

SNIK Ontologie—Lehre und Implementierung

mit nachträglichen Anmerkungen vom 10. Oktober 2016

Konrad Höffner

`konrad.hoeffner@imise.uni-leipzig.de`

5. Oktober 2016

Vorstellung

- ▶ Konrad Höffner
- ▶ Studium Diplominformatik an Uni Leipzig
- ▶ Doktorand der Informatik beim AKSW, Uni Leipzig/InfAI
- ▶ Thema „Question Answering auf RDF Data Cubes“
- ▶ bei IMISE und im SNIK Projekt seit Juli
- ▶ kein Vorwissen über Medizin aber viel praktische Erfahrung mit Semantic Web-Technologien

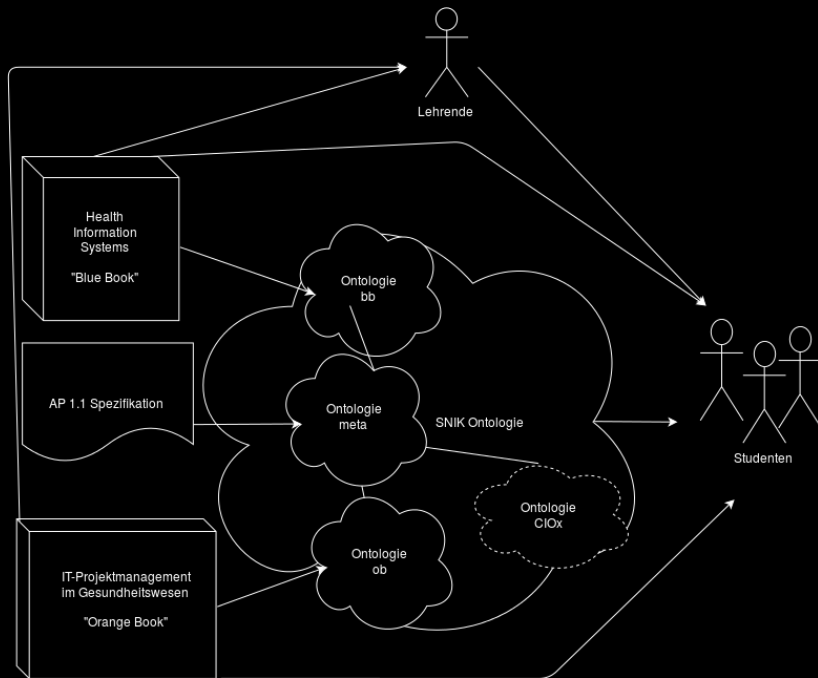
Vorstellung

- ▶ Visualisierung, Implementierung, Serialisierung
- ▶ Qualitätssicherung
- ▶ Aufsetzen von Services

- ▶ Raum 227, Tel. (0341)97-16363
- ▶ konrad.hoeffner@imise.uni-leipzig.de
- ▶ <https://github.com/KonradHoeffner/latex/tree/master/beamer/2016/snik-projekttreffen>

