In: Petra Drewer, Felix Mayer, Donatella Pulitano (Hrsg.)(2023): Terminologie: Tools und Technologien. Akten des Symposions des Deutschen Terminologie-Tags e.V. (DTT). Mannheim, 2.–4. März 2023. München, Köln, Bern: DTT e.V., S. 31-42.

Terminologie und Ontologie: WordNet trifft SKOS

Johannes Busse

1 Einleitung

Als exemplarisches Eingangsbeispiel sei ein kleiner Thesaurus zur Schnitzelkunde gegeben:

```
prefix cpt: <http://jbusse.de/dtt2023/cpt.html#term->
cpt:Schnitzel
  „escalope“@en
  „cutlet“@en
  „шни́цель“@ru
  <Schnitzel BY hat_Herkunft>
    cpt:Schweineschnitzel
      „pork cutlet“@en
      „свино́й шни́цель“@ru
    cpt:Kalbsschnitzel
      „veal cutlet“@en
  <Schnitzel BY hat_Zielgruppe>
    cpt:Kinderschnitzel
```

Abbildung 1: Beispiel für einen Thesaurus (eigene Darstellung)

Wären wir Köche, würden wir diese bereits als Thesaurus ausgearbeitete Terminologie vielleicht zu einer voll axiomatisierten Schnitzel-Ontologie ausbauen wollen. Auf dem DTT-Symposion sind allerdings nicht Köche, sondern Terminologen unterwegs, die im einführenden Schnitzel-Beispiel nicht über einzelne Schnitzel oder das Schnitzel an sich, sondern über Benennungen und Begriffe (und später auch Synsets) reden. Uns interessiert die Struktur, wie wir über eine Domäne reden.

Das ist auch die Perspektive des vorliegenden Aufsatzes: Wie kann man als Semantic Web-Ontologe über Begriffe (und Benennungen und Terminologien etc.) reden – und zwar insbesondere in Zusammenhang mit der Ontologie SKOS und der lexikalischen Ressource WordNet?

2 SKOS

Was genau eine (Semantic-Web-)Ontologie ist, kann an dieser Stelle nicht vorauseilend definiert werden, sondern soll an einem Beispiel Schritt für

Schritt entwickelt werden. Als Einstieg kann aber schon gesagt werden: Mit einer Ontologie können wir sehr exakt vereinbaren, wie wir über Dinge reden und sie unterscheiden wollen – und zwar so exakt formalisiert, dass ein Logiksystem daraus Schlussfolgerungen ziehen kann. Eine Ontologie ist damit ein voll axiomatisierter Bestand an Begriffen, mithin eine axiomatisierte Terminologie, wobei wir Terminologie mit DIN 2342 (2011, S. 16) verstehen als einen

> *„Gesamtbestand der Begriffe und ihrer Bezeichnungen in einem Fachgebiet" (zitiert nach Drewer und Schmitz, Terminologiemanagement, 2017, S.6)*

Um zu verstehen, wie eine solche Formalisierung funktioniert, modellieren wir auszugsweise grundlegende Strukturen von SKOS nach. SKOS ist eine Ontologie, die uns mit Begriffen und Relationen versorgt, mit denen wir eine gegebene Menge von Begriffen und zugehörigen Bezeichnungen (also Termen) (a) aufstellen und sogar (b) als Thesaurus strukturieren können. Um SKOS zu verstehen (und daran zu zeigen, was eine Ontologie ist), machen wir die Struktur explizit, die wir im einleitenden Schnitzel-Beispiel angelegt haben.

