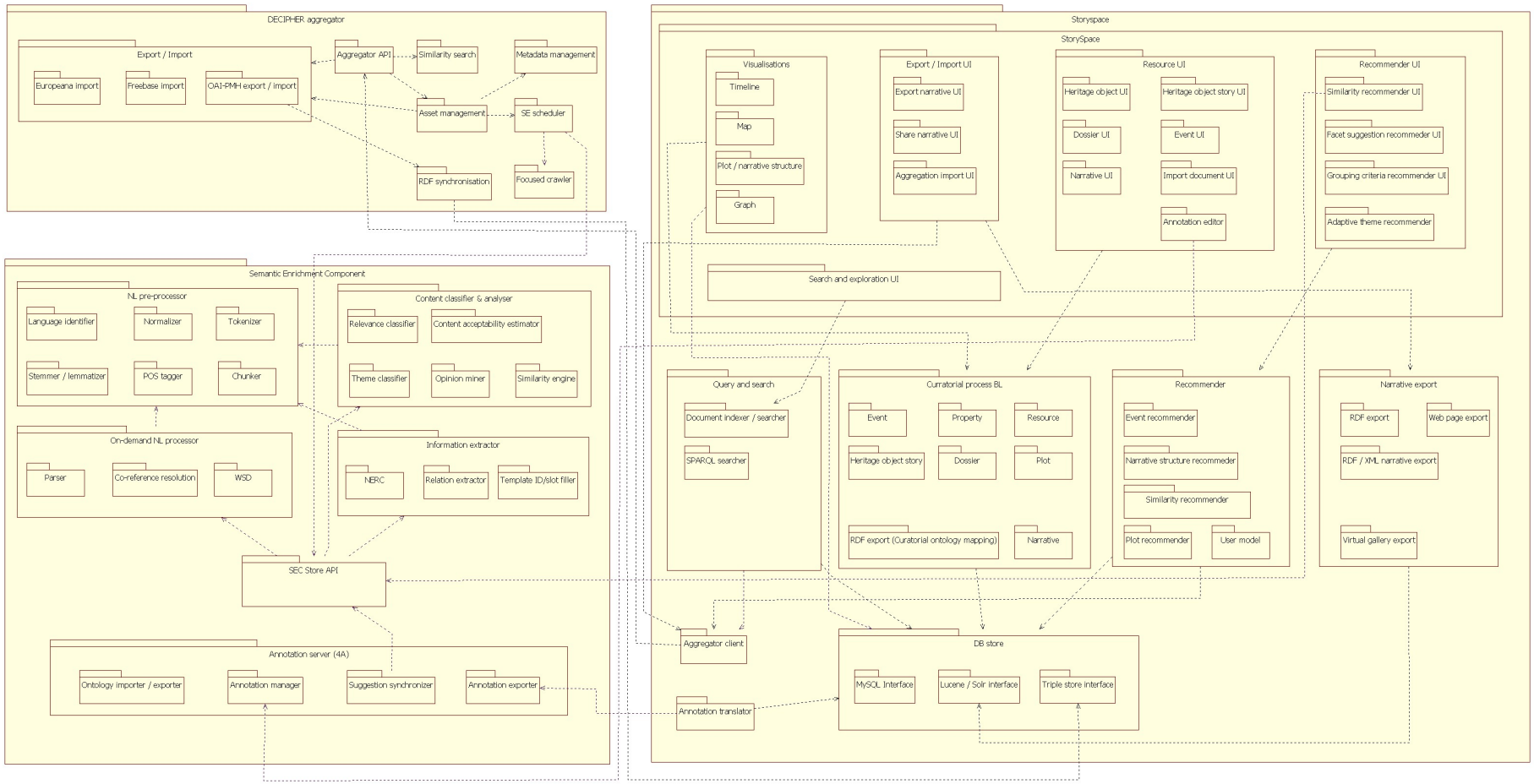
  Kontext:

  ```
  <annotation uri="{URI anotace}"/>
  ```

Příloha G

Integrace v systému z projektu Decipher

V této příloze je umístěn diagram balíčků výsledného systému vytvořeného v rámci projektu Decipher, ze kterého je patrná integrace anotačního nástroje 4A. Klient (Annotation editor) je integrován do uživatelského rozhraní systému (StorySpace). Anotační server je integrován v SEC (Semantic Enrichment Component). Nutno poznamenat, že se jedná o jiný SEC než je zmíněn v textu této práce, protože po ukončení projektu Decipher byla vyvinuta nová verze SEC, ve které již anotační server a *SEC Store API* (API pro indexaci a vyhledávání v indexu) nejsou zahrnuty.



Obrázek G.1: Diagram balíčků systému z projektu Decipher