Section 1

Einsatz in der Lehre



Ziele

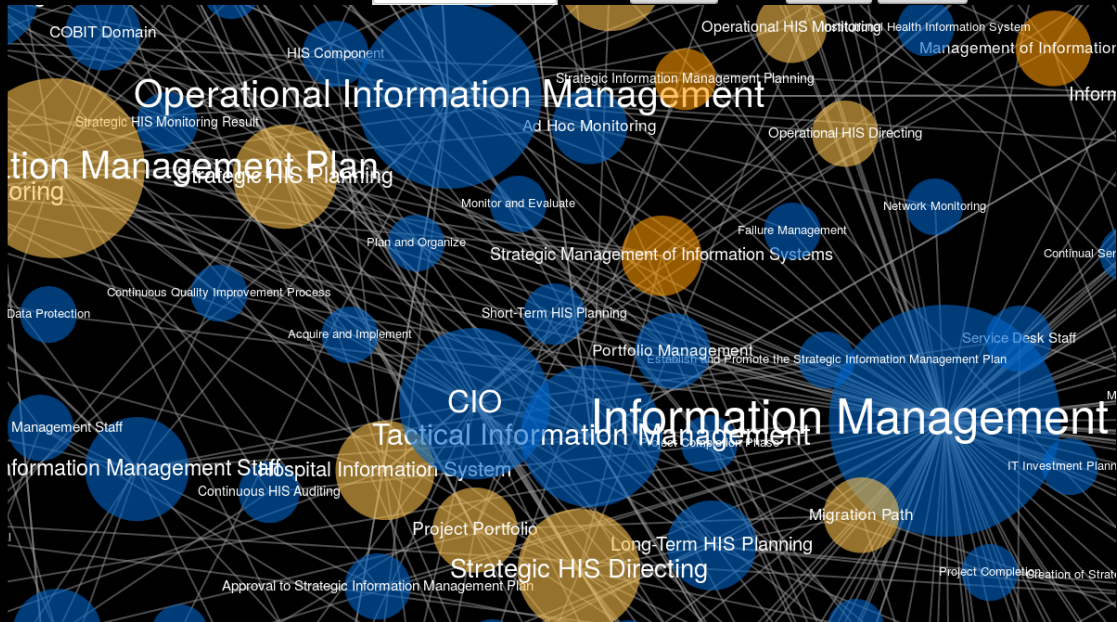
- ▶ modelliertes Wissen vermitteln, zusätzlich zu Lehrbüchern, Vorlesungen und Übungen
- ▶ Exploration
- ▶ Erstellen von Übungsaufgaben
- ▶ Semantic Web nur Mittel zum Zweck, so viel Zeit wie möglich für Gesundheitssysteme

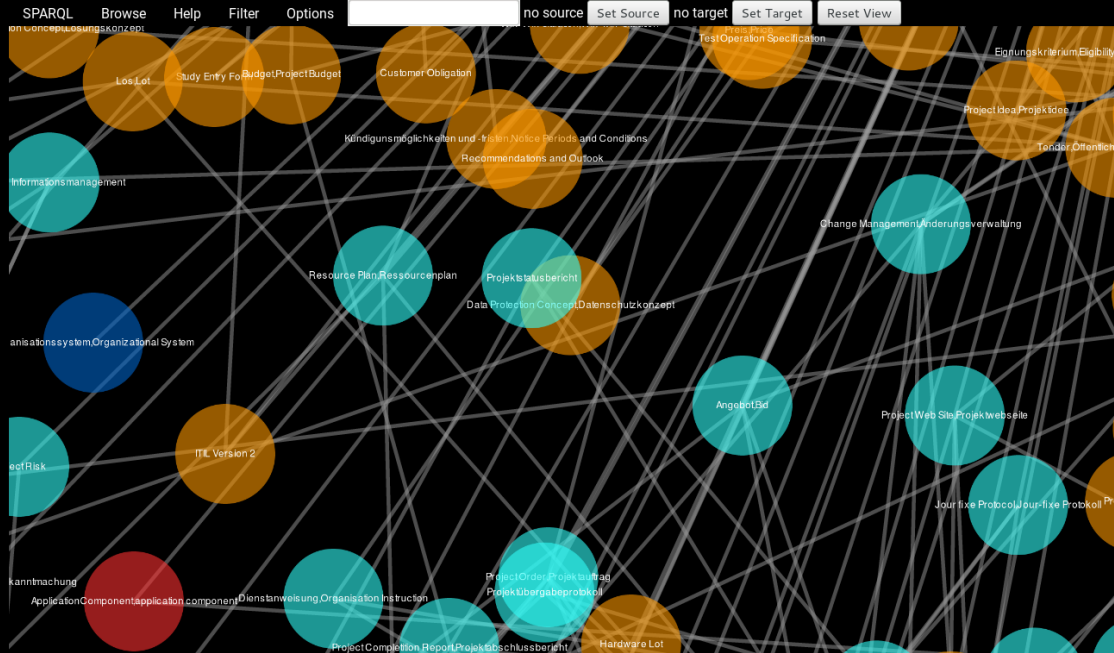
Problem

- ▶ Studenten sind zwar (Medizin-)Informatiker, haben aber nicht zwangsweise die Semantic Web Vorlesungen von Prof. Fährlich besucht
- ▶ → kein Vorwissen in SPARQL und RDF-Serialisierungsformaten voraussetzen
- ▶ Protégé kein intuitiver Gesamtüberblick, getestete Graphplugins skalieren nicht
- ▶ Lösung: Eigenentwicklung einer Visualisierung unter Verwendung ausgereifter Bibliotheken

[http://www.snik.eu/\(p\)graph/](http://www.snik.eu/(p)graph/)

- ▶ Öffentliche alte Version (wird aktualisiert) ohne CIOx
<http://www.snik.eu/graph/>
- ▶ Passwortgeschützte neue Version mit CIOx
<http://www.snik.eu/pgraph/>
- ▶ CIOx-Ontologie enthält Betriebsinterna, Zugangsdaten
nur auf sichere Art und Weise an Befugte Weitergeben!