Unsere Ontologie wird nicht über Dinge selbst (z. B. Schnitzel) und ihre Beziehungen (z. B. `Kalbsschnitzel hat_herkunft Kalb`) Auskunft geben. Denn als Terminologen interessiert „Schnitzel" nicht als einzelnes Schnitzel oder als Klasse Schnitzel in einer Nahrungsmittel-Ontologie, sondern als Bedeutungseinheit, als Begriff. Damit sind zwei grundlegend unterschiedliche Ebenen schon angesprochen:

- Die Ebene der *Meta-Terminologie* verwenden wir, um Begriffe und Relationen bereitzustellen, mit denen wir eine Domänen-Terminologie aufbauen können, wie z. B. die Klasse skos:Concept. Weil diese Meta-Terminologie mit formal-logischen Mitteln voll axiomatisiert vorliegt, spricht man von einer Ontologie (technisch terminology box, TBox[9]). SKOS ist eine solche Tbox / eine solche Ontologie.
- Die Ebene der *Domänen-Terminologie* verwenden wir, um Terminologien für einen Gegenstandsbereich wie z. B. Schnitzelkunde aufzubauen. Weil hier i. A. keine formal-logische Axiomatisierung vorliegt, spricht man bei der Darstellung der entsprechenden Begriffsbeziehungen gerne auch von einem Semantischen Netz oder einem Knowledge Graph. Weil dieser Knowledge Graph im Wesentlichen aus Behauptungen und Sachaussagen zwischen Instanzen (in SKOS hauptsächlich der Klasse `skos:Concept`) besteht, spricht man auch von einer Assertion Box (Abox).

Diese zwei Ebenen spiegeln sich auch in den Visualisierungen des SKOS Core Guide (2005) wider. Die Domänen-Terminologie – die typischerweise aus

[9] Der Aufsatz liegt auch in einer erweiterten und aktualisierten Web-Fassung vor. Die Unterstreichung kennzeichnet die Verlinkungen auf http://jbusse.de/dtt2023/.

einem Netz von hunderten oder tausenden einzelnen Knoten (Objekten, Begriffen, Instanzen von `skos:Concept`) besteht, die sich z. B. als Thesaurus hierarchisch und assoziativ vielfältig vernetzen lassen und für die es verschiedene lexikalische Repräsentationen (Labels, Benennungen) gibt – wird im SKOS-Core-Guide mit runden, violett eingefärbten Knoten visualisiert (siehe Abbildung 2):

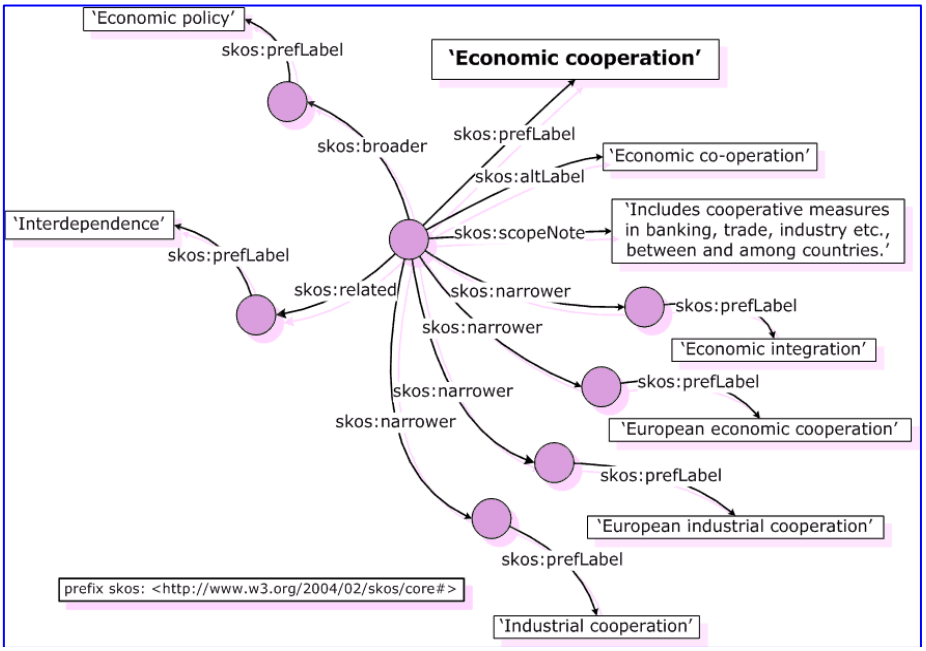

Abbildung 2: „Economic cooperation", aus SKOS Core Guide (2005)

Wie wir das auch schon aus der Terminologiewissenschaft kennen, lässt sich auch in SKOS das Verhältnis von Benennung und Begriff als eine m:n-Relation denken: Die gleiche Benennung kann für verschiedene Begriffe stehen, und jeder Begriff kann verschiedene Benennungen haben.

