

SNIK Ontologie—Lehre und Implementierung

[https://github.com/KonradHoeffner/latex/tree/
master/beamer/2016/snik-projekttreffen](https://github.com/KonradHoeffner/latex/tree/master/beamer/2016/snik-projekttreffen)

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

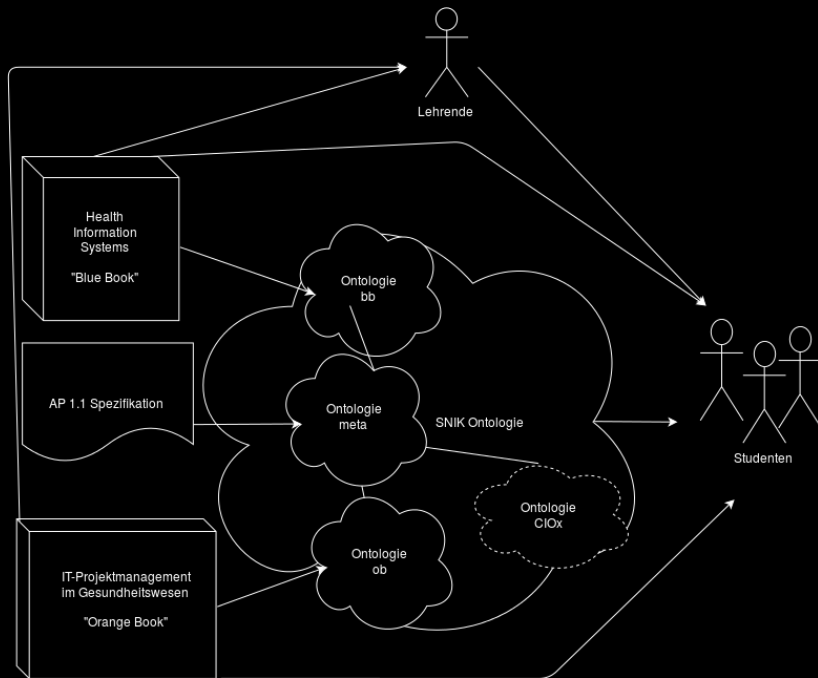
Vorstellung

- ▶ 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

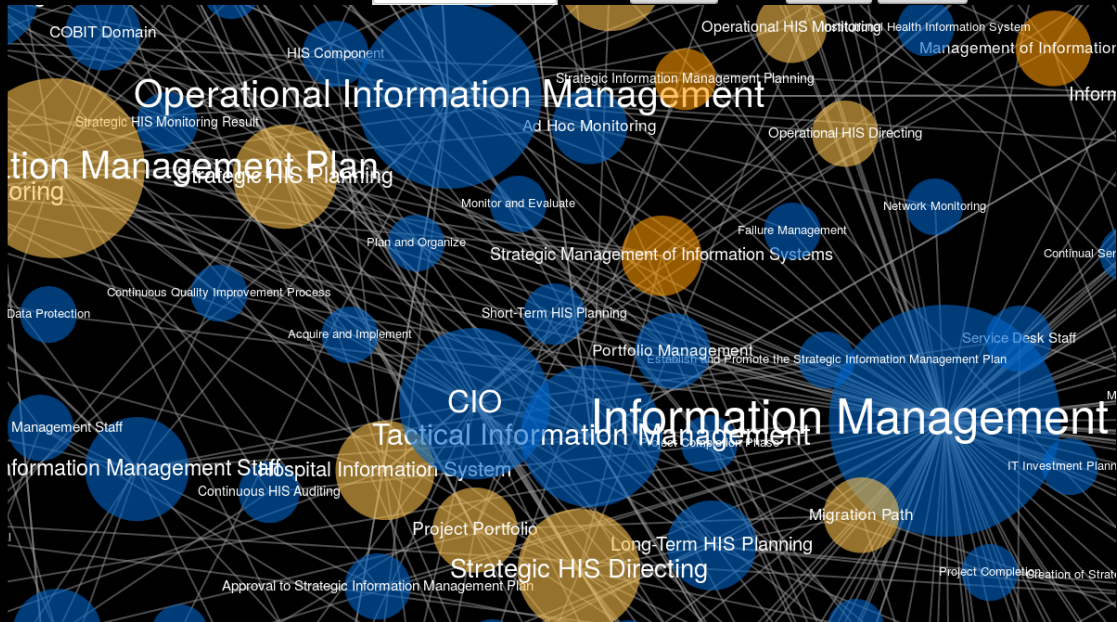
- ▶ modelliertes Wissen vermitteln, zusätzlich zu Lehrbüchern, Vorlesungen und Übungen
- ▶ Exploration
- ▶ Erstellen von Übungsaufgaben
- ▶ Semantic Web nur Mittel zum Zweck, so viel Zeit wie möglich für Gesundheitssysteme