Kürzester Weg

Computer-Based Information System

Tool

System

Application System

Information System

Professional

Information System

isDecomposed

updates

Organiz

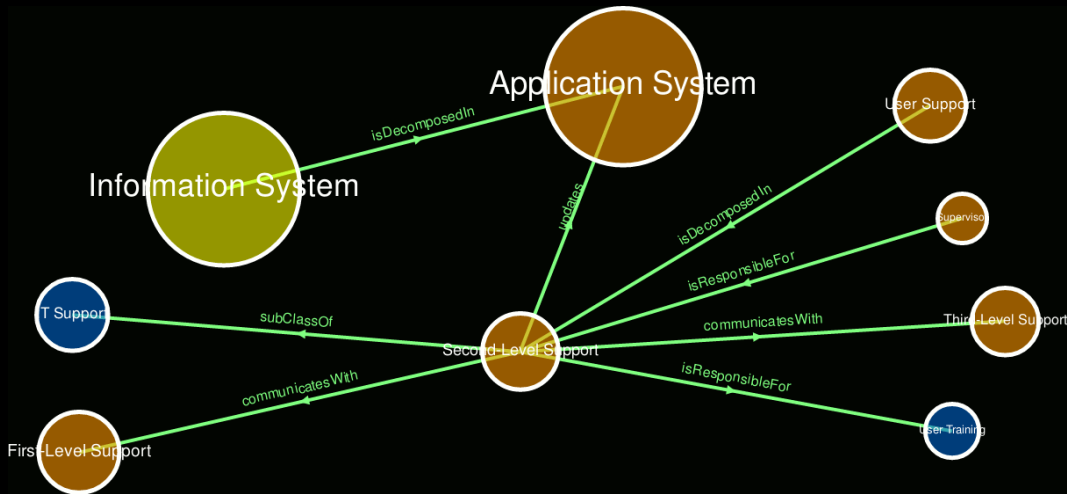
Software Reference Model

Second-Level Support

SWOT Analysis

Installation

Spiderworm



Praktische Vorführung

- ▶ Kürzester Weg und Spiderworm
- ▶ Suche
- ▶ Filterung
- ▶ Hilfe
- ▶ Feedback
- ▶ Browse

Section 2

Implementierung

Modellierung und Serialisierung

Überblick

Ontologie	Anzahl Tripel
meta	244
bb	35803
links-bb	79
ciox	1933
links-ciox	29
ob	25894
gesamt	63982

Modellierung und Serialisierung

Überblick¹

Oberklasse	Anzahl
meta:Role	79
meta:Function	154
meta:EntityType	1395

¹Untere Schranken, noch nicht alle meta:subTop-Beziehungen modelliert.

Modellierung und Serialisierung

Ausgangssituation

- ▶ SNIK-Ontologien bb, ob und ciox wurden mit Protégé bearbeitet und als RDF/XML serialisiert
- ▶ bei Änderungen mussten andere Personen informiert und mit aktualisierten Dateien versorgt werden
- ▶ → schwierige Kooperation

Modellierung und Serialisierung

Lösung

- ▶ Einsatz des Versionskontrollsystems git
- ▶ RDF/XML-Serialisierung mit Texteditor bearbeitet
- ▶ gleichzeitige Änderungen möglich, Konflikte durch git merge-Mechanismus beheben
- ▶ Rückkehr zu jedem früheren Zeitpunkt möglich
- ▶ durch reguläre Ausdrücke gleichzeitige Änderungen an hunderten Entitäten gleichzeitig möglich
- ▶ Änderungen benötigen Kenntnisse in RDF/XML und git
- ▶ wenn großflächige syntaktische Änderungen fertig, Rückkehr zur Protégé möglich