Dass ein bestimmter rund-violetter Knoten als ein Begriff verstanden werden soll, wird in dem SKOS-Core-Guide visualisiert (siehe Abbildung 3). In diesem Beispiel ist `ex:love` ein Objekt aus der Domänen-Terminologie und `skos:Concept` ist ein Objekt aus der Meta-Terminologie, hier der Ontologie SKOS. Wir deklarieren das Objekt `ex:love` als Begriff, indem wir ihm den Typ `skos:Concept` zuweisen. Aus Sicht der formalen Logik wird durch diese Typzuweisung `ex:love` als Instanz der Klasse (Element der Menge) `skos:Concept` behandelt:

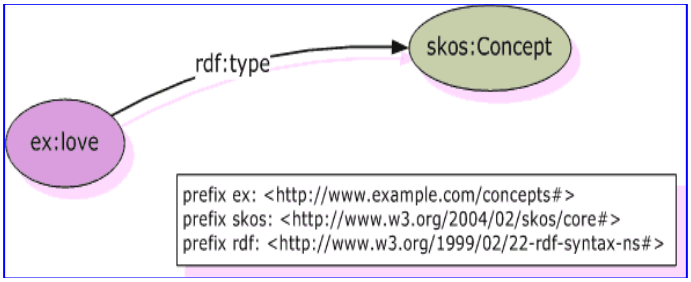

Abbildung 3: ex:love rdf:type skos:Concept

3 Die Struktur von SKOS

Als Terminologen interessieren uns die Strukturen, mit denen sich Terminologien aufbauen lassen, hier also die Struktur von SKOS. Eine erste Frage lautet: Welche Typen von Objekten können wir in der (Meta-)Terminologie der (Domänen-)Terminologie überhaupt unterscheiden?

Auf grundlegender Ebene unterscheiden wir in den SKOS-Visualisierungen zunächst die zwei Grundtypen von Daten in RDF, nämlich

- sogenannte Literale, hier in der Form von RDF-language tagged strings: Mit solchen Literalen werden in SKOS Labels modelliert;
- sogenannte RDF-Resources, die in der weitverbreiteten Notation RDF turtle (2014) als IRIs meist in der Form `prefix:local_part` notiert werden: Damit werden in SKOS Objekte in der Mehrzahl vom Typ `skos:Concept` oder `skos:Collection` notiert.

In unserem einleitenden Schnitzel-Beispiel kommen Resources von verschiedenem Typ vor:

- Resources vom Typ `skos:Concept`: Schnitzel, Schweineschnitzel, Kinderschnitzel etc.; in der Terminologie entspricht dies Begriffen.
- Resources vom Typ `skos:Collection`: <Schnitzel BY hat_Herkunft>: Gruppen von Begriffen, die sich in den Werten eines gemeinsamen Merkmals unterscheiden;
- Resources vom Typ `skos:ConceptScheme`: der Kasten selbst des einleitenden Schnitzel-Beispiele, also der Thesaurus an sich.

Solche Typen wollen wir *Klassen* nennen, von denen wir in unserer Schnitzel-Terminologie dann auch einzelne Instanzen (Exemplare) anlegen können. (Vergleiche an dieser Stelle die Online-Fassung dieses Aufsatzes für konkrete Beispiele.)

Vorsicht Verwechslungsgefahr: Wir bauen keine Schnitzel-Ontologie und axiomatisieren auch nicht eine Klasse `kochen:Schnitzel`, die als Instanz ein konkretes Schnitzel haben kann. Sondern wir bauen eine Thesaurus-

Ontologie, in der wir eine Klasse `skos:Concept` axiomatisieren, die dann z. B. den Schnitzel-Begriff als Instanz haben kann. Nur wenn man die Klasse `kochen:Schnitzel` und die intellektuelle Konstruktion eines Begriffs `cpt:Schnitzel` als Instanz der Klasse `skos:Concept` analytisch als unterschiedliche Dinge erkennt, kann man auch über ihren Zusammenhang diskutieren. Ein solcher wird z. B. in der unten diskutierten Ontologie Ontolex-Lemon mit der Relation `concept/isConceptOf` hergestellt.

