SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4198

Programski alati za automatsko mapiranje tematskih područja

Žad Deljkić

Zagreb, srpanj 2015.

SADRŽAJ

1. Uvod

U akademskoj zajednici i šire često se javlja potreba za istraživanjem tematskih područja. Iako postoje brojni alati za podršku ovom procesu, prvenstveno alati za izradu umnih mapa i vođenje bibliografije, takav pristup istraživanju je vremenski intenzivan te je postupak vizualizacije rezultata kompleksan. Cilj ovog rada bio je razviti alate koji nude prihvatljivo rješenje za identificirane probleme.

U sklopu ovog rada najprije je dan pregled identificiranih problema u 2. poglavlju. Glavne značajke postojećih rješenja opisane su u 3. poglavlju, dok je programsko rješenje koje je poslužilo kao osnova za daljnje nadogradnje opisano u 4. poglavlju. Ciljevi razvijenog rješenja su opisani u 5. pogavlju. Razvijeno rješenje, primjer scenarija istraživanja koje ono podržava i ostvareni rezultati su opisani u 6. poglavlju.

2. Problemi klasičnog pristupa istraživanju

U svrhu boljeg razumijevanja, opisan je klasični pristup istraživanju kroz primjer.

Stručnjak u području informacijske sigurnosti želi napisati rad o novom načinu povrata obrisanih datoteka. Iako je razvio novu tehniku povrata obrisanih datoteka, nije dobro upoznat sa svim radovima, osobama i organizacijama koji bi mogli biti relevantni.

Prvo je potrebno pronaći relevantne radove i istraživanja. Zatim, saznati koje su osobe i organizacije značajne u tom području. Također, treba saznati postoje li događaji koji su relevantni istraživanju, npr. konferencije.

Knjige i radove je moguće pronaći pretražujući različite baze. Zatim treba procijeniti koji su radovi više, a koji manje značajni u kontekstu istraživanja koje se provodi. U nekim bazama postoje mjerila koja mogu pomoći u tome, no općenito to nije lak posao. Što se tiče osoba i organizacija, moguće je dio njih pronaći gledajući autore i organizacije povezane s radovima. Događaje, kao na primjer konferencije, je također moguće naći pretražujući baze namijenjene za tu svrhu.

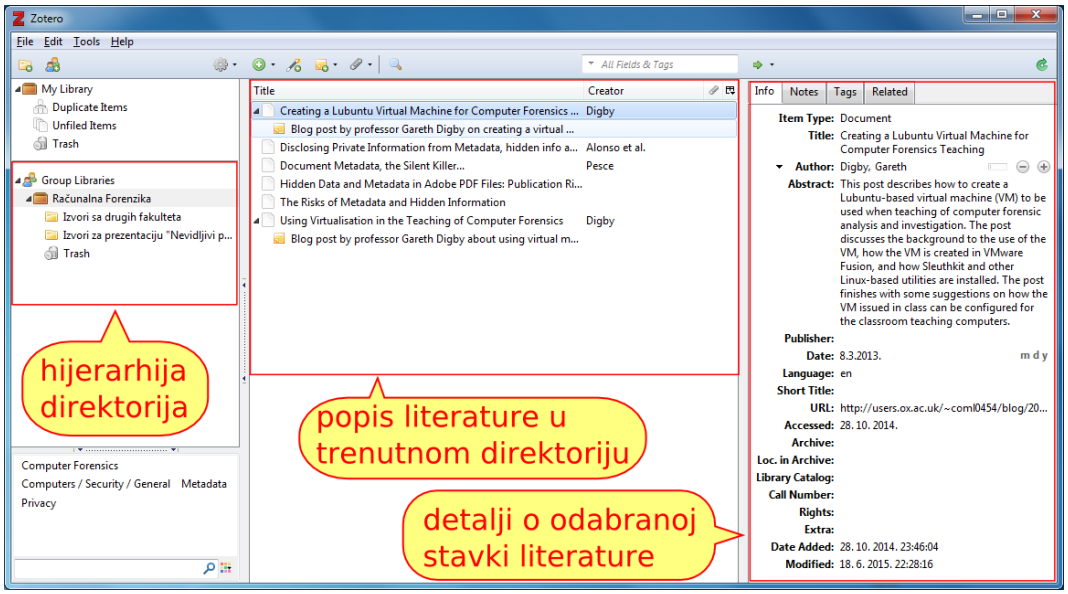
Problem koji se odmah javlja kod ovakvog pristupa istraživanju je vremenska intenzivnost prikupljanja podataka. Iste ili slične pretrage ponavljaju se u različitim bazama.

Slično, kod traženja radova, prilikom pretraživanja svake baze je obično potrebno ponoviti pretragu s nekoliko različitih fraza. Na primjer, kod pretrage radova uz frazu „digitalna forenzika“, često će se pretraživati i radovi uz frazu „računalna forenzika“. Porastom broja baza radova i fraza za pretraživanje vremenska intenzivnost ovakvog postupka naglo raste.

Dobivanje informacija o značajnim osobama i organizacijama je također vremenski intenzivan postupak. Jedan pristup tome je bilježenje koji autori i organizacije su sudjelovali u izradi pronađenih radova. S velikim brojem radova ovaj postupak isto postaje vremenski intenzivan.

Jednom kada su podaci prikupljeni, potrebno ih je organizirati, prikazati i na kraju podijeliti.

Jedna opcija za to je grupiranje bibliografije u relevantne podgrupe i dodavanje bilježaka, primjerice koristeći alat *Zotero*[9]. Pomoću Zotera, samo dodavanje zapisa u biblioteku je prilično efikasno. Zotero je nakon instalacije integriran u Internet preglednik te jednom kada je pronađena zanimljiva stavka, za njeno dodavanje u biblioteku je dovoljno samo pritisnuti tipku na sučelju. Ipak, ovakvim pristupom nije moguće vizualno prikazati veze i značaj već isključivo hijerarhijsku vezu direktorija i poddirektorija u koje su zapisi dodani. Primjer Zotero biblioteke prikazan je na slici 2.1.



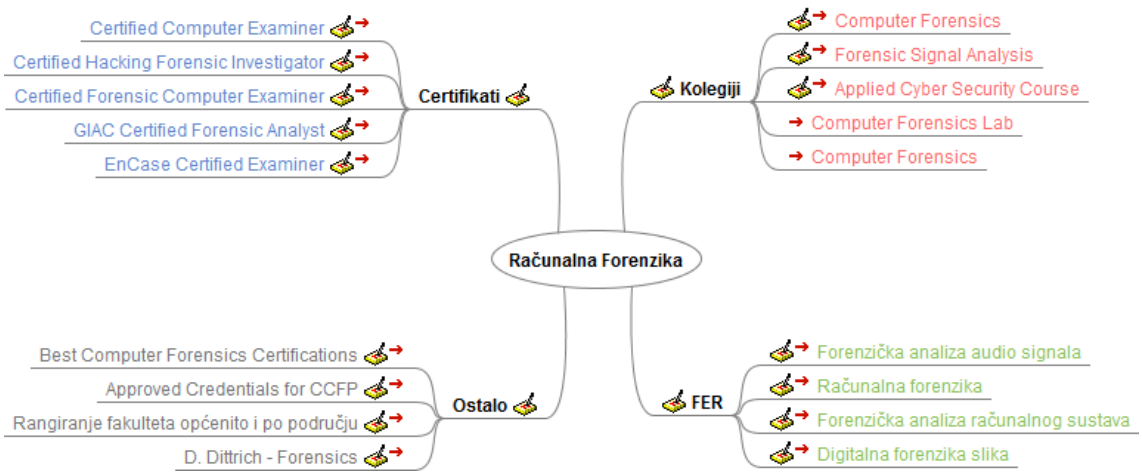
Slika 2.1: Primjer Zotero biblioteke

Još jedno često korišteno rješenje je *Freemind*[2], alat namijenjen za izradu umnih mapa (engl. *mind map*). Za razliku od Zotera, koji omogućava vremenski efikasno dodavanje zapisa, svi elementi u umnu mapu moraju biti dodani ručno. Dodatno ograničenje je što se ne mogu vizualno prikazati proizvoljne veze između elemenata — oni su isključivo hijerarhijski organizirani. Primjer umne mape prikazan je na slici 2.2.

Zaključno, dva temeljna problema koji se javljaju su:

1. Cijelokupni postupak istraživanja je vremenski intenzivan, od prikupljanja podataka do prikaza rezultata

2. Postupak organizacije i vizualnog prikaza rezultata je kompleksan i nedovoljno razvijen, nema alata koji omogućavaju prikazivanje proizvoljnih veza i značaja stavki



Slika 2.2: Primjer umne mape

3. Postojeća programska rješenja

Postojeća programska rješenja koja rješavaju sve identificirane probleme ne postoje. Alati kao što su Zotero samo ublažavaju simptome neefikasnog istraživanja. Umne mape uz alate kao Freemind solidno rješavaju problem prikaza rezultata — no i dalje je potrebno nekako provesti istraživanje te izraditi umnu mapu.

