

# Handreichung für die Seminararbeiten im Modul *Angewandte Informatik* im Wintersemester 2020/21

Hans-Gert Gräbe

28. Januar 2021

## 1 Hintergrund

Im Bereich der TRIZ-Forschung wurde vor einem Jahr das *TRIZ Summit Ontologie-Projekt*<sup>1</sup> begonnen, um die verschiedenen Teile der TRIZ-Theorie ontologisch zu „kartieren“ und die wesentlichen Zusammenhänge zwischen den einzelnen Teilen mit Mitteln semantischer Technologien zu erfassen. Hierzu liegen aktuell vor

- (1) eine „Übersichtskarte“ der verschiedenen Theoriefelder und der Zusammenhänge zwischen diesen,
- (2) ein „Atlas“ von Ontokarten, die grob einzelne Bereiche markieren, die weiter zu detaillieren sind,
- (3) zu einzelnen dieser Ontokarten erste Versuche, Struktur in das begriffliche Chaos zu bringen,
- (4) ein Glossar (oder auch nur ein Thesaurus) von Begriffen, die hierfür wichtig sind.

Während zu (1) und (2) weitgehend Konsens besteht, sind die Modellierungen in (3) stark umstritten, da die entsprechenden Semantiken und Zusammenhänge in den unterschiedlichen TRIZ-Schulen naturgemäß unterschiedlich verstanden werden.

Streit gibt es auch zu (4), der aber deutlich einfacher zu klären ist, wenn

- (4a) zunächst einmal alle Begriffe gesammelt und „URIifiziert“ werden (Thesaurus raw),
- (4b) URIs so weit zusammengeführt werden, dass verschiedene URIs auf verschiedene Konzepte verweisen, aber Raum bleibt, für gleiche Konzepte verschiedene Semantiken zu hinterlegen (Thesaurus final),
- (4c) diese verschiedenen Semantiken auch wirklich zusammengetragen und formalisiert werden (Glossar raw) und schließlich
- (4d) die Semantiken in einem komplexen sozialen Abstimmungsprozess so weit wie *möglich* abgeglichen und essenzielle Differenzen semantisch modelliert werden.

In [2] wird das Projekt genauer beschrieben, in [3] der aktuelle Stand dargestellt. Andere Vorarbeiten fanden wenig Berücksichtigung<sup>2</sup>, insbesondere weder [1] noch [5] oder [6]. Im

---

<sup>1</sup>Siehe [https://triz-summit.ru/onto\\_triz/](https://triz-summit.ru/onto_triz/) (in Russisch).

<sup>2</sup>Details dazu auf der Webseite <https://wumm-project.github.io/Ontology.html>.

Herbst 2020 lief eine Webinarreihe (in Russisch), deren Materialien auch verfügbar<sup>3</sup> und teilweise ins Englische übersetzt sind.

Arbeitsgrundlage der TRIZ Summit Ontologie-Projekts ist ein Glossar von V. Souchkov [4], wobei dessen Einträge sowohl drei TRIZ-Generationen (TRIZ-1..3) als auch 5 Kategorien (Basisbegriffe, Modelle, Regeln, Begriffsgruppen, Synonyme) zugeordnet werden. Ein Webinarteilnehmer wies auf einen weiteren Thesaurus [5] auf den einschlägigen russischen Altschullerseiten hin, der bereits multilingual vorliegt.

Im Rahmen des WUMM-Projekts wurden und werden Teile dieser Ontologisierungen nachmodelliert und im *WUMM RDFData github Repo*<sup>4</sup> hochgeladen, die im Original bisher ausschließlich durch grafische Ontogramme sowie die Möglichkeit einer visuellen Inspektion im verwendeten OSA-Ontologie-Editor<sup>5</sup> zugänglich sind. Diese im Original ausschließlich russischsprachigen Quellen wurden dabei in Teilen auch ins Englische und Deutsche übertragen. Weitgehend semantisch erfasst sind (1) und (2). Weiterhin wurde bereits früher das VDI-Glossar „RDFiziert“ und die dort vorhandenen deutschen und englischen Erläuterungen um eine russische Übersetzung ergänzt. Dies sowie der einfach zu transformierende Thesaurus auf den Altschullerseiten bilden die Grundlage für einen eigenen Thesaurus nach (4a), der mit den Begrifflichkeiten des Originalprojekts weiter abzugleichen ist. Inzwischen wurde auch das Glossar [4] in der aktuellen Version 1.2 auf diese Weise aufbereitet. Details dazu weiter unten.