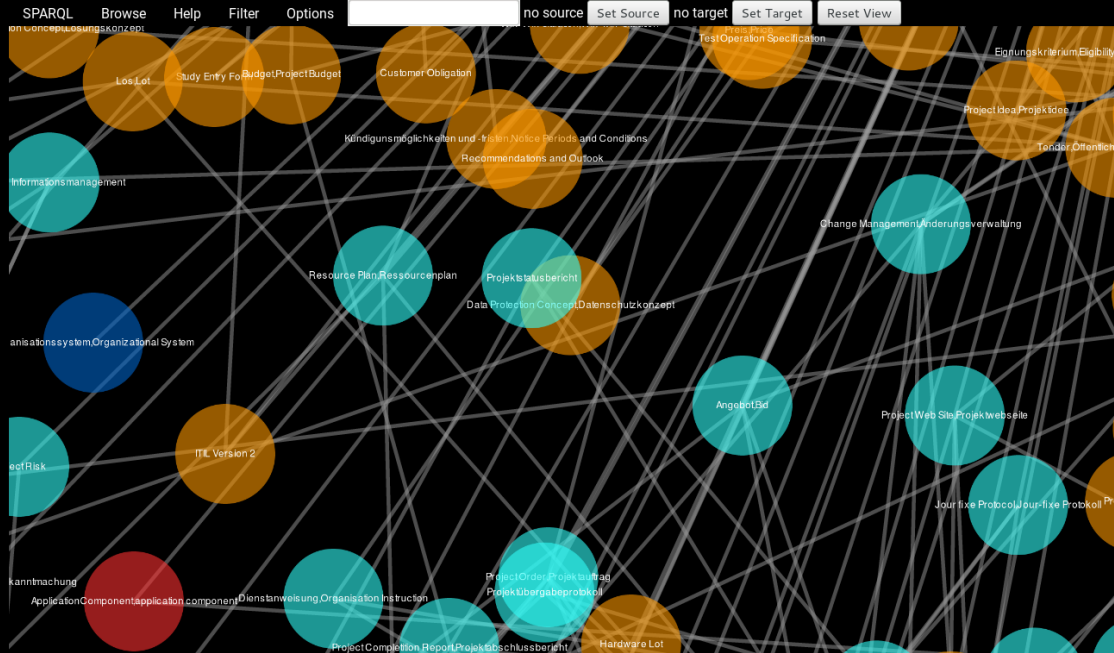
Problem

- ▶ 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 ohne CIOx
<http://www.snik.eu/graph/>
- ▶ Passwortgeschützte neue Version mit CIOx
<http://www.snik.eu/pgraph/>