Najbliže rješenju koje rješava identificirane probleme je predloženo u radu *Automatic generation of research trails in web history*[11], te je opisan razvijeni prototip u radu *Research trails: getting back where you left off*[10]. Predložen je koncept istraživačkih tragova (engl. *research trails*) koji bi pomogao korisnicima weba stvoriti i obnoviti kontekst kroz fragmentirane istraživačke procese bez da oni sami moraju eksplicitno strukturirati i organizirati materijal.[11]

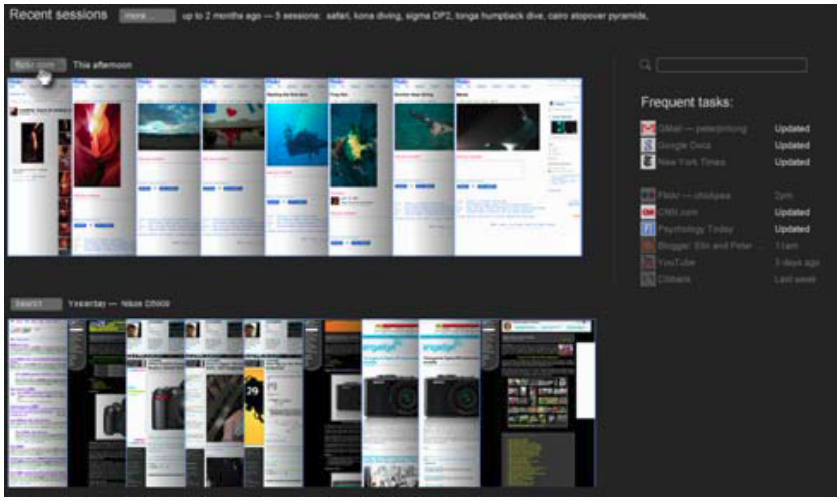
Pedersen, Gyllstrom, Gu, i Hong [11] definiraju pojam istraživačkog traga kao uređeni slijed web stranica posjećenih u sklopu većeg istraživanja. U sklopu predloženog rješenja su istraživački tragovi automatski generirani iz korisnikove povijesti posjećenih web stranica. Automatsko generiranje je izvedeno filtrirajući korisnikovu web povijest pomoću semantičkih kriterija i kriterija temeljenima na korisnikovoj aktivnosti kako bi grupirali slične posjećene web stranice.[11]

Takvo rješenje omogućava korisniku da istraži temu koja ga zanima na način na koji bi to prirodno učinio pomoću Internet preglednika, bez vođenja brige o prikupljanju i organiziranju materijala. Jednom kada želi nastaviti istraživanje ili pogledati rezultate, to mu je omogućeno pomoću automatski generiranih istraživačkih tragova.

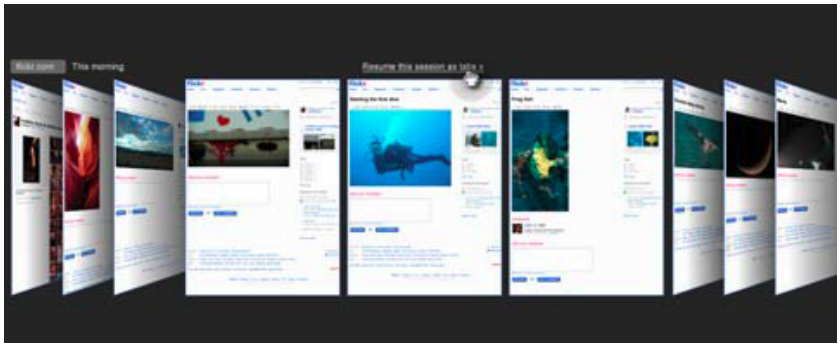
Slike 3.1 i 3.2 iz rada *Research trails: getting back where you left off*[10] ilustriraju jedan način vizualnog prikaza istraživačkih tragova korisniku. Slika 3.1 prikazuje dva istraživačka traga prikazana u prozoru preglednika, dok slika 3.2 prikazuje detaljniji prikaz dijela istraživačkog traga koji se prikaže kada korisnik pozicionira pokazivač na njega.

No i kod ovakvog rješenja je postupak prikupljanja podataka vremenski intenzivan jer korisnik i dalje ručno mora pretraživati baze.

Što se tiče vizualnog prikaza rezultata, ovakav pristup je dobar isključivo kada



Slika 3.1: Dva istraživačka traga prikazama u prozoru preglednika



Slika 3.2: Detaljniji prikaz dijela istraživačkog traga

je moguće jednu elementarnu stavku informacije povezati s jednom web stranicom. To je primjerice često moguće za knjige (stranica koja opisuje knjigu) i organizacije (stranica same organizacije), no ne i za osobe. Za efektivan proces istraživanja je potreban mehanizam razlikovanja različitih vrsta elemenata, dok uz ovo ograničenje to nije moguće. Kod ovog rješenja su svi elementi web stranice te nije moguće lako znati opisuju li one primjerice knjigu, osobu ili nešto treće.

Uz to, ne postoji ni mehanizam koji bi razlikovao elemente po značaju. Primjerice, nije moguće razlikovati značajniju organizaciju od manje značajne u kontekstu istraživanja, kako god definirali značajnost u tom slučaju.

Pristup koji koristi istraživačke tragove namijenjen je za općenito istraživanje weba i olakšavanje ranih faza istraživanja, te je u tome uspješan. No što se tiče cijelog postupka istraživanja i mapiranja tematskih područja, i dalje ne rješava identificirane probleme.

4. Alati za prikupljanje i vizualizaciju javno dostupnih podataka

Iako ne postoje alati koji rješavaju identificirane probleme, postoje drugi alati koje je moguće iskoristiti u izradi konačnog programskog rješenja. Alati za prikupljanje javno dostupnih podataka (engl. *open-source intelligence*)[8] i njihovu vizualizaciju imaju karakteristike poželjne u rješavanju identificiranih problema, samo ih obično koriste u druge svrhe. Najčešće su korišteni u kontekstu informacijske sigurnosti gdje služe za mapiranje organizacija u svrhu poboljšanja sigurnosti.

4.1. Maltego

Jedan od najpopularnijih takvih alata trenutno je Maltego[3]. Na službenoj stranici alata Maltego je opisan kao platforma razvijena da dostavi jasnu sliku sigurnosnih prijetnji informacijskom sustavu organizacije.[3] Uz to, piše da Maltego može biti korišten za fazu prikupljanja informacija svakog posla vezanog uz sigurnost te da on pomaže u misaonom procesu vizualno demonstrirajući međusobne veze između traženih stavki[3].

Korisnički vodič za Maltego verziju 3 ukratko opisuje kako alat funkcionira:

Maltego uzima razne komadiće informacija (koje naziva Entiteti unutar aplikacije) te ih pretvara (pomoću koda kojeg naziva transformacije) u druge Entitete. Primjer toga bi bio kada biste stavili Entitet web sjedišta na graf unutar Maltega s vrijednošću 'www.paterva.com' i pokrenuli 'To IP Address [DNS]' transformaciju. Dobili biste zatim obavijest da je novi Entitet, konkretno IP adresa s vrijednošću 74.207.243.85, generiran kao dijete originalnog entiteta web sjedišta.[5]

Entitet može biti bilo što što sadrži informaciju. Često se entiteti definiraju samo elementarnom informacijom umjesto većim skupom informacija. Tako se u konačnici mogu bolje vizualno prikazati veze među njima. Primjerice, umjesto entiteta koji je definiran URL-om web stranice, organizacijom koja ju posjeduje i njenim IP adresama,

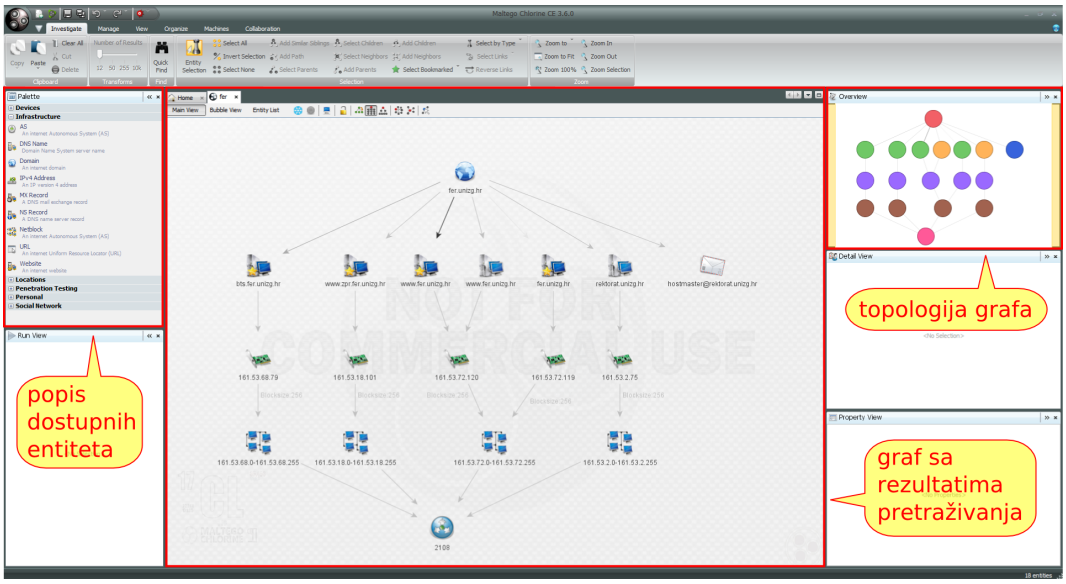
obično bi postojao poseban entitet za URL, organizaciju i svaku IP adresu, te bi bile prikazane veze između njih.

Transformacijama je iz jednog entiteta moguće dobiti jedan ili više drugih entiteta bilo koje vrste.

Uz postojeće entitete i transformacije, korisnik može definirati i vlastite.

Maltego vizualno prikazuje entitete i njihove veze kao grafove. Vrhovi grafa predstavljaju entitete, dok se grane nalaze između ulaznih i izlaznih entiteta kod transformacija te tako predstavljaju veze između njih.

Korisničko sučelje Maltega s označenim osnovnim dijelovima je prikazano na slici 4.1. Na sredini prozora se nalazi glavni dio grafičkog sučelja — prikaz svih entiteta kao rezultata pretraživanja. Na lijevom dijelu je označen popis entiteta koje je moguće pokazivačem povući i dodati na graf. Na desnom dijelu je označen prikaz topologije grafa gdje su različite vrste entiteta označene različitom bojom. Na slici je u glavnom dijelu sučelja prikazan rezultat jednostavne analize domene *fer.unizg.hr*. Iz originalnog entiteta domene su transformacijama generirani entiteti web sjedišta (npr. *www.zpr.fer.unizg.hr*), DNS servera (npr. *rektorat.unizg.hr*) i entitet e-mail adrese *hostmaster@rektorat.unizg.hr*. Dalje su još generirani entiteti IP adresa, IP blokova i jedan entitet autonomnog sustava kojem oni pripadaju.



Slika 4.1: Maltego korisničko sučelje

Uz već spomenute entitete, drugi primjeri entiteta su osoba, dokument, pa čak i *tweet*.

Jednostavan primjer transformacije je već spomenuta transformacija koja iz entiteta web sjedišta generira entitete IP adresa postupkom DNS razrješavanja (engl. *DNS*

resolution).