## 2 Themen für Seminararbeiten

In den Seminararbeiten soll am Punkt (3) weitergearbeitet werden, indem für eine konkrete Ontokarte *X* (bzw. einen anderen konkreten Modellierungszusammenhang im TRIZ-Kontext entsprechend den mündlichen Absprachen)

- (A) die Zusammenhänge für das WUMM-Projekt in RDF in einer gemeinsamen Rahmensezung nachmodelliert werden sowie
- (B) differierende Semantiken, Probleme und Widersprüche im Verständnis der modellierten TRIZ-Konzepte zusammengetragen und systematisiert werden.

Während (A) primär einen ingenieur-technischen Charakter hat, erfordert (B) stärker einen akademischen Zugang von Recherche und Vergleich einschlägiger Publikationen.

Informationen zu (A) liegen typischerweise in Form von RDF-Diagrammen vor, wie etwa die *Top Level Ontologie* in Abbildung 1, und sollen in eine Turtle-Datei vergleichbar zur Datei *TopLevel1.ttl* im WUMM RDFData github Repo überführt werden.

---

<sup>3</sup>Siehe <https://wumm-project.github.io/OntologyWebinar>.

<sup>4</sup>Siehe dazu das Verzeichnis *Ontologies* im github-Repo <https://github.com/wumm-project/RDFData>

<sup>5</sup>Siehe <https://onto.devtas.ru/ts2o1> (in Russisch).

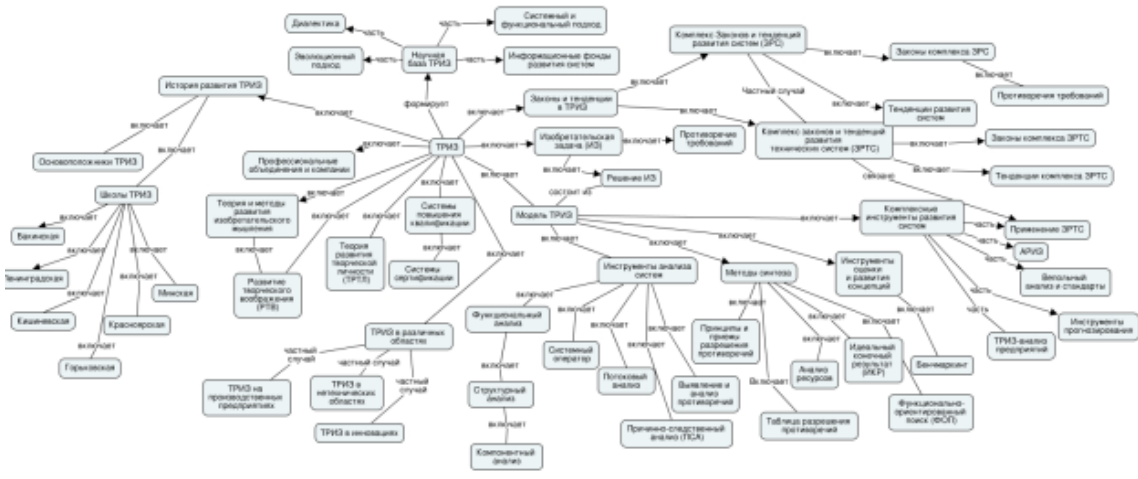


Abbildung 1: Die Top Level Ontologie als RDF Graph

Entsprechende Quellen für einen Startpunkt Ihrer Recherchen finden Sie in der README-Datei in Ihrem Projektverzeichnis.