Kürzester Weg

Computer-Based Information System

Tool

System

Application System

Information System

Professional

Information System

isDecomposed

updates

Organiz

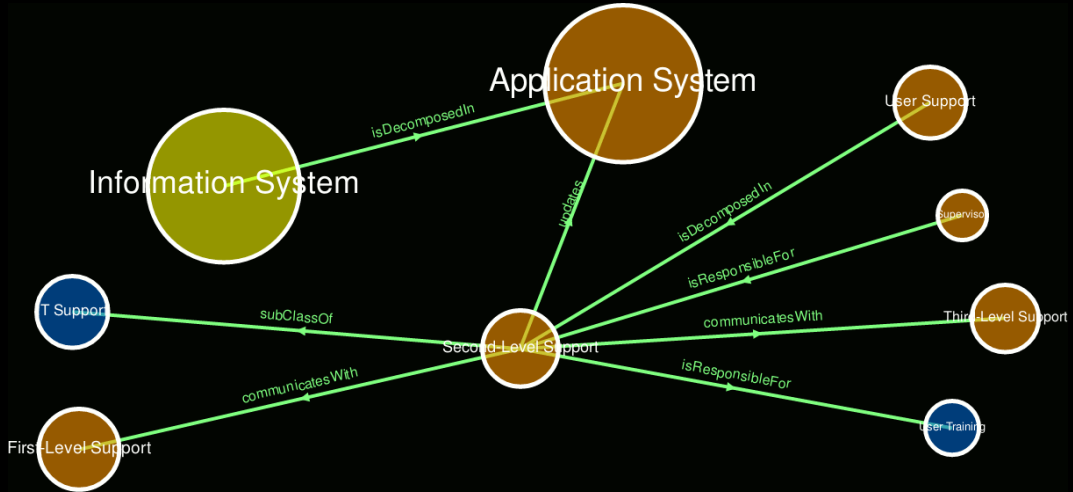
Software Reference Model

Second-Level Support

SWOT Analysis

Installation

Spiderworm



Praktische Vorführung

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

Section 2

Implementierung

Visualisierung

Anforderungen

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

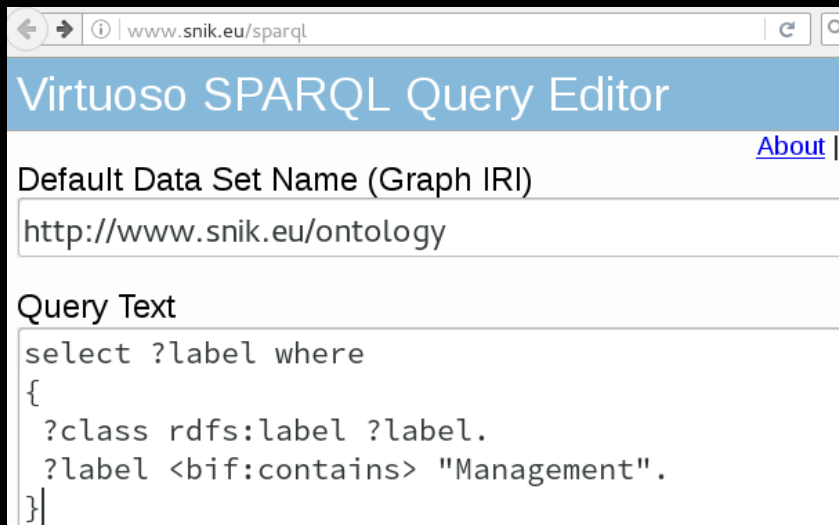
Visualisierung

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
- ▶ Pubby SPARQL Browser zur Detailansicht

Visualisierung

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.

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



Visualisierung



Vokabulare

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

work - zweisprachig (englisch und deutsch) - synonyme ->
bessere suchfunktion - suchfunktion im javascript basiert auf
bif:contains, nutzt virtuoso index, ist schnell - nachteil: muss
exakt vorkommen, edit distance = 0 - future work: edit
distance > 1, z.B. mit apache lucene/solr index
design decisions - rdfs:label vs skos:altLabel:
ausgeschriebenes ist rdfs:label, abkürzung skos:altLabel (z.B.
CIO, chief information officer) - aussagen über aussagen mit
owl axiomen - triple page - materialisierung von transitiven
properties wie rdfs:subclassOf, hier kann man: nichts
materialisieren, alles materialisieren, meta oberklasse
materialisieren (Role, Function, ...) -> nichts materialisierung,
weil sonst visualisierung unübersichtlich, es gibt sparql 1.1
property paths
probleme: owl modellierung und rdf-serialisierung/sicht -

protege zeigt gut an aber andere rdf tools haben probleme
mit vielen owl statments beispiel owl restriction -> some ...
-> blanknodes, komplizierte visualisierung, ist aber nötig
wegen meta-ontologie bei modellierung wurde mehr wert auf
owl gelegt, bei anwendung dann aber eher rdf
aufteilung: meta ontologie für gemeinsam verwendete
begriffe
future work: - bessere bibliographie (ohne eigenes vokabular
mit triple page und so) ersion