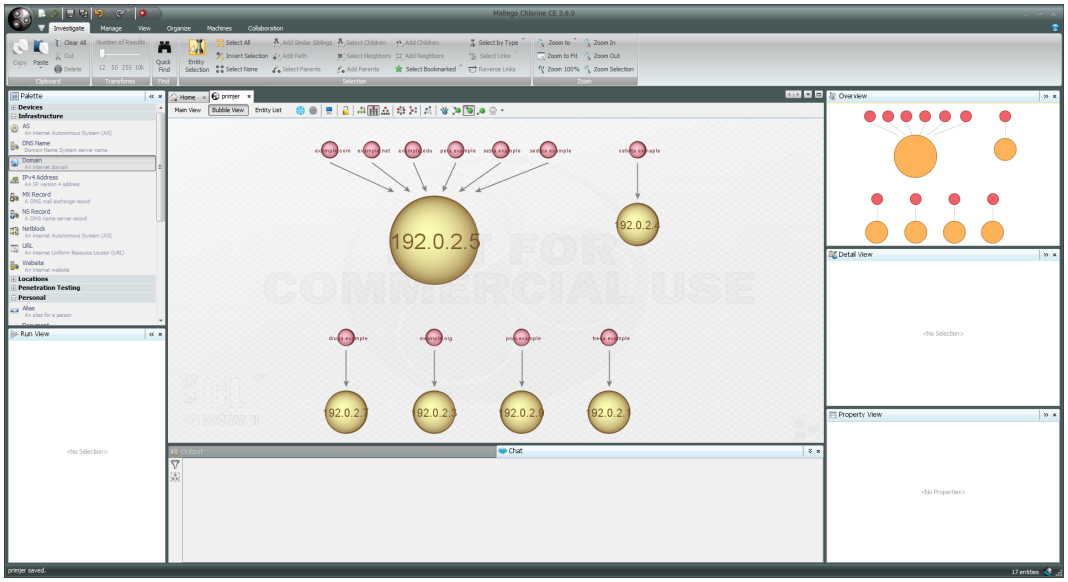
Primjer malo kompleksnije transformacije je transformacija koja iz entiteta osobe generira entitete URL-a pomoću Internet tražilica. Takva transformacija pretražuje web koristeći ime osobe kao ulaznu frazu te zatim dobivene URL-ove vraća u obliku entiteta.

Što se tiče izvršavanja samih transformacija, one se mogu izvršavati lokalno na računalu krajnjeg korisnika ili na udaljenom poslužitelju, ovisno o potrebama.

Ukoliko transformacija vrati entitet kojeg Maltego smatra identičnim nekom entitetu koji je već na grafu, on će ih spojiti umjesto da ostavi dva čvora koja zapravo prikazuju isti entitet.

Jednom kada na grafu postoji veći broj entiteta i veza između njih, moguće je odabrati različite načine prikaza koji omogućuju izvlačenje informacija iz njih. Moguće je različito grupirati entitete ovisno o vezama (npr. hijerarhijski ili kružno) te na primjer podesiti da im veličina ovisi o broju ulaznih ili izlaznih grana.

Primjerice, na grafu se nalazi nekoliko entiteta domena i IP adresa. Iz više različitih domena je transformacijom dobivena IP adresa 192.0.2.5, dok su iz svih ostalih domena dobivene različite IP adrese. IP adresa 192.0.2.5 u ovakvom kontekstu ima poseban značaj. Ukoliko je podešeno da veličina entiteta ovisi o broju ulaznih grana, entitet IP adrese 192.0.2.5 će biti veći od svih ostalih entiteta IP adresa i tako će se isticati njegov značaj. Slika 4.2 ilustrira upravo ovaj slučaj.



Slika 4.2: Maltego sučelje uz podešeno skaliranje entiteta ovisno o broju ulaznih grana

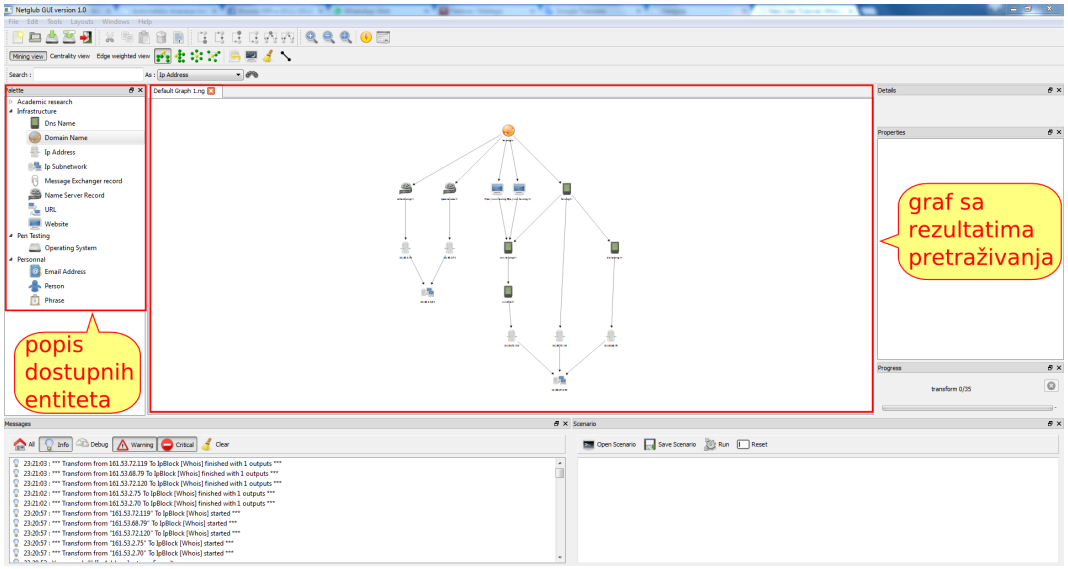
Maltego je moguće preuzeti u dvije različite verzije, *community* i *commercial*. Glavna razlika između njih je što je *community* verzija besplatna, no ima ograniče-

nje od maksimalno 12 rezultata po transformaciji, dok se *commercial* verzija plaća, no zato nema nikakvih ograničenja.[4]

4.2. Netglub

Netglub[6] je alat identičan Maltegu po svrsi, te ima i sve glavne mogućnosti Maltega. Glavna razlika između njih je što je Netglub u potpunosti slobodan (engl. *free and open source*) te je zato besplatan i nema nikakvih ograničenja.

Na isti način kao Maltego definira pojmove entiteta i transformacija te konceptualno isto funkcionira. Korisničko sučelje Netgluba je prikazano na slici 4.3. Isto kao i Maltegovo sučelje, Netglubovo sučelje ima popis entiteta s lijeve strane te u sredini glavni dio sučelja koji prikazuje rezultate pretraživanja.



Slika 4.3: Netglub korisničko sučelje

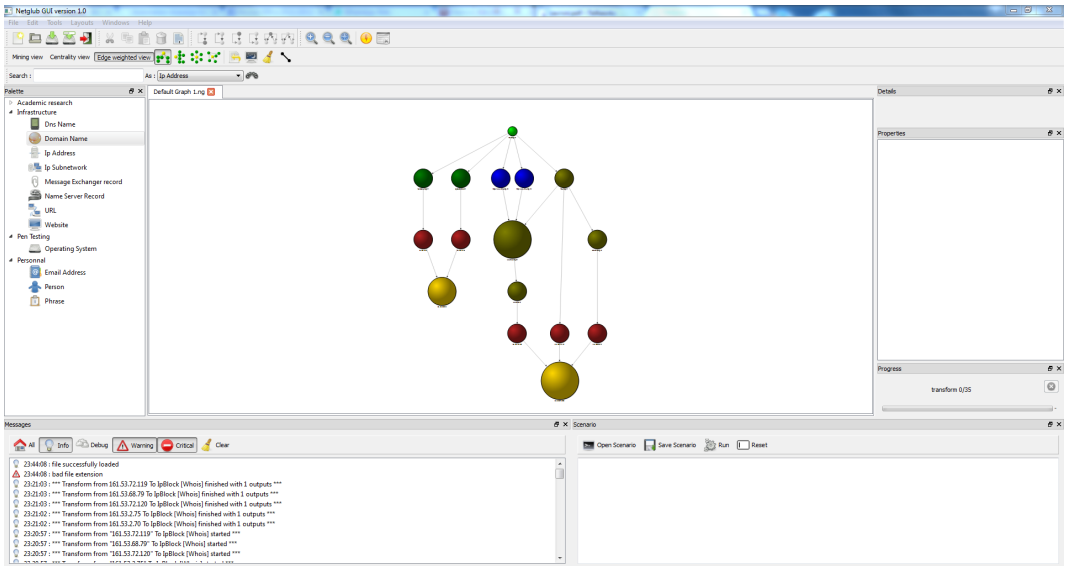
Isto kao i Maltego, ukoliko transformacija vrati entitet za koji Netglub smatra da je identičan postojećem entitetu na grafu, on će ih spojiti umjesto da ostavi dva odvojena čvora.

Konkretno, u Netglubu su entiteti definirani barem jednom obveznom stavkom i proizvoljnim brojem neobaveznih stavki. Netglub smatra da su dva entiteta identična ukoliko im se podudaraju sve obvezne stavke.

Primjerice, entitet vrste URL je definiran obveznom stavkom *value* koja sadrži vrijednost URL-a te neobaveznom stavkom *title* koja sadrži naslov resursa na koji URL pokazuje (ukoliko on postoji). Ako na grafu već postoji primjerice URL entitet sa stavkom *value* jednakom „http://example.com“ i stavkom *title* jednakom „Example

Domain“ te neka transformacija vrati URL entitet sa stavkom *value* isto jednakom „http://example.com“, ona će biti spojena s prethodno navedenim entitetom neovisno o neobaveznoj stavki *title*.

Kao i Maltego, Netglub nudi mogućnost da veličina entiteta skalira ovisno o broju ulaznih ili izlaznih grana. Slika 4.4 prikazuje isti graf kao slika 4.3 samo uz uključenu opciju skaliranja veličine entiteta ovisno o broju ulaznih grana.