Ähnlich, wie man Klassen unterscheiden kann, werden in Ontologien typischerweise auch *Relationen* unterschieden. In einem Thesaurus sind das z. B. die Relationen `BT` (broader term, in SKOS: skos:broader) oder `NT` (narrower term, in SKOS: skos:narrower). Diese Relationen verhalten sich, wie man leicht sieht, *invers* zueinander: Wenn „Schweineschnitzel" als breiteren Begriff „Schnitzel" hat, dann hat umgekehrt auch „Schnitzel" als engeren Begriff „Schweineschnitzel".

Eine Ontologie dient nun dazu, solche Zusammenhänge sehr exakt festzuhalten. Dazu axiomatisiert man sie, um darauf aufbauend logische Schlüsse ziehen zu können (siehe Beispiel online).

In Axiomatisierungen liegt die Stärke von Ontologien, dafür werden sie entwickelt. Aus unserem Beispiel `BT` lässt sich auch noch mehr ablesen: Immer dann, wenn zwei Entitäten A und B mit BT in Beziehung gesetzt (z. B. `Schweineschnitzel BT Schnitzel`) sind, kann man daraus *ableiten*, dass A und B Entitäten vom Typ `concept` sind (siehe Beispiel online).

Bei Lesern mit einem fachlichen Hintergrund in Datenbanken könnte an dieser Stelle folgendes Vorwissen aktiviert werden: „Wenn A und B mit skos:broader verbunden sind, wäre es ein Fehler, wenn A und B nicht vom Typ skos:Concept sind!". In dieser Lesart würde man Definitions- und Wertebereich (Domain und Range) einer Relation als Integritätsbedingung behandeln: Wo immer ein skos:broader zwischen einem A und einem B behauptet wird, für die nicht explizit der Typ *concept* angegeben ist, müsste eine Datenbank-Engine einen fehlerhaften Datensatz melden.

Warnung: Die aus der Datenbanktechnik bekannte Interpretation eines Schemas als Integritätsbedingung ist in unserem Kontext *nicht* zutreffend! Fast so etwas wie das Gegenteil ist der Fall: Gerade dann, wenn die fragliche Typ-Information eben nicht explizit deklariert wurde, dient die Angabe von Domain und Range dazu, diese Information *ableiten*, *schlussfolgern* zu können. (Durch eine solche Logik wird übrigens kein neues Wissen erzeugt, sondern lediglich implizites Wissen explizit gemacht: Kreativ ist diese Logik nicht.)

4 WordNet

Während in SKOS vor allem zwischen Label (technisch als Literal realisiert) und skos:Concept unterschieden wird, kommt im (nur auf EN verfügbaren) Princeton WordNet noch die Idee des Synsets hinzu. Dieses steht in engem Zusammenhang mit Polysemie, also Wörtern mit ähnlichen, aber im Detail doch verschiedenen Bedeutungen. Wir betrachten den folgenden Satz:

„Our gas station around the corner is open seven days a week, day and night."

In diesem Satz haben die verschiedenen Vorkommen der Zeichenkette *day* offensichtlich verschiedene Bedeutungen. Wir schlagen *day* im Web-Frontend des Princeton WordNet nach und untersuchen die im Beispielsatz relevanten Bedeutungen genauer. Die ersten vier Einträge von Princeton WordNet -> Use Wordnet Online > „day" lauten so (siehe Abbildung 4):

Noun

- {15180180} S: (n) **day#1**, twenty-four hours#1, twenty-four hour period#1, 24-hour interval#1, solar day#1, mean solar day#1 (time for Earth to make a complete rotation on its axis) *"two days later they left"; "they put on two performances every day"; "there are 30,000 passengers per day"*
- {15148032} S: (n) **day#2** (some point or period in time) *"it should arrive any day now"; "after that day she never trusted him again"; "those were the days"; "these days it is not unusual"*
- {15182185} S: (n) **day#3** (a day assigned to a particular purpose or observance) *"Mother's Day"*
- {15190004} S: (n) **day#4**, daytime#1, daylight#1 (the time after sunrise and before sunset while it is light outside) *"the dawn turned night into day"; "it is easier to make the repairs in the daytime"*