Modellierung und Serialisierung

Modellierungsprinzipien

- ▶ Verwendung existierender Vokabulare
- ▶ Konsistenz: gleiche Eigenschaften auf gleiche Weise modellieren
- ▶ Zusammenfassen von gleichen Werten zu mehrfach genutzten Objekten, ähnlich Normalform bei Datenbanken, reduziert Inkonsistenzen, Arbeitsaufwand und Fehleranfälligkeit (Bsp.: Lehrbuchquelle)
- ▶ Bevorzugen von Object Properties gegenüber Data Type Properties

Modellierung und Serialisierung

Prefixe und Vokabulare

Ontologie	Prefix	Inhalt
meta	http://www.snik.eu/ontology/meta	SNIK Meta-Ontologie
bb	http://www.snik.eu/ontology/bb	SNIK Blaues Buch
ob	http://www.snik.eu/ontology/ob	SNIK Oranges Buch
ciox	http://www.snik.eu/ontology/meta	SNIK CIOx Interviews
ov	http://open.vocab.org/terms/	Ontologiedefinition
skos	http://www.w3.org/2004/02/skos/core#	Interlinks, Definitionen
dc	http://purl.org/dc/terms	Metadaten
bibo	http://purl.org/ontology/bibo/	Bibliographie

Dazu Standardvokabulare RDF, RDFS, OWL.

Modellierung und Serialisierung

Anwendung der Prinzipien

- ▶ konsequente Anwendung der Prinzipien zieht große Zahl an Änderungen nach sich
- ▶ in 3 Monaten: 31000 hinzugefügte, 28000 entfernte Zeilen
- ▶ teilweise automatisierbar, teilweise Entscheidung bei jedem Fall nötig
- ▶ Gartenmetapher: es ist immer etwas zu tun

Modellierung und Serialisierung

Anwendung der Prinzipien: Beispiel Synonyme

- ▶ Synonyme sind mit `<Synonym>Text</Synonym>` modelliert
- ▶ Problem 1a: Benutzung des leeren Präfixes führt bei jeder Teilontologie zu anderer URI (ob:Synonym, bb:Synonym, ...)
- ▶ Problem 1b: Synonym ist nicht definiert, daher genaue Semantik unbekannt, wird auch anderswo nicht verwendet

Modellierung und Serialisierung

Anwendung der Prinzipien: Beispiel Synonyme

- ▶ Typische Lösung: Identifizieren und Verwenden eines existierenden Vokabulars
- ▶ Also: Ersetzen von Synonym durch `skos:altLabel`
- ▶ Problem 2a: Wie entscheidet sich, welches `label` `rdfs:label` und welches `skos:altLabel` wird?
Existierende Daten inkonsistent z.B. bei Abkürzungen.
- ▶ Problem 2b: Language tags fehlen, entweder "deöder en".
- ▶ → manuelles Entscheiden in > 500 Fällen, Abkürzungen immer `skos:altLabel`

Modellierung und Serialisierung

Anwendung der Prinzipien: Beispiel Transitivität

- ▶ Materialisierung von transitiven Properties wie `rdfs:subClassOf`
- ▶ $A \subseteq B \subseteq C \rightarrow A \subseteq C$
- ▶ Diese Tripel können inferiert werden, Virtuoso und die Cytoscape.js unterstützen dies aber nicht.
- ▶ Materialisierte Tripel können von anderen nicht unterschieden werden und machen Visualisierung unübersichtlich.

Modellierung und Serialisierung

Anwendung der Prinzipien: Beispiel Transitivität

- ▶ Entscheidung: Nichts materialisieren, alles materialisieren oder nur zweitoberste Klasse (meta:Role/Function/EntityType) materialisieren? (oberste ist meta:Top)
- ▶ Anfangszustand: Teilweise nichts materialisiert, teilweise zweitoberste Klasse materialisiert.
- ▶ Entscheidung: Oberste Klasse mit neuer Property, meta:subTopClass angegeben, in Visualisierung nicht angezeigt

Modellierung und Serialisierung

Ausblick

- ▶ Fertigstellung großflächiger syntaktische Änderungen
- ▶ Kooperative punktuelle semantische Änderungen durch Domänenexperten
- ▶ siehe Abschnitt Qualitätssicherung