Slika 4.4: Netglub sučelje uz podešeno skaliranje entiteta ovisno o broju ulaznih grana

Što se tiče izvođenja transformacija i prikaza rezultata, Netglub razdvaja cijeli proces u tri različita tipa programa: *gospodar* (engl. *master*), *sluga* (engl. *slave*) i *klijent*. U jednom Netglub sustavu se nalazi točno jedan gospodar, barem jedan sluga i bilo koji broj klijenata.

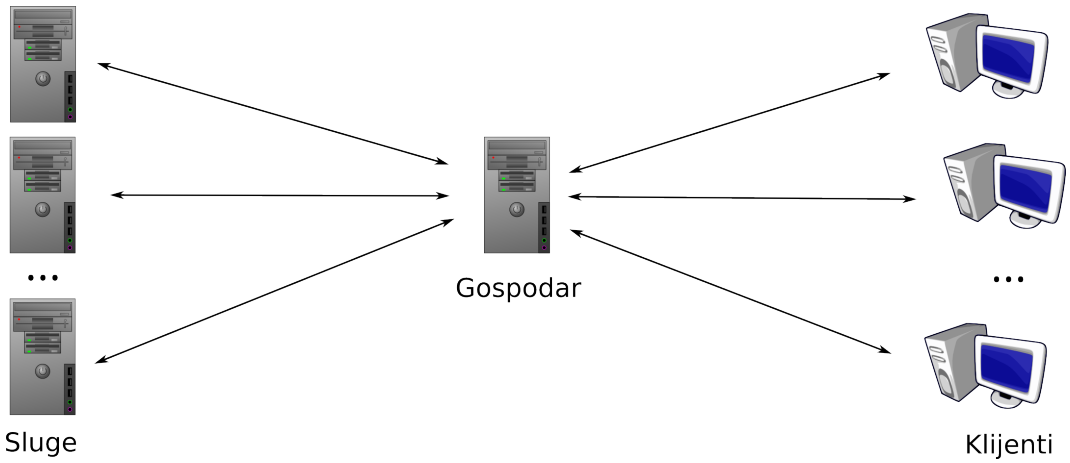
– Gospodar je zadužen za koordiniranje sluga i klijenata.

– Sluge su zaduženi za obavljanje samih transformacija — oni dobivaju ulazne entitete od gospodara uz informaciju o tome koju transformaciju treba provesti, te mu zatim vraćaju rezultat.

– Svrha klijenta je da bude sučelje prema korisniku. On je zadužen za primanje naredba od korisnika, njihovo prenošenje gospodaru i prikaz rezultata koje dobije nazad od njega.

Komunikacija gospodara, sluga i klijenata se odvija mrežno. Ukoliko korisnik želi koristiti Netglub samo na svojem računalu, može pokrenuti sva tri programa lokalno na svojem računalu. No isto tako je i moguće raspodijeliti programe gospodara i sluga na više računala i efektivno pružati uslugu korištenja sustava korisnicima koji se spajaju

s klijentom. Tako izvršavanje transformacija obavljaju poslužiteljska računala, dok računala klijenata samo moraju mrežno komunicirati i crtati rezultate. Slika 4.5 ilustrira opisanu arhitekturu Netglub sustava.



Slika 4.5: Arhitektura Netglub sustava

Netglub, kao i Maltego, pruža korisniku mogućnost definiranja vlastitih entiteta i transformacija. Time je moguće proširiti osnovne mogućnosti alata. Netglubov skup gotovih transformacija nije toliko razvijen kao Maltegov, tako da je ova mogućnost ključna.

Sve u svemu Netglub nema sve tehničke značajke razvijene koliko i Maltego, što je i za očekivati jer je Maltego komercijalni alat dok je Netglub slobodan projekt otvorenog koda. No Netglub potpuno dobro obavlja osnovni proces prikupljanja i prikazivanja podataka te ga je moguće proširiti vlastitim entitetima i transformacijama. Uz to, za razliku od besplatne *community* verzije Maltega, Netglub nema nikakvih ograničenja jer je u potpunosti slobodan kao alat. Upravo to je razlog zbog kojega se konačno rješenje razvijeno u sklopu ovog rada temelji na Netglubu, a ne na Maltegu.

5. Ciljevi razvijenog rješenja

Svrha ovog rada je razvijanje alata koji rješava identificirane probleme kod dosadašnjih pristupa istraživanju tematskih područja. Potrebno je da alat:

1. Može raditi s bilokojim izvorom podataka.
2. Automatizira postupak prikupljanja podataka koliko god je to moguće, tražeći korisničku interakciju samo kada je to nužno.
3. Prikupljene podatke automatski prikaže strukturirano, s njihovim međusobnim vezama i vizualnom naznakom značaja.
4. Omogući korisniku pohranu rezultata u obliku prikladnom za daljnju analizu.

S tim potrebama zadovoljenim, razvijenim alatom će biti moguće brzo obuhvatiti temeljne informacije o tematskom području, grafički ih prikazati na način prikladan za interpretaciju te pohraniti rezultate u obliku pogodnom za daljnju analizu.

Konačni alat je namijenjen svim korisnicima koji imaju potrebu za istraživanjem tematskih područja. To su primjerice stručnjaci koji se započinju baviti područjem s kojim do sada nisu upoznati ili studenti koji pišu diplomski rad te se trebaju upoznati s područjem svoga rada. U tu svrhu, za krajnjeg korisnika rješenje mora biti jednostavno za početno postavljanje i korištenje.

6. Razvijeno rješenje

Razvijeno rješenje se temelji na proširivanju Netgluba dodatnim entitetima i transformacijama. Uz takvo proširenje moguće je Netglub koristiti za efikasno istraživanje i mapiranje tematskih područja.

6.1. Razvoj entiteta i transformacija

Za razvoj vlastitog entiteta potrebno je prvo izraditi XML datoteku u određenom formatu koja opisuje entitet te sliku koja će prikazivati entitet unutar Netgluba u *png* i *svg* formatu.

XML datoteka primarno definira naziv entiteta i njegove stavke (njihov naziv, format, jesu li obvezne ili ne). Slika u *png* formatu je prikazana u popisu entiteta, dok slika u *svg* formatu služi za prikaz entiteta na grafu (*svg* format je u tom slučaju prikladniji jer može skalirati bez gubitka kvalitete). Primjer razvijenog entiteta je dan u dodatku A.

Jednom kada su XML datoteka i slike izrađeni, potrebno ih je postaviti u direktorij entiteta čija je pozicija definirana u konfiguracijskoj datoteci gospodara te upisati podatke o entitetu u bazu podataka na koju se gospodar spaja.

Slično kao za razvoj entiteta, i za razvoj transformacije je potrebno definirati XML datoteku koja ju opisuje. XML datoteka u ovom slučaju opisuje ime transformacije, ulazne i izlazne tipove entiteta, parametre koje transformacija može od korisnika primiti (primjerice željeni broj rezultata) te željeni način primanja ulaznih podataka (argumenti pri pozivu ili standardni ulaz).

Uz XML datoteku koja opisuje transformaciju, potrebno je i napisati samu transformacije kao program. Ona može biti u bilokojem formatu dok god ju gospodar može izvršiti. Transformacija ulaz dobiva kroz argumente pri pozivu ili na standardni ulaz te vraća rezultat na standardni izlaz. Za Python i PHP je već razvijen API koji se bavi primanjem ulaznih i ispisivanjem izlaznih entiteta u ispravnom formatu, tako da ako korisnik koristi jedan od ta dva jezika za definiranje vlastitih transformacija, ne

mora se brinuti o tome. Primjer razvijene transformacije je dan u dodatku B.

Jednom kada su XML datoteka i program transformacije izrađeni, potrebno ih je postaviti u određene direktorije. U ovom slučaju, samo XML datoteku je potrebno pozicionirati u gospodarov direktorij transformacija, dok je i XML datoteku i program transformacije potrebno pozicionirati u direktorij transformacija svakog sluge koji će moći izvršavati tu transformaciju. Konkretna pozicija oba direktorija je definirana u odgovarajućim konfiguracijskim datotekama. Uz to, podatke o transformaciji je potrebno ubaciti u bazu podataka na koju se gospodar spaja.

6.2. Razvijeni entiteti i transformacije

Entiteti razvijeni u sklopu ovog rada su opisani u tablici 6.1. Uz njih, razvijeno rješenje koristi i već neke postojeće entitete. Oni su opisani u tablici 6.2.

Pri razvoju rješenja je prioritet bio koristiti što više već postojećih entiteta jer je za njih moguće koristiti postojeće transformacije koje iako nisu namijenjene za tu svrhu, mogu biti korisne pri mapiranju tematskih područja. Primjerice, ranija verzija rješenja je uključivala definiranje posebnog entiteta *Author* koji bi opisivao autora. No konačno rješenje za tu svrhu koristi postojeći entitet *Person* jer je on potpuno prikladan za opisati osobu kao autora, te ima dodatnu prednost — moguće je koristiti postojeće transformacije razvijene za njega.

Tablica 6.1: Razvijeni entiteti

Naziv	Kratki opis	Obvezne stavke	Neobavezne stavke
Article	entitet koji opisuje bilo kakav članak ili znanstveni rad	naslov, autori	godina publikacije, URL
Book	entitet koji opisuje knjigu	naslov, autori	godina publikacije, URL
Conference	entitet koji opisuje konferenciju	naslov	URL

Po konvenciji postojećih transformacija u Netglubu, transformacije imaju naziv oblika:
„<*vrste ulaznih entiteta*>To<*vrste izlaznih entiteta*><*skraćenica koja opisuje korišteni servis*>“

Tablica 6.2: Korišteni postojeći entiteti

Naziv	Kratki opis	Obvezne stavke	Neobavezne stavke
Phrase	primarni ulazni entitet, on predstavlja frazu koja opisuje područje koje nas zanima	fraza	
Person	entitet koji opisuje osobu	puno ime	nadimci, dodatne riječi
URL	entitet koji opisuje URL	vrijednost URL-a	naslov resursa na koji url pokazuje
Website	entitet koji opisuje web sjedište	URL web sjedišta	tip servera
Domain	entitet koji opisuje DNS domenu	domena	

Na primjer, transformacija *PersonToEmailSE* uzima entitet koji predstavlja osobu (engl. *person*), vraća entitete koji predstavljaju e-mail adrese, pri tome koristeći Internet tražilice (engl. *search engines*). Pomoću takvog naziva je lagano brzo uočiti bitne informacije o transformaciji, tako da i razvijene transformacije se drže te konvencije.