Abbildung 4: WordNet > day

In diesem Beispiel ist die Zeichenkette „day" offensichtlich eine lexikalische Entität, und die technisch anmutenden Zeichenketten wie z. B. **day#1** (resp. **day%1:28:00::**) sind *sense numbers* (resp. *sense keys*) und damit offensichtlich so etwa wie semantische Entitäten. Was in SKOS mit einem *language-tagged string* notiert wird, wird in WordNet *word* genannt. Ein *word* kann mehrere Bedeutungen haben, die durch eine laufende *sense number* (resp. *sense key*) voneinander unterschieden werden.

Wir betrachten den Eintrag für **day#1** genauer. Ergänzend zur Erklärung „time for Earth to make a complete rotation on its axis" werden bei **day#1** auch noch andere *sense numbers* aufgezählt, u. A. *twenty-four hours#1, 24-hour interval#1* und auch mean solar day#1. Überrascht bemerken wir, dass *day#1* und *mean solar day#1* dieselbe Definition und dieselbe Database Location `{15180180}` aufweisen! Tatsächlich ist die Erklärung *time for Earth to make a complete rotation on its axis* nicht einer bestimmten *sense number*

(resp. einem *sense key*), sondern einer ganzen Gruppe von *sense numbers* zugeordnet.

WordNet unterscheidet sich von SKOS darin, dass Bedeutungen in Synsets zusammengefasst sind und Bedeutungsdefinitionen nicht *word senses*, sondern Synsets zugeordnet sind. Die 2006 publizierte W3C WordNet erläutert unter der Überschrift Introduction to the WordNet datamodel die Idee eines Synsets wie folgt:

> *The core concept in WordNet is the synset. A synset groups words with a synonymous meaning, such as {car, auto, automobile, machine, motorcar}. Another sense of the word „car" is recorded in the synset {car, railcar, railway car, railroad car}. Although both synsets contain the word „car", they are different entities in WordNet because they have a different meaning.*
>
> *More precisely: a synset contains one or more word senses and each word sense belongs to exactly one synset. In turn, each word sense has exactly one word that represents it lexically, and one word can be related to one or more word senses.*

Mit dem Unterschied von *word sense* und Synset begegnen wir in WordNet einer Differenzierung, die wir in SKOS so nicht kennen. Einerseits besteht die Synonym- und Antonym-Relation zwischen *word senses*, andererseits bestehen für Thesauri typische hierarchische Relationen wie Hyponym (Oberbegriff, skos:broader) oder Meronym (Teil-Ganzes, in SKOS nicht enthalten) nicht zwischen *word senses*, sondern zwischen Synsets (!). Auch die normalsprachliche Erklärung eines Begriffs sowie exemplarische Verwendungszusammenhänge sind, wie wir gesehen haben, in WordNet nicht auf Wörter- oder *word sense, sondern* auf Synset-Ebene untergebracht. (Siehe Beispiel online.)

Wordnets gibt es in vielen Sprachen, siehe GWA WordNets. Während das kostenpflichtige Tübinger GermaNet seit 1996 Jahren manuell (!) stetig erweitert wurde und so eine sehr hohe Qualität besitzt, wird das frei verfügbare OdeNet durch Datenintegration aus Open-Source-Quellen weitgehend automatisch aufgebaut. Die Pluralität verschiedener Wordnets ist dabei kein Problem: Auch die Daten des Princeton WordNet sind als RDF-Datensatz (WordNet RDF) erhältlich. Und ja, natürlich wird die Semantik der dort enthaltenen Relationen durch einen Verweis auf eine Ontologie geklärt, in diesem Fall auf die Ontologie OntoLex-Lemon, deren wichtigste Konzepte sich wie in OntoLex Lemon (siehe Abbildung 5) visualisieren lassen.