Für jede Seminararbeit ist ein Unterverzeichnis eingerichtet. Bitte forken Sie (so weit noch nicht geschehen) das Repo auf einen eigenen Account, erstellen die Seminararbeit und geben Zwischen- und das Endergebnis über einen Merge-Request zur Begutachtung frei. Ich werde mich bemühen, die Ergebnisse kurzfristig zu durchzusehen und mit Ihnen zu besprechen.

Die Seminararbeit selbst soll in  $\text{\LaTeX}$  erstellt werden und möglichst in Englisch verfasst sein.

### 3 Grundlagen des WUMM-Ontologie-Projekts

Im TRIZ Summit Ontologie-Projekt geht es zunächst darum, die TRIZ-Begriffslandschaft zu „kartieren“. Das WUMM-Ontologie-Projekt begleitet diese Aktivitäten, um

1. eine Remodellierung nach semantischen Standards auszuführen,
2. die Materialien multilingual aufzubereitet und
3. auf dieser Basis eine LOD-Infrastruktur aufzubauen,

und so eine bessere Basis für die erforderlichen sozialen Abstimmungsprozesse zu schaffen.

#### 3.1 Die SKOS Ontologie als Basis

Dafür wird die SKOS-Ontologie [8] verwendet, die mit den Konzepten (K)

- `skos:Concept`, `skos:prefLabel`, `skos:altLabel` – Objektbenennung
- `skos:definition`, `skos:example`, `skos:note` – Objekteigenschaften
- `skos:narrower`, `skos:broader` – Objektrelationen

einen ersten Beschreibungsrahmen für Begrifflichkeiten bietet. Für die genauere Bedeutung der einzelnen Konzepte wird auf [8] verwiesen. Auf der Basis wurden bisher drei Quellen ausgewertet,

- der mehrsprachige Thesaurus [5],
- das Glossar der VDI-Norm [6] und
- das Glossar [4], Version 1.2, von V. Souchkov.

Außerdem wurde das Glossar des TRIZ Summit Projekts teilweise integriert. Deren Glossar ist derzeit öffentlich nur in einer HTML-Version verfügbar. Interne maschinenlesbare Versionen wurden mit dem Verweis auf den vorläufigen Charakter der Modellierung noch nicht zur Veröffentlichung freigegeben. Auch lassen sich Daten in der OSA-Plattform nicht ohne Weiteres über eine API in gängigen RDF-Formaten extrahieren. Deshalb habe ich den genaueren Abgleich zugunsten der Konsolidierung des WUMM-Thesaurus zurückgestellt.

### 3.2 URIs und Namensräume

Eines der zentralen Probleme der Überführung der vorhandenen Datenbestände zu TRIZ-Konzepten ist die Zuweisung sinnvoller URIs, da die einzelnen Glossareinträge in den bisherigen Quellen einzig durch ihre Bezeichner („labels“) identifiziert werden. Hier macht auch die OSA-Plattform keine Ausnahme, denn die dort vergebenen URIs (sowohl für die Knoten als auch die Kanten des aufgebauten RDF-Graphen) sind nicht öffentlich sichtbar.

Bei der maschinellen Transformation der Datenbestände in ein valides RDF-Format wurden für alle Konzepte automatisiert URIs erzeugt, die im Namensraum `tc:` (wie *TRIZ Concepts*) liegen. Eine wesentliche noch bevorstehende Aufgabe ist die Unifizierung dieser URIs, also die Zusammenführung von verschiedenen URIs, welche auf denselben Begriff verweisen. Diese Arbeit soll demnächst in Angriff genommen werden.

### 3.3 Provenienz von Erläuterungen

Ein weiteres Problem dieser ontologischen Modellierung ist die genauere Darstellung der Provenienz der einzelnen Erläuterungen. Hierzu wurden die unter (K) aufgeführten SKOS-Konzepte für jede einzelne Quelle durch Notationen aus dem Namensraum `od:` verfeinert, um zunächst die „Welten“ der einzelnen Autoren und TRIZ-Schulen separat zu erfassen.