Razvijene transformacije su:

- *PhraseToPhraseW* - transformacija koja uzima frazu, i pomoću njene *Wikipedia* stranice pronalazi slične fraze. Sama transformacija otvara *Wikiepdia* stranicu fraze i s nekoliko definiranih mjesta na njoj uzima fraze koje su obično povezane s originalnom frazom. Primjerice za ulaznu frazu „računalna forenzika“ (engl. *computer forensics*) bi kao rezultat dobili fraze „digitalna forenzika“ (engl. *digital forensics*), „forenzička znanost“ (engl. *forensics science*) i slično.
- *PhraseToBookLOC* - transformacija koja pretražuje *Library of Congress* (*http://loc.gov/*) za knjige povezane uz ulaznu frazu.
- *PhraseToBookArticleNSK* - transformacija koja pretražuje Nacionalnu Sveučilišnu Knjižnicu (*http://www.nsk.hr/*) za knjige i članke povezane uz ulaznu frazu.
- *PhraseToBookArticleKoha* - transformacija koja pretražuje bilo koju stranicu koja koristi *Koha* sustav za *online* knjižnice. Kao ulaz uzima frazu, a kao izlaz

vraća knjige i članke. URL konkretne stranice koju korisnik želi pretražiti se zadaje posebno kao parametar. Taj sustav primjerice koriste i knjižnica FER-a (*http://lib.fer.hr/*) i FFZG-a (*https://koha.ffzg.hr/*).

- *PhraseToArticleGS* - transformacija koja pretražuje popularni *Google Scholar* (*https://scholar.google.com/*) servis za članke vezane uz ulaznu frazu.
- *PhraseToArticleIEEE* - transformacija koja pretražuje *IEEE Xplore* (*http://ieee.org/ieeexplore*) digitalnu knjižnicu za članke vezane uz ulaznu frazu.
- *PhraseToConferenceCA* - transformacija koja pretražuje servis *Conference Alerts* (*http://www.conferencealerts.com/*) za konferencije vezane uz ulaznu frazu.
- *BookArticleToPerson* - transformacija koja iz entiteta *Book* ili *Article* izvlači informacije o autorima iz njihove stavke u kojoj su autori zapisani i vraća ih u obliku entiteta *Person*.
- *PersonToUrlSE* - transformacija koja uzima entitet *Person* i vraća URL-ove dobivene pretraživanjem *Google* tražilice s punim imenom osobe kao ulaznom frazom.

Sve transformacije uz ulazni entitet mogu primati i parametre koje zadaje korisnik. Primjerice, gotovo svaka transformacija ima parametar koji zadaje koliko rezultata korisnik želi, dok transformacije koje vraćaju knjige i članke imaju još i parametre koji ograničavaju raspon godina unutar kojih su objavljeni vraćeni rezultati.

6.3. Primjer scenarija korištenja

Jednom kada su entiteti i transformacije razvijeni i integrirani u Netglub sustav, konačno rješenje je moguće ovako koristiti kroz klijentski program:

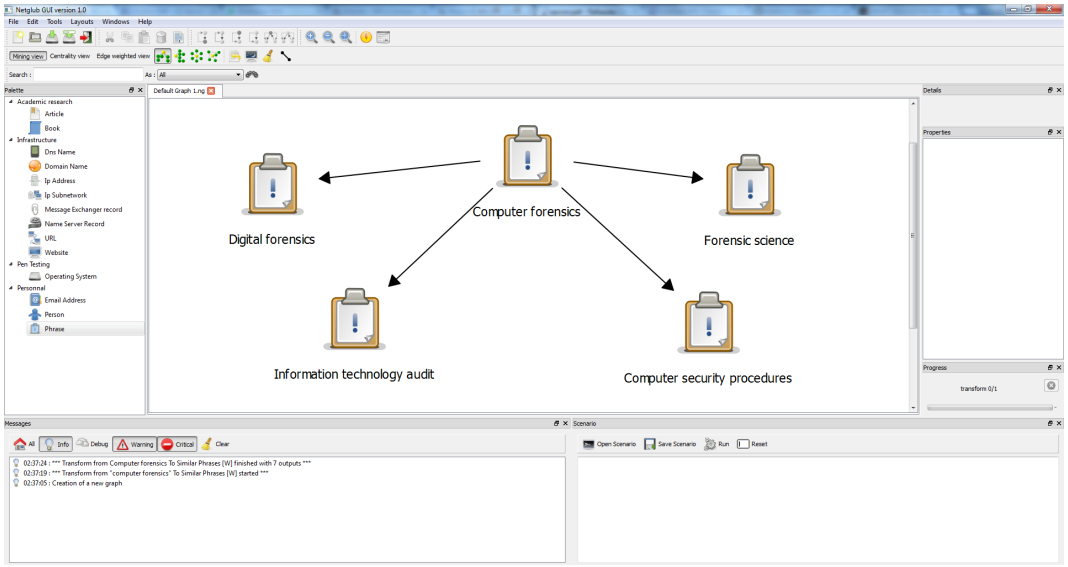
1. Zadaju se početne fraze koje opisuju područje interesa.
2. Nad početnim frazama se pokrene transformacija *PhraseToPhraseW* koja generira slične fraze. Od dobivenih fraza izbrišu se one koje nisu povezane uz područje interesa. Slika 6.1 prikazuje graf s jednom početnom frazom i sličnim frazama generiranima ovom transformacijom iz nje.
3. Nad preostalim frazama pokrenu se transformacije koje primaju frazu kao ulaz. Obično se pokrenu sve takve transformacije, no ovisno o preferencijama, moguće je i pokrenuti samo dio njih. Te transformacije sada vraćaju entitete knjiga, članaka i konferencija. Slika 6.2 prikazuje rezultat pokretanja tih transformacija

nad frazama iz slike 6.1. Na slici crvenim čvorovima odgovaraju fraze, zelenim knjige, žutim članci te plavim konferencije.

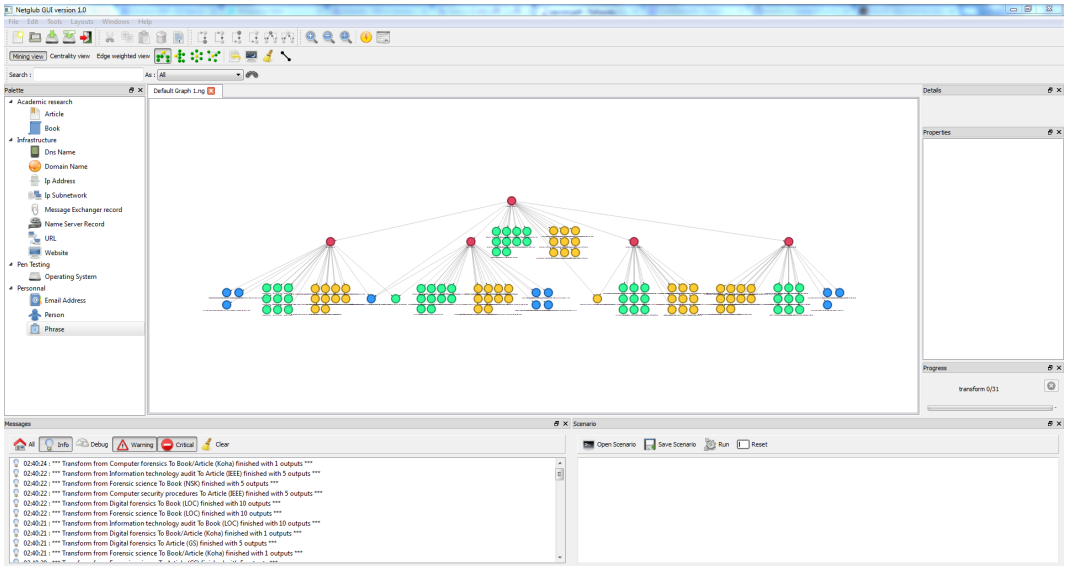
4. Nad dobivenim knjigama i člancima se pokrene transformacija *BookArticleToPerson*. Kao rezultat su dobiveni autori svih knjiga i članaka s grafa.

5. U ovome trenutku se obično uključi opcija koja skalira veličinu entiteta ovisno o broju ulaznih grana. Time će se postići da autori koji su sudjelovali u pisanju većeg broja knjiga i članaka budu vizualno veći. Ovakvu vizualno informaciju nije moguće dobiti klasičnim postupkom istraživanja i dio je prednosti ovakvog načina istraživanja. Slika 6.3 prikazuje rezultat pokretanja transformacije *BookArticleToPerson* nad entitetima knjiga i članaka sa slike 6.2 te uključivanje opcije skaliranja veličine entiteta ovisno o broju ulaznih grana. Na slici ljubičastim čvorovima odgovaraju autori, te veći ljubičasti čvorovi prikazuju autore koji su sudjelovali u pisanju većeg broja knjiga i članaka od drugih autora.