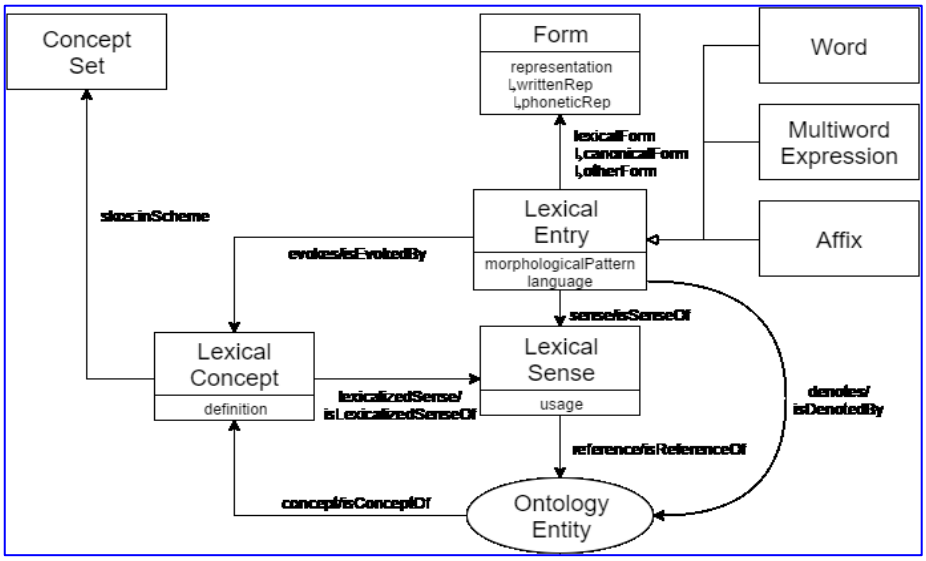

Abbildung 5: OntoLex Lemon, Quelle: https://www.w3.org/2016/05/ontolex/Lemon_OntoLex_Core.png

OntoLex-Lemon ist ein schönes Beispiel für den Mehrwert einer Ontologie: Solange die verschiedenen Wordnets dieselbe Ontologie nutzen, kann das jeweils genutzte spezifische Format vergleichsweise frei gewählt werden.

5 Was ist eine Ontologie?

Mit SKOS haben wir exemplarisch eine Ontologie kennengelernt, mit der Klassen wie skos:Concept, skos:ConceptGroup und skos:ConceptScheme axiomatisiert werden. Die Ontologie SKOS schafft eine axiomatisierte begriffliche Grundlage, um den Aufbau und Austausch eines Thesaurus zu ermöglichen. *Tom Gruber* beschreibt den Begriff der Ontologie so:

> *An ontology is an explicit specification of a conceptualization. […] In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, an ontology is the statement of a logical theory. (Gruber 1995)*
>
> *Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies; in practice, the languages of ontologies are closer in expressive power to first-order logic than languages used to model databases. (Gruber 2008)*

Wir wollen Gruber folgen und als Konzeptualisierung ein formal-logisches Modell der Klassen und Relationen verstehen, mit deren Hilfe wir in einer „Domain of Interest" unser Wissen modellieren. Solange wir es als

Terminologen primär mit Begriffen und Benennungen zu tun haben, bietet uns SKOS die gesuchte Konzeptualisierung von Begriffen wie „Begriff" oder „Benennung". Sind wir außerdem auch noch lexikalisch interessiert, wird OntoLex-Lemon interessant.

Es stellt sich die Frage, ob auch Wissensrepräsentationen, die keine formal-logisch axiomatisierten Begriffe enthalten, Ontologien sind. Relevant ist die Frage natürlich insbesondere für unseren SKOS-Thesaurus selbst, aber auch für Semantische Netze, viele diagramm-basierte Wissensrepräsentationen, allgemein für alle mehr oder weniger differenzierte Wissensdatenbanken, insbesondere auch Terminologien. Weitgehender Konsens besteht darin, dass insbesondere „Glossar", „Thesaurus" und „axiomatische Theorie" eine Ordnung von ontologisch zunehmend präziseren Wissensrepräsentationen bilden (hier z. B. Guarino 2006, entnommen aus ISKO Biagetti 2021, siehe Bsp. für SKOS ABox, hier „Economic cooperation" – siehe Abbildung 6):