Visualisierung

Anforderungen

- ▶ performant bei mehreren tausend Knoten und Kanten
- ▶ keine Installation nötig
- ▶ geringer Implementationsaufwand
- ▶ Suchfunktion
- ▶ Filterung
- ▶ Graphoperationen wie kürzeste Wege, Spiderworm

Visualisierung

Designentscheidungen

- ▶ Javascript → keine Installation nötig, immer verfügbar, kein Server nötig
- ▶ Cytoscape.js performante Graphbibliothek mit genügend Funktionalität
- ▶ SPARQL Endpunkt mit bif:contains-Index für schnelle Suche (future work: Lucene Index)
- ▶ Pubby SPARQL Browser zur Detailansicht

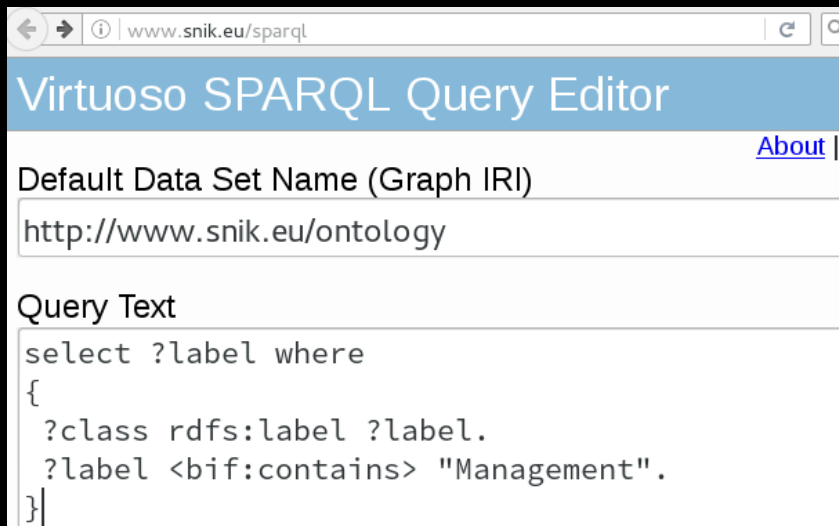
Visualisierung

Datenbereitstellung

- ▶ Cytoscape.js kann RDF nicht direkt verarbeiten, hat aber CSV import
- ▶ Virtuoso SPARQL Endpunkt kann Ergebnisse als CSV-Dateien abspeichern
- ▶ Ontologie nicht 1:1 abgebildet, z.B. Abflachen von OWL Restrictions
- ▶ CSV Dateien Cytoscape Desktop importieren, als JSON exportieren
- ▶ JSON-Datei mit Cytoscape.js laden

Visualisierung

Suche mit bif:contains SPARQL Query



The screenshot shows a web browser window with the address bar displaying `www.snik.eu/sparql`. The page title is "Virtuoso SPARQL Query Editor". Below the title, there is a link labeled "About". The main form contains two sections: "Default Data Set Name (Graph IRI)" with a text input field containing `http://www.snik.eu/ontology`, and "Query Text" with a text area containing a SPARQL query. The query is: `select ?label where { ?class rdfs:label ?label. ?label <bif:contains> "Management". }`

← → ⓘ | www.snik.eu/sparql | ↻ 🔍

Virtuoso SPARQL Query Editor

[About](#) |

Default Data Set Name (Graph IRI)

`http://www.snik.eu/ontology`

Query Text

```
select ?label where
{
  ?class rdfs:label ?label.
  ?label <bif:contains> "Management".
}
```

Visualisierung

Suche mit bif:contains SPARQL Query

label
"Fulfillment of Laws relevant to Information Management"@en
"Administration Management"@en
"Approval to Strategic Information Management Plan"@en
"Blood Bank Management System"@en
"Life Cycle Management of Strategic Information Management Plan"@en
"Change Management"@en
"Change Management"@en
"Configuration Management"@en

Visualisierung

Ausblick

- ▶ Suche von Phrasen statt Wörtern
- ▶ Suche mit Synonymen und Schreibfehlern
- ▶ z.B. durch Apache Lucene/SOLR index
- ▶ Bugfixing
- ▶ Wechsel von Pubby zu modernerem RDF browser
- ▶ Export von Selektionen