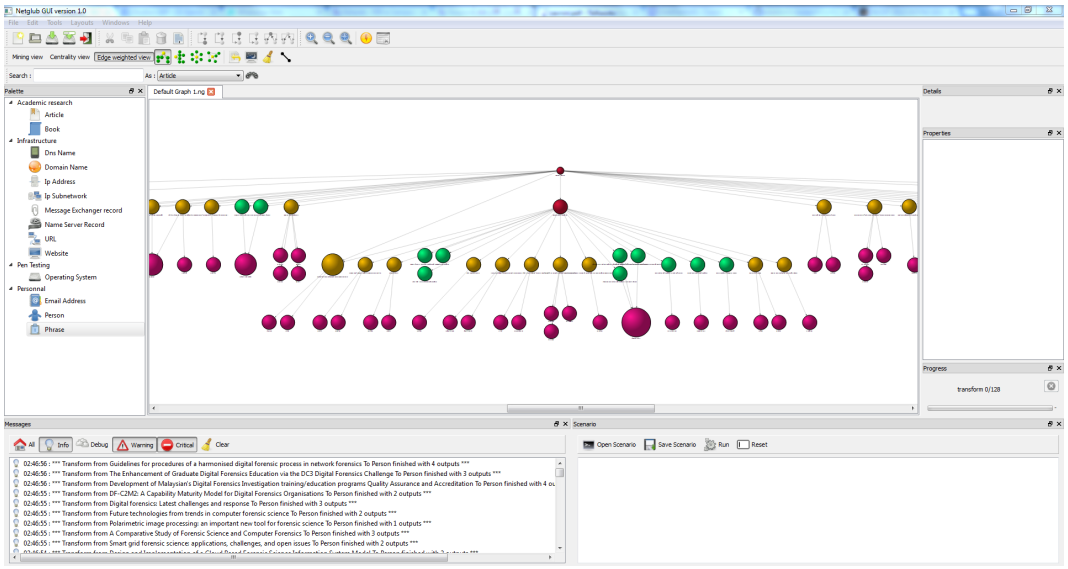
6. Na kraju, dodatni korak bi bio još povezati autore u cjeline. Moguće je koristiti transformacije nad dobivenim entitetima autora (vrsta *Person*) kako bi se dobila nekakva ideja o tome kojoj organizaciji oni pripadaju. Zatim, na isti način kao i autori, organizacije kojima više autora pripada će biti vizualno veće i tako se isticati. Može se čak i uključiti opcija koja povećava entitete ovisno o broju izlaznih grana — tada će autori koji pripadaju većem broju organizacija biti vizualno veći.



Slika 6.1: Netglub graf s početnom zadanom frazom i par generiranih fraza



Slika 6.2: Netglub graf s frazama te knjigama, člancima i konferencijama generiranima pomoću njih



Slika 6.3: Netglub graf s frazama, knjigama, člancima, konferencijama i autorima uz uključeno skaliranje entiteta ovisno o broju ulaznih grana

Koristeći Netglubove mogućnosti skriptiranja, i ovaj cijeli postupak je moguće automatizirati. Uz tu mogućnost je moguće lako pokrenuti isti postupak za drugo područje, ili čak kasnije za isto, ukoliko informacije više nisu ažurne.

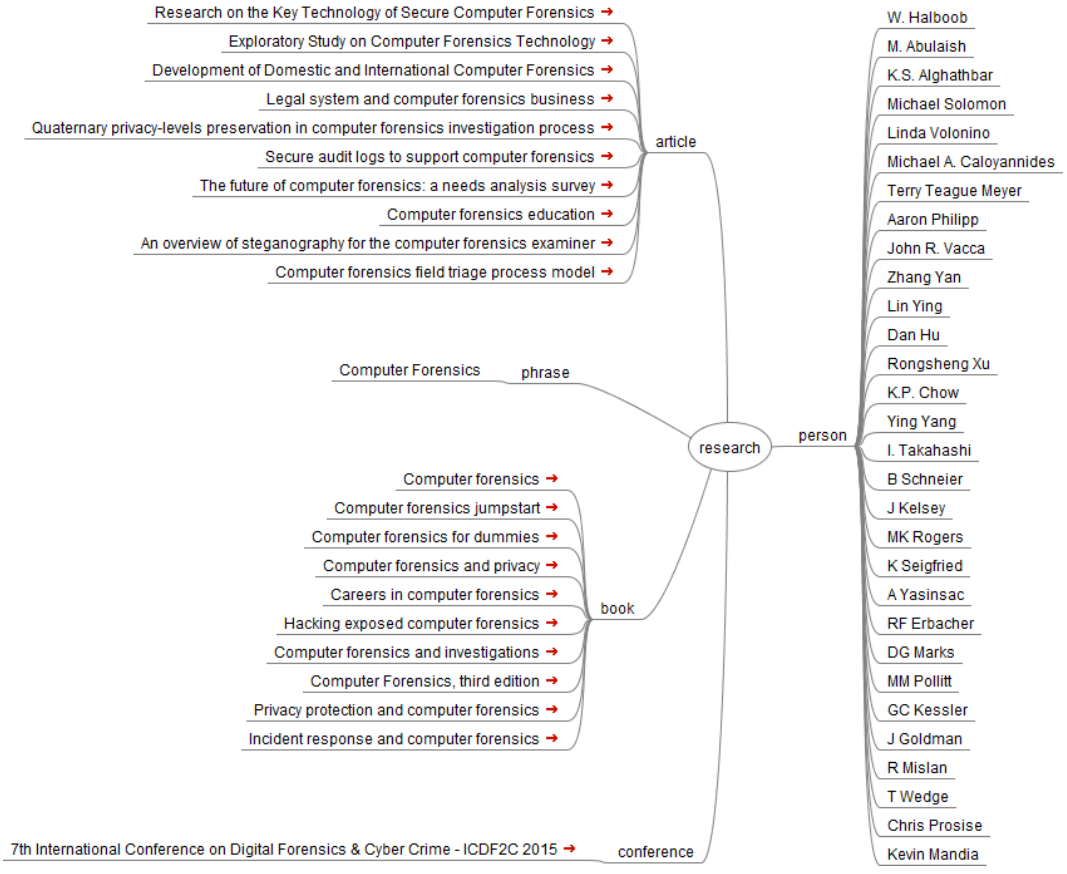
6.4. Pretvorba rezultata u umnu mapu

Netglub omogućuje pohranu konačnog rezultata u formatima prikladnima za ponovno otvaranje u Netglubu ili Maltegu.

U sklopu ovog rada razvijen je i alat koji pretvara rezultate iz Netglub (.ntg) formata u Freemind (.mm) format za umne mape. Pretvorba grupira entitete po vrsti, te će prilikom otvaranja umne mape u prikladnom alatu biti moguće jednostavnim klikom na entitet otvoriti odgovarajući URL u pregledniku.

Konačan rezultat pretvorbe je pogodniji za krajnjeg korisnika i za daljnje pretvaranje u druge formate, primjerice HTML. Nedostatak takvog prikaza rezultata je gubitak veza među entitetima.

Slika 6.4 prikazuje rezultat pretvorbe jednostavnog Netglub grafa u Freemind umnu mapu koristeći razvijeni alat.



Slika 6.4: Umna mapa dobivena pretvorbom Netglub grafa pomoću razvijenog alata

6.5. Postavljanje sustava, distribucija alata i sigurnosni aspekti

Za distribuciju i šire korištenje konačne verzije razvijenog rješenja potrebno je postaviti Netglub gospodara i jednog ili više sluga na jedno ili više poslužiteljskih računala te distribuirati klijentski program svima koji žele koristiti sustav.

Programi Netglub gospodara i sluge su prilagođeni izvođenju na računalima s GNU/Linux operacijskim sustavom. Uz njih je za rad gospodara potrebno postaviti i MySQL bazu podataka u kojoj su zapisani podaci o entitetima i transformacijama.

Klijentski program je moguće prevesti za GNU/Linux, Windows i (Mac) OS X operacijske sustave. Krajnji korisnici trebaju pokrenuti samo klijentski program kako bi provodili istraživanje pomoću ovog sustava. S ova tri operacijska sustava su pokriveni gotovo svi korisnici stolnih i prijenosnih računala.

Uvijek kada se postavljaju poslužitelji koji obrađuju mrežne zahtjeve, u ovom slučaju gospodar i sluge, potrebno se zapitati o sigurnosnim aspektima rješenja. Postoji nekoliko aspekata na koje je potrebno obratiti pozornost:

1. Programi gospodara, sluge i klijenta nisu detaljno istraženi imaju li sigurnosne propuste (engl. *vulnerabilities*), tako da samo njihovo korištenje je već potencijalni rizik.

2. U konfiguracijskoj datoteci gospodara, postoji opcija koja se odnosi na IP adresu (te time i sučelje) na kojoj bi on trebao očekivati zahtjeve. Ta IP adresa se odnosi i na zahtjeve od sluga, i od klijenata. Zbog toga nije moguće odvojiti sučelje na kojemu gospodar komunicira sa slugama i sučelje na kojemu komunicira s klijentima. Primjerice, ukoliko su gospodar i sluge pokrenuti na istom poslužiteljskom računalu te je potrebno da se klijenti mogu spojiti na gospodara od bilo kuda, bilo bi poželjno da gospodar komunicira sa slugama preko lokalnog (*loopback*) sučelja, a s klijentima preko pravog mrežnog sučelja. Ovakvim mogućnostima upravljanja trenutno to nije moguće. I dalje je potrebno podesiti koja TCP vrata (engl. *port*) gospodar koristi za komunikaciju sa slugama, a koja za komunikaciju s klijentima, tako da ih je moguće razlikovati kod postavljanja vatrozida (engl. *firewall*).

3. Gospodar naizgled podržava autorizaciju sluga pomoću TLS certifikata, no ta mogućnost zapravo nije implementirana, već postoji samo njen „kostur“.

4. Potrebno je posebno obratiti pozornost na to da je konfiguracijski direktorij

/etc/netglub zaštićen, tj. da mu mogu pristupiti isključivo programi kojima je to potrebno, a to su programi gospodara i sluge. U tom direktoriju se nalazi konfiguracijska datoteka za gospodara u kojoj je zapisano korisničko ime i lozinka za bazu podataka koju on koristi te privatni ključevi certifikata.

Drugu i treću činjenicu bi zločudna stranka mogla iskoristiti tako da na gospodara spoji svog vlastitog slugu i time izvršava dio transformacija koje su korisnici zahtjevali. Tada bi bilo moguće promijeniti samo izvršavanje transformacija i vratiti proizvoljne rezultate umjesto onih koje korisnik očekuje. Primjerice, ukoliko jedna takva zloćudna transformacija vrati 10000 knjiga umjesto 10 koje je korisnik zatražio, korisnikov klijentski program će pokušati nacrtati tih 10000 knjiga na grafu te će vrlo vjerojatno zablokirati.

U svrhu sprječavanja takvog scenarija je potrebno ograničiti koji se sluge mogu spojiti na gospodara. Autorizacija sluga nije implementirana te gospodar komunicira sa slugama na istom sučelju kao i s korisnicima, tako da je potrebno potražiti dodatno rješenje. Jedno rješenje tog problema je postavljanje vatrozida koji će ograničavati od kuda se sve mogu spajati sluge i klijenti. To je moguće napraviti jer se sluge i klijenti spajaju na različita TCP vrata, te se taj kriterij zatim može koristiti u postavkama vatrozida.