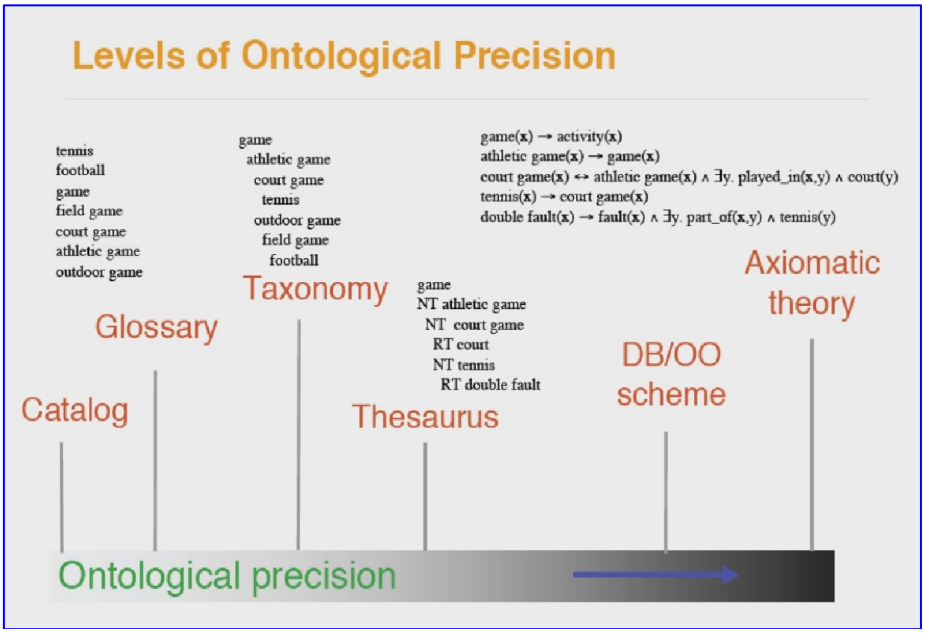

Abbildung 6: Beispiel für SKOS ABox, hier „Economic cooperation".
Quelle: SKOS Core Guide (2005)

Gemäß dieser Ordnung ist die TBox von SKOS selbst eine Ontologie, eine SKOS-ABox jedoch lediglich ein Thesaurus. SKOS konzeptualisiert die Begriffe, mit denen wir einen Thesaurus bauen, aber nicht die Konzepte aus der Domänen-Terminologie selbst.

6 Exkurs: Was ist eine Konzeptualisierung?

Es stellt sich sofort die Frage, was unter einer Konzeptualisierung zu verstehen ist. Sowohl textimmanent in den zitierten Aufsätzen wie auch aus dem Verständnis der Forschungshistorie heraus lässt sich feststellen, dass Ontologien in formaler Logik notiert und durch *Reasoner-* oder *Inferencing-Systeme* in Verbindung mit einer Logik-Programmiersprache – früher oft Prolog, heute z. B. RDF(S) oder OWL, andere spielen derzeit kaum keine Rolle – auswertbar sein müssen, also durch logisches Schlussfolgern implizites Wissen explizit gemacht werden kann.

Die Axiomatisierung kann, wie oben gezeigt, über Axiome aus RDF(S) wie z. B. `rdfs:domain` und `rdfs:range` erfolgen. Für komplexere Axiomatisierungen verwendet man heute OWL2-Elemente aus verschiedenen OWL language profiles, darunter insbesondere Axiome vom Typ *Restriction*. OWL2 ist eine mächtige Logiksprache, die zu beherrschen allerdings ein erhebliches formal-logisches Hintergrundwissen voraussetzt.

7 Will man eine Ontologie selber bauen?

In OWL eine Ontologie zu formulieren und mit anderen Ontologien zu vernetzen *ist* „rocket science". Schon SKOS zu bauen (und mit SKOS-XL vorsichtig zu erweitern) hat sich in den 2000er-Jahren lange hingezogen und hat viel Expertise erfordert. Eine große Herausforderung jeder neuen Ontologie besteht darin, die Interoperabilität und Komplementarität mit existierenden Ontologien aufrechtzuerhalten. Selbst wenn existierende Ontologien die eigene Weltsicht nur teilweise abbilden: Will man die eigene Weltsicht wirklich mit einer eigenen Ontologie konkurrierend auf den Markt bringen?