Qualitätssicherung

Ausgangspunkt: 5 Star Linked Data

1. Daten sind im Web in irgendeinem Format verfügbar ✓
2. maschinenlesbare strukturierte Daten (z.B. kein PNG) ✓
3. nichtproprietäres Format (z.B. CSV, nicht Excel) ✓
4. nach den offenen Standards des W3C publiziert (RDF and SPARQL) ✓
5. mit Links zu anderer Linked Data ✓ (zwischen Teilontologien) / ✗ (außerhalb SNIK)

Qualitätssicherung

Dimensionen der Qualität

- ▶ <http://www.semantic-web-journal.net/content/quality-assessment-linked-data-survey>, Meta-Studie von Amrapali Zaveri
- ▶ 18 gemeinsame Dimensionen
- ▶ teilweise subjektiv
- ▶ aufwändig zu bestimmen, teilweise Crowdsourcing nötig

Qualitätssicherung

Dimensionen der Qualität—Beispiele

- ▶ Accessibility – Serialisierte Dateien, SPARQL Endpunkt, im Browser aufrufbare URLs
- ▶ Lizenzen – bei Datenbanken nicht betrachtet aber bei Linked Data notwendig
- ▶ Interlinking – Verknüpfungen von und zu anderen Datensets
- ▶ Performance – Latenzzeit, Skalierbarkeit
- ▶ Understandability, Completeness, Relevanz, ...

Qualitätssicherung

Designierte Manuelle Korrektur

- ▶ semantische Korrektheit von Fakten benötigt manuellen Input
- ▶ serielles Durcharbeiten der serialisierten Ontologien beschränkt Personen auf Schnittmenge von Semantic-Web-Experten und Domänenexperten
- ▶ besser: zufällige Stichproben von Fakten
- ▶ manuell ausgezeichnet als korrekt, falsch oder ungewiss

Qualitätssicherung

Designierte Manuelle Korrektur

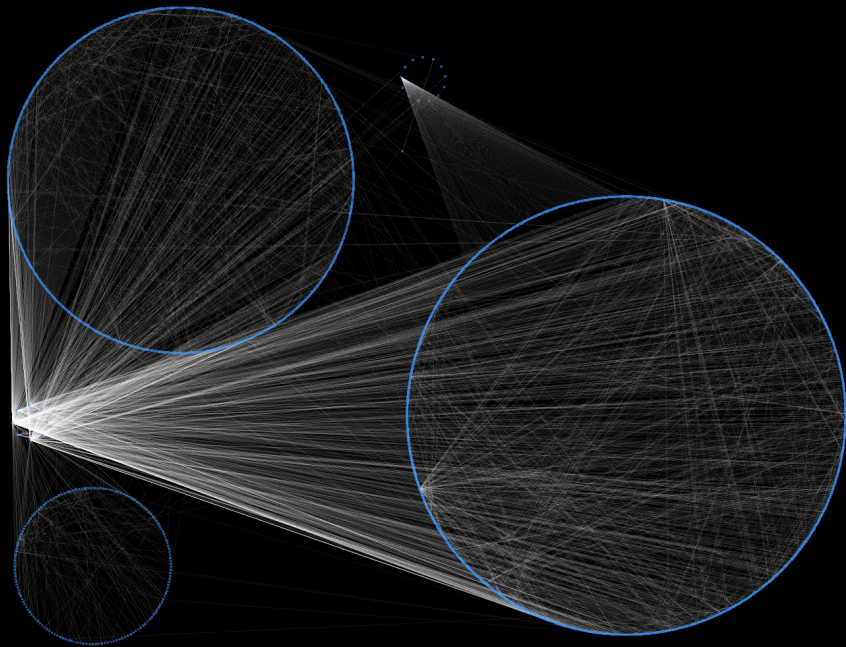
- ▶ Korrektur kann in beliebig großen Arbeitsabschnitten erfolgen
- ▶ bei Überschneidung inter-rater-agreement
- ▶ Triple Checkmate Tool von AKSW