Uz to, općenito je pametno ograničiti od kuda se sve klijenti i sluge mogu spojiti na gospodara jer je to mrežni servis u čiju sigurnost bez detaljnijeg istraživanja ne treba imati puno povjerenja. U praksi je često moguće ograničiti od kuda se spajaju sluge samo na primjerice lokalnu mrežu. Isto tako, ako je postavljeni sustav namijenjen posluživanju korisnika samo na lokalnoj ili nekoj užoj mreži od cijelog Interneta, moguće je postaviti vatrozid da ograničava spajanje klijenata na samo tu mrežu.

Što se tiče četvrtog aspekta, potrebno je jednostavno provjeriti imaju li dopuštenja za pristup datotekama u direktoriju /etc/netglub isključivo oni koji ih zaista i koriste.

6.6. Nedostaci ovakvog pristupa

Koliko god ovakav pristup istraživanju i mapiranju tematskih područja bio dobar, kroz razvijanje rješenja identificirani su i neki nedostaci.

Što se tiče vremena izvođenja, postoje dva problema.

Prvi se temelji na tome da za svaku transformaciju nad ulaznim entitetom sluga mora kreirati novi proces. Čak i za jednostavnije transformacije kao što je transformacija koja čita stavku autora iz entiteta knjiga i članaka i vraća entitete autora to je

potrebno — a takve transformacije se ponekad pozivaju na velikom broju entiteta, u ovom slučaju nad svim knjigama i člancima s grafa. Primjerice, transformacija koristi 10MB radne memorije (u praksi realno kada je transformacija Python skripta) i ne izvršava se izrazito brzo. Ukoliko je ta transformacija pozvana nad 100 ulaznih entiteta, biti će paralelno pokrenuto 100 procesa te transformacije koji će sveukupno koristiti 1GB radne memorije, što može biti problem ovisno o specifikacijama računala na kojemu je sluga pokrenut.

Drugi problem je crtanje velikog broja entiteta na klijentu. Ukoliko se na grafu nalazi velik broj entiteta što je realno u praksi, klijentski program može raditi prilično sporo zbog kompleksnosti crtanja grafa s toliko čvorova.

Kod transformacije koja pretražuje *Google Scholar*, potrebno je posebno pripaziti na broj zahtjeva koji se šalje. Ukoliko se pošalje prevelik broj zahtjeva, moguće je da će servis prepoznati transformaciju kao program koji automatski šalje zahtjeve (što i je) te zatražiti ga da riješi izazov kao primjerice *CAPTCHA*[1]. Neovisno o tome blokira li servis transformaciju nakon previše poslanih zahtjeva ili ne, poželjno je uvijek pripaziti na broj poslanih zahtjeva. Ukoliko korisnik primjerice traži knjige o računalnoj forenzici iz 2015. godine i želi 20 rezultata, takva potraga možda nikada ne bi završila jer toliko knjiga takvih knjiga možda ni ne postoji. Zato je potrebno postaviti neku granicu kod pretraživanja kako bismo na kraju krajeva bili pristojni korisnici servisa kojeg koristimo.

Također, ukoliko servis koji transformacija koristi ne nudi nikakav API, obično je potrebno ručno izvući strukture informacija iz stranice. Ukoliko se stranica u nekom trenutku značajnije promijeni, moguće je da transformacije više neće funkcionirati. Ta činjenica znatno otežava dugoročno održavanje takvih transformacija.

Osim toga, nije još pronađeno dobro rješenje za povezivanje osoba u smislene cjeline (primjerice organizacije). Isprobana su dva pristupa.

Prvi pristup je bio koristeći transformaciju koja vraća URL-ove povezane s osobom. Ti URL-ovi su zatim pretvoreni u entitete *Website*, koji uzmu samo bazu URL-a (bez puta do same datoteke) — primjerice iz *http://www.fer.unizg.hr/predmet/zavrad* uzmu samo *http://www.fer.unizg.hr*. Ideja je bila da će tim pristupom autori konvergirati oko nekih stranica, primjerice stranica fakulteta i tvrtki.

No za sada s prilično jednostavnom transformacijom za dobivanje URL-ova iz entiteta osobe to nije moguće. Ta transformacija jednostavno uzme prvih nekoliko URL-ova pretražujući *Google* imenom osobe. Krajnji rezultat je bila konvergencija oko op-

[1]CAPTCHA - izazov posebno namijenjen za razlikovanje ljudi i programa, najčešće radi tako da korisniku prikaže sliku iskrivljenih slova i znamenaka koje on zatim mora ispravno prepoznati.

ćenitih stranica koje agregiraju informacije o osobama. Dakle sve od velikih stranica kao što su *Facebook* i *LinkedIn*, do manjih, specijeliziranijih stranica koje su posvećene akademskom istraživanju. No moguće je da bi ovaj pristup mogao biti dobar s nešto drugačijim načinom dobivanja URL-ova od osoba.

Drugim pristupom je pokušan nešto drugačiji način dobivanja URL-ova od osoba. Korištena je transformacija koja iz osobe nalazi e-mail adrese, te zatim druga koja iz tih e-mail adresa izvlači domenu. Cilj tog pristupa je zaobilaženje stranica kao što su *Facebook* i *LinkedIn* jer one ne pružaju e-mail servis. Ideja je bila da će, primjerice ako je osoba profesor s FER-a, dobiti e-mail adresu oblika *ime.prezime@fer.hr*. Jednom kada su sakupljene takve e-mail adrese, lagano bi bilo izvući domene iz njih, te bi se onda oko njih trebali smisleno grupirati autori.

Problem u ovom pristupu je bila težina pronalaženja e-mail adresa za dovoljno velik broj osoba. Postojeća transformacija koja to radi praktički nije uopće funkcionirala. Uz nekoliko modifikacija, uspješno je dovedena do stanja da za neke osobe pronalazi e-mail adrese, no i to nije bilo dovoljno. Ovakvim pristupom nije dobiveno niti blizu dovoljno entiteta za bilo kakve vizualne informacije.

Zaključno, ovi problemi su najviše stvar implementacije i smišljanja ideja o tome kako iskoristiti podatke koji su dostupni u javnim izvorima. Uz još vremena za implementaciju i istraživanje, za navedene probleme je moguće naći dobra rješenja.

7. Daljnji rad

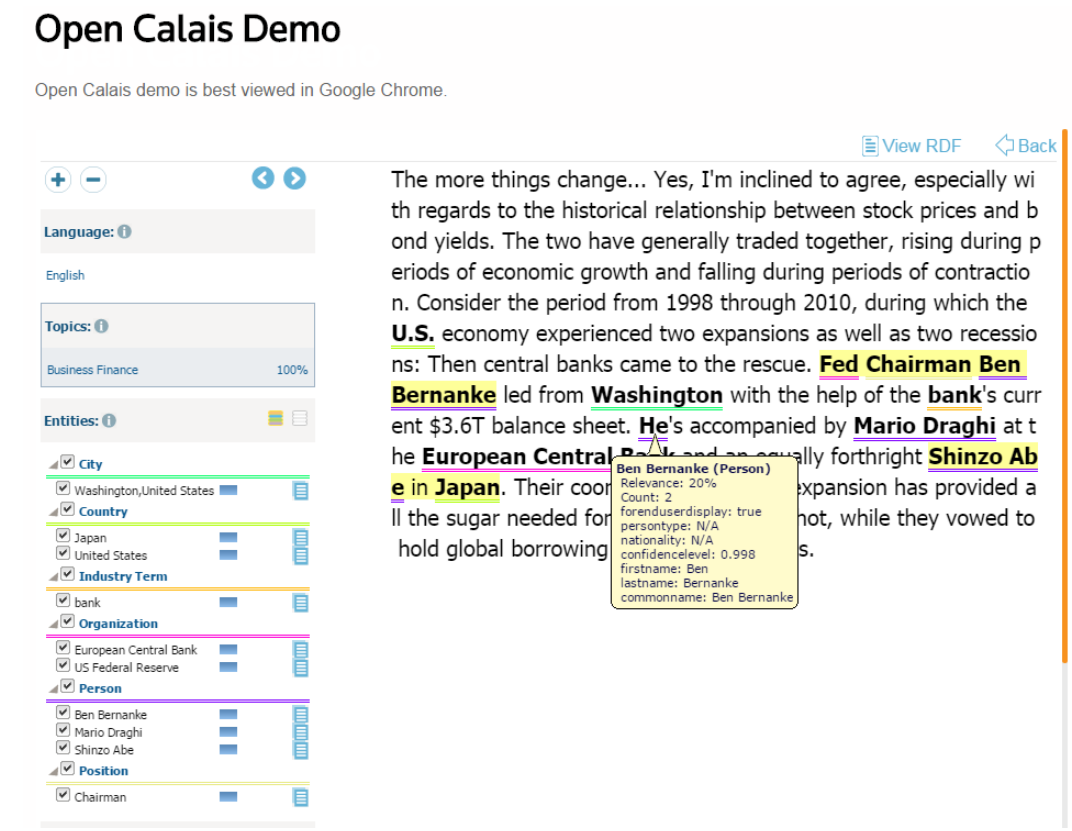
Razvijeno rješenje pokazuje osnove onoga što je moguće napraviti s ovakvim pristupom. Za bolje mapiranje i općenitiju primjenu, potrebno je istražiti na koje je još načine moguće iskoristiti javno dostupne podatke.

Jedan daljnji pristup uključuje korištenje servisa obrade prirodnog jezika (engl. *natural language processing*). Primjeri takvih servisa su *AlchemyAPI*[1] i *OpenCalais*[7]. Oni omogućuju korisniku da kao ulaz zada tekst ili web stranicu. Zatim servis obradi ulaz i vrati korisniku informacije o entitetima i vezama u strukturiranom obliku. Slika 7.1 prikazuje demonstraciju analize teksta pomoću servisa Open Calais. Transformacije koje koriste takve servise bi se mogle koristiti kao alternativni put od početnih fraza do entiteta koji nas zanimaju. Prvo bi se koristile transformacije koje pretvaraju fraze u URL-ove, zatim bi ti URL-ove bili analizirani pomoću navedenih servisa. Time primjerice nema više ograničenja isključivo na autore knjiga i članaka, već je moguće dati informacije o osobama i ostalim entitetima koji su bilo kako povezani s početnom frazom. Takav pristup bi bio posebno koristan kada područje od interesa ne postoji nužno u akademskom kontekstu, tj. ne se može kvalitetno istražiti putem knjiga, članaka i sl.