`od:` ist der Namensraum, den das WUMM-Projekt verwendet, um dort seine eigenen Konzepte zu entwickeln. Eine genauere Beschreibung dieses Namensraums in einer eigenen RDF-Datei steht noch aus, die durch die URIs repräsentierten Konzepte sind bisher nur mündlich abgestimmt.

Entsprechende Notationsvariationen sind etwa

- `skos:Concept` → `od:GSAThesaurusEntry`, `od:VDIGlossaryEntry` ...
- `skos:definition` → `od:SouchkovDefinition`, `od:VDIGlossaryDefinition` ...
- `skos:example` → `od:VDIGlossaryExample` ...

usw. Siehe dazu die RDF-Daten selbst, die über den SPARQL-Endpunkt

`http://wumm.uni-leipzig.de:8891/sparql`

des WUMM-Projekts durchsucht werden können.

Die *Verwendung verschiedener Prädikatsbezeichner* erlaubt es, auf einfache Weise die Herkunft – etwa von Definitionen – einzelnen Quellen bereits auf der Ebene von Tripeln zuzuordnen:

(Begriff X) – (wird in Quelle A definiert als) – (Definition von X in Quelle A)  
(Begriff X) – (wird in Quelle B definiert als) – (Definition von X in Quelle B)

Würde einheitlich `skos:definition` verwendet, so müssten komplexere Konstrukte

(Begriff X) – `skos:definition` –  $\left[ \begin{array}{l} \text{(Quelle A) – (Definition von X in Quelle A)} \\ \text{(Quelle B) – (Definition von X in Quelle B)} \end{array} \right]$

eingesetzt werden. Auch wenn dies das finale Ziel der Übung ist, soll im aktuellen Stand der Modellierung die einfachere Form verwendet werden, obwohl damit die Zahl der Prädikate aufgebläht wird. Eine Konsolidierung der Modellierung in Richtung des „idealen Endresultats“ ist später durch eine einfache Modelltransformation zu erreichen.

Dasselbe gilt für die Verwendung von provenienzzabhängigen Unterklassen von `skos:Concept`. Da der Abgleich der für Konzepte vergebenen URIs unterklassenübergreifend erfolgen muss, sind die Instanzen jeder Unterklasse auch als `skos:Concept` ausgezeichnet, obwohl dies auch über einen generalen Satz wie etwa

`od:VDIGlossaryEntry rdfs:subClassOf skos:Concept`

beschrieben und dann durch Inferenz festgestellt werden könnte. Das Inferieren von Eigenschaften gehört aber zu komplexeren semantischen OWL-Konzepten und wird von einfachen RDF-Stores nicht unterstützt. Auch hier gilt, dass die Daten später durch eine einfache Modelltransformation konsolidiert werden können.

### 3.4 Modellierung eines TRIZ-Teilgebiets

Typischerweise liegt das zu modellierende Teilgebiet als RDF-Graph vergleichbar zu Abbildung 1 vor. Dazu ist eine eigene Turtle-Datei anzulegen und zu befüllen, welche diesen Graphen nachmodelliert.

Die RDF-Modellierung eines solchen Graphen sollte in drei Schritten vorgenommen werden:

1. Modelliere zunächst die Knoten als TRIZ-Konzepte, wobei möglichst URIs und damit Konzepte aus dem vereinten Glossar (also bereits vorhandene Instanzen vom Typ `skos:Concept`) verwendet werden sollten. Die Knotenbeschriftung ist entweder schon im Glossar oder als `skos:prefLabel` oder (in einzelnen Fällen) als `skos:altLabel` (in Ihrer Datei) zu ergänzen, ggf. auch mehrsprachig. In jedem Fall muss das Label einen RDF-Sprachtag tragen.