Qualitätssicherung

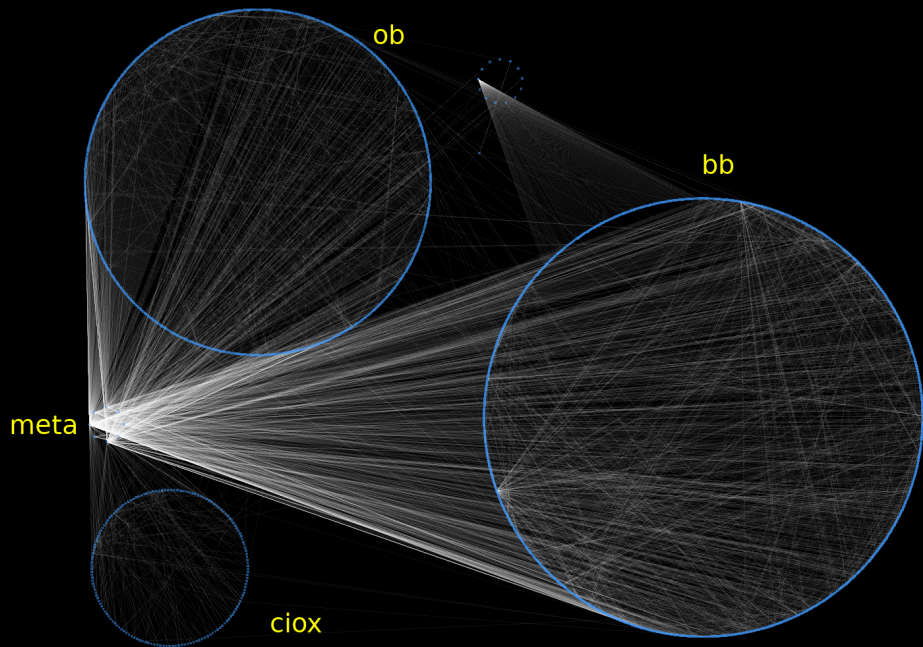
Feedback von Visualisierung

- ▶ wenn Fehler bemerkt werden, kann Ticket erstellt werden
- ▶ `https://bitbucket.org/imise/snik-ontology/issues`
- ▶ Feedback für Visualisierung und Ontologie getrennt
- ▶ (wenn Internet funktioniert) Vorführung