Neovisno o samom postupku i implementaciji istraživanja, treba istražiti različite načine prikazivanja rezultata krajnjem korisniku. Konačni prikaz unutar Netglub klijenta je dobar, no ne uvijek i najpraktičniji. Jedno rješenje je već predloženo i u osnovi implementirano — pretvorba rezultata iz Netglub formata u Freemind format umne mape. Pretvorba u taj format otvara daljnja vrata. Postoje već alati koji pretvaraju Freemind umne mape u HTML dokumente što je puno praktičnije od originalnog formata. Takav dokument svaki Internet preglednik može otvoriti te je iznimno jednostavno ga integrirati u postojeće web stranice.

Također, treba razmisliti o samom klijentu. Netglub klijentski program je dobro rješenje, no još pristupačnije rješenje bi bilo implementirati klijenta unutar Internet preglednika. Implementacija klijenta i pohrana konačnog rezultata u obliku prikladnom Internet pregledniku bi učinila ovaj način istraživanja puno pristupačnijim široj

javnosti.



Slika 7.1: Open Calais demonstracijska analiza paragrafa teksta

8. Zaključak

Ovakav pristup istraživanju i mapiranju područja ima velik broj prednosti nad dosadašnjim pristupima. Ovim postupkom je iz ulaznih fraza dobiven vizualni prikaz područja kroz entitete i njihove veze. Te veze je moguće dalje iskoristiti kako bi se istaknuli entiteti koji imaju veći broj ulaznih ili izlaznih veza, ovisno o tome koja je interpretacija potrebna.

Uz to, osim što nije potrebno ručno pretraživati različite baze, čak je i cijeli ovaj postupak moguće automatizirati pomoću skriptiranja. To omogućava lako ponavljanje istraživanja za druga područja ili ažuriranje rezultata već istraženog područja.

Iako razvijeno rješenje nije sveobuhvatno, ono je dobar prikaz mogućnosti ovakvog pristupa. Daljnjim istraživanjem, razvojem i integriranjem već dostupnih servisa kao što su npr. servisi obrade prirodnog jezika u proces je moguće postići kvalitetno i automatsko mapiranje tematskih područja.

LITERATURA

[1] Alchemyapi. URL http://www.alchemyapi.com/.

[2] Freemind. URL http://freemind.sourceforge.net/wiki/index.php/Main_Page.

[3] Maltego, . URL https://www.paterva.com/web6/products/maltego.php.

[4] Maltego - downloads, . URL https://www.paterva.com/web6/products/download2.php.

[5] Maltego version 3 user guide, . URL http://www.paterva.com/web6/documentation/M3GuideGUI.pdf.

[6] Netglub. URL http://www.netglub.org/.

[7] Open calais. URL http://new.opencalais.com/.

[8] Intelligence collection disciplines (ints). URL http://www.fbi.gov/about-us/intelligence/disciplines.

[9] Zotero. URL https://www.zotero.org/.

[10] Jiahui Liu, Peter Jin Hong, i Elin Rønby Pedersen. Research trails: getting back where you left off. U *Proceedings of the 19th international conference on World wide web*, stranice 1151–1152. ACM, 2010.

[11] Elin Rønby Pedersen, Karl Gyllstrom, Shengyin Gu, i Peter Jin Hong. Automatic generation of research trails in web history. U *Proceedings of the 15th international conference on Intelligent user interfaces*, stranice 369–372. ACM, 2010.

Dodatak A
Primjer razvijenog entiteta

Navedena je XML datoteka *desc.xml* koja definira razvijeni entitet *Article*. Entitet *Article* opisuje jedan članak pomoću obveznih stavki za naslov (*Title*) i autore (*Authors*) te neobaveznih stavki za godinu publikacije (*Date*) i URL članka ili stranice koja opisuje članak (*URL*).

desc.xml:

```
<?xml version='1.0'?>
<!DOCTYPE EntitySchema>
<entity name="article" longName="Article" color="#ffc000"
    pngImage="image.png" svgImage="image.svg">
  <description>article</description>
  <fieldList>
    <field name="value" longName="Title" description="
       title" default="" optional="false" level="default"
        format="string" ioMode="input">
      <string regex=".*"/>
    </field>
    <field name="authors" longName="Authors" description
       ="authors" default="" optional="false" level="
       advanced" format="string" ioMode="input">
      <string regex=".*"/>
    </field>
    <field name="date" longName="Date" description="date"
        default="" optional="true" level="advanced"
       format="string" ioMode="input">
      <string regex=".*"/>
    </field>
```

```
    <field name="url" longName="URL" description="URL"
       default="" optional="true" level="advanced" format
       ="string" ioMode="input">
      <string regex=".*"/>
    </field>
  </fieldList>
</entity>
```

Dodatak B
Primjer razvijene transformacije

Navedena je XML datoteka *conf.xml* koja opisuje transformaciju i Python skripta *transform* koja izvršava transformaciju *PhraseToBookLOC*. Ona prima frazu kao ulaz te vraća knjige s *Library of Congress* web stranice (*http://loc.gov/*) kao izlaz. Kao parametre od korisnika prima broj rezultata (*numResults*) i raspon godina unutar kojih je knjiga izdana (*minYear* i *maxYear*).

conf.xml:

```
<?xml version='1.0'?>
<!DOCTYPE TransformSchema>
<transform name="PhraseToBookLOC" longName="To Book (LOC)
   " type="generic" >
  <description>find books from phrase using library of
     congress</description>
  <parameters>
    <param name="numResults" longName="Number of results
       desired" description="number of result asked"
       default="" optional="true" level="advanced" format
       ="int">
      <int min="1" />
    </param>
    <param name="minYear" longName="lower bound for year
       range (0 = no limit)" description="lower bound for
        year range (0 = no limit)" default="" optional="
       true" level="advanced" format="int">
      <int min="0" max="10000" />
    </param>
    <param name="maxYear" longName="upper bound for year
```

```
        range (0 = no limit)" description="upper bound for
         year range (0 = no limit)" default="" optional="
        true" level="advanced" format="int">
       <int min="0" max="10000" />
     </param>
   </parameters>
   <config>
     <value name="need_root">true</value>
     <value name="input_as_args">true</value>
   </config>
   <input>
     <entity type="phrase" />
   </input>
   <output>
     <entity type="book" />
   </output>
</transform>
```

transform:

```
#!/usr/bin/python

import requests
from bs4 import BeautifulSoup
from generic_transform import *

def year_in_bounds(year, minYear, maxYear):
  try:
    year = int(year)
  except:
    year = 0
  if minYear and year < minYear:
    return False
  if maxYear and year > maxYear:
    return False

  return True
```

```
e_type,e_values,params = get_from_args()

query = e_values["value"]

numResults = 10
minYear = 0
maxYear = 0

if "numResults" in params:
  numResults = int(params["numResults"])
if "minYear" in params:
  minYear = int(params["minYear"])
if "maxYear" in params:
  maxYear = int(params["maxYear"])

par = {"q": query, "fa": "original-format:book", "all": "
   true", "c": "50"}

page_counter = 1
while page_counter <= 4:
  par["sp"] = page_counter
  req = requests.get("http://www.loc.gov/search/", params
      = par)
  soup = BeautifulSoup(req.text)

  for parent_tag in soup.find_all(class_ = "description")
     :
    title_tag = parent_tag.h2.a

    title_string = title_tag.text.strip()
    while not title_string[-1:].isalnum():
      title_string = title_string[:-1]

    url_string = "http:" + title_tag["href"]
```

```
    authors_tag = parent_tag.find(class_ = "contributor")

    if authors_tag is not None:
      authors_string = authors_tag.span.text
    else:
      authors_string = ""

    date_tag = parent_tag.find(class_ = "date")

    if date_tag is not None:
      date_string = date_tag.span.text
    else:
      date_string = ""

    if not year_in_bounds(date_string, minYear, maxYear):
      continue

    write_result("book", {"value": title_string, "authors
       ": authors_string, "date": date_string, "url":
       url_string})
    numResults -= 1

    if numResults == 0:
      sys.exit(0)

  page_counter += 1
```

Programski alati za automatsko mapiranje tematskih područja

Sažetak

U ovom radu je identificirana potreba za brzim obuhvaćanjem temeljnih informacija o nekom tematskom području te nedostatak postojećih programskih alata koji to omogućavaju. U sklopu rada je razvijen programski alat koji automatski prikuplja podatke o tematskom području te grafički prikazuje međuzavisnosti na način prilagođen korisniku za daljnju analizu. Razvijeni alat se temelji na nadogradnji alata za prikupljanje i vizualizaciju javno dostupnih podataka pod imenom Netglub. Konačno rješenje predstavlja novi pristup istraživanju tematskih područja koji ima velik broj prednosti nad dosadašnjim pristupima. Iako razvijeni alat nije sveobuhvatan, on dobro prikazuje mogućnosti korištenog pristupa istraživanju. Uz dodatno istraživanje, razvoj i integriranje već dostupnih servisa kao što su servisi obrade prirodnog jezika moguće je postići još kvalitetnije i efikasnije automatsko mapiranje tematskih područja.

Ključne riječi: automatsko istraživanje, automatsko mapiranje, javno dostupni podaci, OSINT, vizualizacija pretraživanja informacija

Software tools for automatic mapping of subject areas

Abstract

In this paper, the need for quick acquisition of basic information for a given subject area and the lack of existing tools that would satisfy this need was identified. A software tool that automatically collects general information about a given subject area and graphically shows dependencies between elements of obtained information in a manner suitable to its user for further analysis was developed. The developed tool is based on upgrading the existing open-source intelligence gathering and visualisation tool Netglub. The developed solution represents a new approach to subject area research which has a large number of advantages to current approaches. Even though the developed tool is not all-encompassing, it shows the possibilities of the used approach well. With additional research, development and integration of available services such as natural language processing services into the tool, it is possible to achieve even better and more efficient automatic mapping of subject areas.

Keywords: automatic research, automatic mapping, open-source intelligence, OSINT, information search visualization