Wenn Sie neue Konzept-URIs einführen, müssen diese im Namensraum `tc:` liegen und als sowohl zum RDF-Typ `skos:Concept` als auch `od:AdditionalConcept` gehörend ausgezeichnet sein.

2. Modelliere danach die Kanten, wobei neue oder bereits vorhandene Prädikat-URIs aus dem Namensraum `od:` verwendet werden. Die Wahl der URI sollte sich an einem kurzen englischen Begriff als „sprechender Name“ orientieren.

Prädikate haben zunächst keinen Label, allerdings müssen Kanten im Graphen mit gleichem Label durch dieselbe Prädikat-URI modelliert werden. Zur Auswahl von Prädikat-URIs unten mehr.

3. Ergänze schließlich Definitionen, Erläuterungen und Beispiele zu den Knoten (so weit vorhanden), die als `skos:definition`, `skos:node` und `skos:example` zu modellieren sind. Auch hier bitte konsequent RDF-Sprachtags einsetzen. Wenn Sie differierende Erläuterungen desselben Prädikats aus unterschiedlichen Quellen erfassen, dann spalten Sie das SKOS-Prädikat in verschiedene Unterprädikate aus dem Namensraum `od:` auf wie im Abschnitt 3.3 erläutert.

Wenn Sie nicht genau wissen, wie eine Objektinformation an einen Knoten angebunden werden soll, dann verwenden Sie das Prädikat `rdfs:comment`.

Bitte beachten Sie, dass Literale in einfachen Quotes keine Zeilenumbrüche enthalten dürfen und Quotes im Text durch `\` ausgezeichnet werden müssen. Am einfachsten vermeiden Sie einfache Quotes innerhalb von Strings und ersetzen diese durch German Quotes („...“) oder French Quotes («...» in der russischen und deutschen Schreibweise oder »...« in der französischen). Die entsprechenden Zeichen sind auf gängigen deutschen Tastaturen in der Alt-Gr-Belegung verfügbar.

Mehrzeilige Literale können durch Einschließen in Triple-Quotes erstellt werden. Dann müssen Quotes im Text auch nicht ausgezeichnet werden.

### 3.5 Modellieren von Prädikaten und UML-Diagramme

Ein Prädikat wird in einem RDF-Graphen durch einen beschrifteten gerichteten Pfeil zwischen zwei Knoten markiert. Oft gibt es zu einem Prädikat auch ein inverses Prädikat (etwa `org:memberOf` und `org:hasMember`), das einem Pfeil in umgekehrter Richtung entspricht.

Bitte beachten Sie, dass die Pfeile in den OSA-Diagrammen des TRIZ Summit Ontologie-Projekts willkürlich gerichtet sind, da im Modellierungs-Tool stets *beide* Prädikate (direktes und inverses) benannt werden können, die Richtung der Pfeile im Diagramm willkürlich ist und in verschiedenen Ansichten auch wechselt. Das gilt nicht für Cmap<sup>6</sup>-Diagramme wie in Abbildung 1.

In Arbeiten wie etwa [1] werden *UML-Diagramme* zur Modellierung verwendet, siehe etwa [1, Fig. 2]. Neben (wenigen) benannten Pfeilen werden dort UML-Konstrukte wie Unterklasse und Aggregation zur Darstellung von Beziehungen zwischen Klassen verwendet sowie Objekteigenschaften durch Klassenattribute modelliert. Diese müssen alle durch Prädikate vom Typ `rdf:Property` [9] modelliert werden, da erst in höheren semantischen Konzepten wie OWL [7] Verfeinerungen wie `owl:ObjectProperty` zur Verfügung stehen. Es gibt keine etablierten Standards, wie relationale UML-Konstrukte nach RDF umgesetzt werden, weshalb

---

<sup>6</sup>Cmap ist ein weiteres grafisches Modellierungstool, das im TRIZ Summit Ontologie-Projekt eingesetzt wurde.