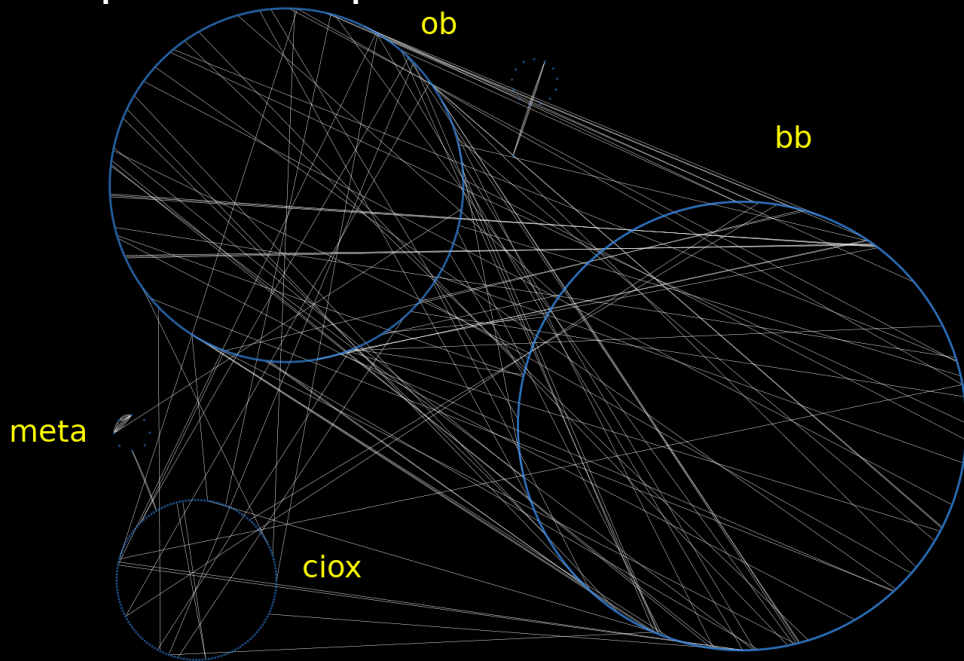
Cytoscape Desktop



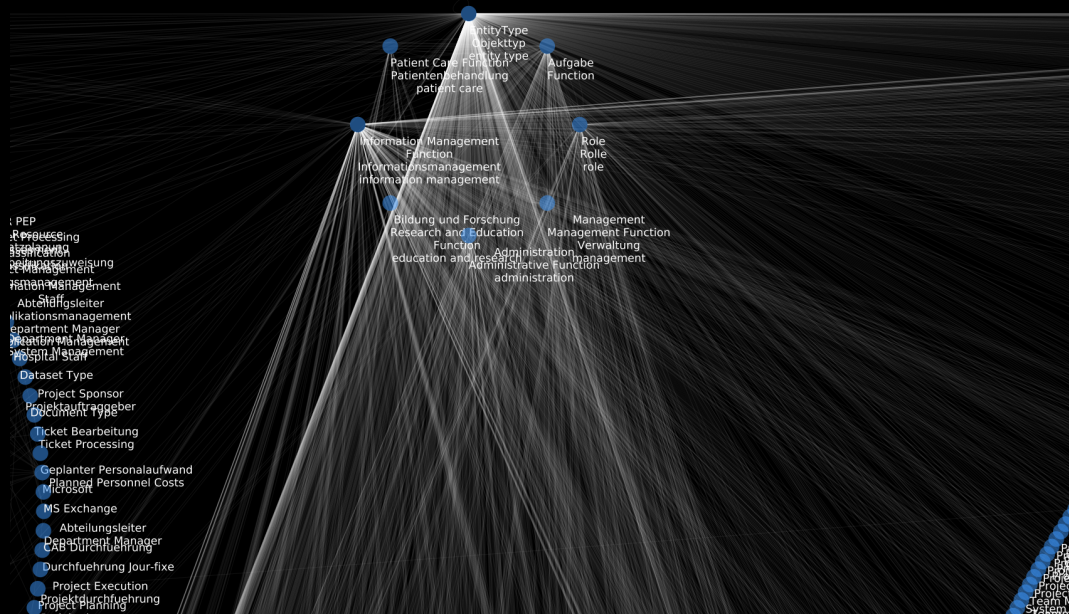
Cytoscape Desktop



Cytoscape Desktop



Cytoscape Desktop



Section 3

Nachträgliche Anmerkungen

Diskussion

- ▶ Transformation RDF->CSV mittels SPARQL Anatoli Zeiser beibringen
- ▶ Visualisierung bereitet Einigen Kopfschmerzen, Option auf hellen Hintergrund gewünscht
- ▶ Hervorheben von Interontologierelationen durch Filter (bereits erledigt)
- ▶ Hinzufügen fehlender Daten im RDF-Browser

Diskussion

- ▶ Beschriebene Prozesse stehen in der Mitte des Gesamtworkflows, Faktenextraktion ist davon unberührt.
- ▶ Für zukünftige Extraktionen (z.B. durch Birgit Schneider) ändert sich also nicht direkt etwas.
- ▶ Allerdings ist Änderung der Extraktion geplant, um großen Aufwand der Ontologiequalitätsverbesserung nach Extraktion zu verkleinern, durch Angleichen von Tabellenformular mit RDF und Ontologie.
- ▶ Außerdem Untersuchung von Excel2OWL-Alternativen geplant.

Referenzen und Weitere Informationen

- ▶ <https://wiki.imise.uni-leipzig.de/Projekte/SNIK/ontologie/workflow>
- ▶ <https://github.com/IMISE/snik-cytoscape.js>
- ▶ <https://github.com/IMISE/snik-ontology> (URL wird evtl. geändert)
- ▶ <https://bitbucket.org/imise/snik-ontology> (URL wird geändert)
- ▶ <http://www.snik.eu/graph>
- ▶ <http://www.snik.eu/pgraph>
- ▶ <http://www.snik.eu/sparql>
- ▶ <http://www.snik.eu/ontology>
- ▶ Und natürlich
[mailto://konrad.hoeffner@imise.uni-leipzig.de](mailto:konrad.hoeffner@imise.uni-leipzig.de) :-)

Mitarbeit

- ▶ alles open source (außer CIOx)
- ▶ alle können sehr gerne Mitentwickeln
- ▶ entweder pull-Request oder mich fragen und GitHub/Bitbucket-Account in Entwicklerteam aufnehmen lassen
- ▶ readme-Dateien im Wurzelordner der Repositories lesen
- ▶ Bug gefunden? Ticket im Repository erstellen.