Sie hier frei in der Wahl von Prädikat-URIs (aus dem Namensraum `od:`) sind, wenn Sie gleiche UML-Konstrukte auch gleich benennen. Bei Attributen ist die Sache einfach, da sowohl der `rdfs:domain` (die Klasse, zu welcher das Attribut gehört) als auch das Label des Attributs gegeben sind. Allein der `rdfs:range` muss gelegentlich genauer bestimmt werden, wenn er nicht `rdfs:Literal` ist. Bitte orientieren Sie sich auch hier an der Umsetzung der Top Level Ontologie in der Datei `TopLevel.ttl`, insbesondere an den dort bereits verwendeten Prädikat-URIs.

## 4 Werkzeuge

Im Zuge des Seminars haben Sie gezeigt, dass Sie mit  $\text{\LaTeX}$  umgehen können. Bitte verwenden Sie möglichst die Style-Vorlage `ls.sty` im Repo für die grundsätzlichen Definitionen und ergänzen Sie dies lokal in Ihrer  $\text{\LaTeX}$ -Präambel. `ls.sty` kann etwa durch einen symbolischen Link in Ihrem Arbeitsverzeichnis verfügbar gemacht werden.

Das Erstellen einer RDF-Ontologie bedarf keiner besonderen Werkzeuge, da eine Turtle-Datei eine Textdatei ist und mit jedem Texteditor erzeugt und bearbeitet werden kann, der plain text (!) liefert. Für die Validierung und Transformation solcher Dateien haben sich die *Redland RDF Libraries* <http://librdf.org> und dort besonders das Kommandozeilen-Tool **raper**<sup>7</sup> bewährt. Mit `raper -gc` (RDF-Datei) kann damit die Syntax einer RDF-Datei validiert, mit

```
raper -g (Eingabedatei) -o (format)
```

eine Eingabedatei in ein anderes RDF-Format (turtle, ntriples) umgewandelt oder im selben Format normalisiert (und formatiert) werden. Die Ausgabe erfolgt auf `stdout` und muss ggf. umgelenkt werden.

## Literatur

- [1] D. Cavallucci, F. Rousselot, C. Zanni (2011). An ontology for TRIZ. Proc. TRIZ Future Conference 2009. Procedia Engineering 9, 251–260.  
<https://doi.org/10.1016/j.proeng.2011.03.116>.
- [2] A. Kuryan, V. Souchkov, D. Kucharavy (2019). Towards ontology of TRIZ. TRIZ Developers Summit, Minsk 2019.  
<https://wumm-project.github.io/Texts/Ontology-TDS2019-en.pdf>
- [3] A. Kuryan, M. Rubin, N. Shchedrin, O. Eckardt, N. Rubina. TRIZ Ontology. Current State and Perspectives. TRIZ Developers Summit 2020 (in Russisch). Englische Übersetzung:  
<https://wumm-project.github.io/Texts/Ontology-TDS2020-en.pdf>
- [4] V. Souchkov. Glossary of TRIZ and TRIZ-related Terms. Mehrere Versionen seit 1991. Letzte Version 1.2 von 2014.

---

<sup>7</sup><http://librdf.org/raptor/raper.html>. In gängigen Linux-Distributionen kann das über die Paketverwaltung installiert werden.

- [5] Thesaurus auf `altshuller.ru`. <https://www.altshuller.ru/thesaur/thesaur.asp>
- [6] VDI Richtlinie 4521. Erfinderisches Problemlösen mit TRIZ.  
Blatt 1: Grundlagen und Begriffe (April 2016).  
Blatt 2: Zielbeschreibung, Problemdefinition und Lösungspriorisierung (April 2018).  
Blatt 3: Lösungssuche (Juli 2020).
- [7] W3C (2004). OWL Web Ontology Language Guide.  
<https://www.w3.org/TR/2004/REC-owl-guide-20040210>
- [8] W3C (2008). SKOS Simple Knowledge Organization System RDF Schema.  
<https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html>
- [9] W3C (2014). RDF Schema 1.1. <https://www.w3.org/TR/rdf-schema>