Vor allem aber verkörpern Ontologien auch philosophische Sichtweisen auf die Welt – insbesondere auch solche, die nicht kompatibel, sondern konkurrierend sind. Die Recherche nach geeigneten Bezugsontologien ist mühsam, da der formale OWL-Code von Ontologien ohne eine gute – und das heißt auch: anschauliche – Dokumentation kaum verständlich ist. Existierende Ontologien als Bezugspunkt auszuwählen erfordert insbesondere auch ein Verständnis der philosophischen Ontologie der jeweiligen Bezüge. Der vorliegende Aufsatz versucht, zu einem solchen Verständnis mit Bezug auf SKOS und WordNet beizutragen, die für Terminologen wichtige Bezugspunkte sind.

Das Verzeichnis Linked Open Vocabularies (LOV) verzeichnet knapp 800 (Stand November 2022) als Ontologie axiomatisierte Vokabulare, u. A. auch SKOS. Hier werden tausende von Begriffen und Relationen z. T. rudimentär, z. T. sehr komplex axiomatisiert. Damit liegt de facto eine modularisierte und verteilte, oft überraschend durchdachte, nicht immer leicht verständliche und nicht immer konsistente Open-Source-Terminologie bereits vor.

8 Wozu Ontologien?

Auch wenn man keine eigene Ontologie herstellen will, bietet das Ökosystem „Semantic Web" einiges an Mehrwert. Wenn man ein firmeninternes Datenschema auf die TBox existierender Ontologien mappt und/oder die eigenen (hier: Terminologie-)Daten als Instanzen von Klassen aus existierenden Ontologien anlegt, macht man diese Daten für Zweit- und Mehrfachnutzungen nutzbar. Ist man zusätzlich an Interoperabilität interessiert, nimmt man idealerweise auf kleine, benutzbare, anerkannte Ontologien Bezug.

Wann immer man in einem Knowledge Graph über den Typ eines Knotens (z. B. „Schnitzel": Das Schnitzel „an sich"? Eine Klasse in einer Nahrungsmittel-Ontologie? Ein Begriff in einem Thesaurus? Ein mehrdeutiges Wort in einem Text?) oder die ableitbare Domain und Range einer Relation Auskunft geben will, kann man diesen Knoten zu einer öffentlichen Ontologie in Bezug setzen – falls man damit leben kann, wie der fragliche Begriff in der jeweiligen Ontologie axiomatisiert wurde. Alternativ zur anspruchsvollen Strategie, auf einen in einer Ontologie axiomatisierten Begriff zu verweisen, bietet sich auch die schwächere Strategie an, lediglich auf Begriffe aus einer nicht formal axiomatisierten Terminologie, sondern z. B. in SKOS formulierten Terminologie zu verweisen.

SKOS bietet sich insbesondere auch dann als erste Wahl an, wenn man in der Praxis eine eigene Terminologie erstellen will: Anders als in OWL formulierte Axiomatisierungen sind in SKOS formulierte Terminologien fehlertolerant und einfach genug, um von Nicht-Logikern aufgebaut und adäquat gepflegt werden zu können – eben weil ein SKOS-Thesaurus keine Ontologie ist, sondern „nur" ein Wissensgraph auf Instanzebene.

9 Quellen

Busse, J. (2014): „Semantische Modelle Mit Mindmaps". In: Keller, S. A., Schneider, R. und Volk, B. (Hrsg.): Wissensorganisation und -repräsentation mit digitalen Technologien. DeGruyter 2014, S. 115-127, https://www.degruyter.com/document/doi/10.1515/9783110312812.115/pdf.

Busse, J. (2022): „Kernkonzepte der Taxonomiesprache GenDifS", In: Eggert, S. et al. (Hrsg.): AKWI 2022: Tagungsband zur 35. Jahrestagung des AKWI, 2022 DOI: https://doi.org/10.30844/AKWI_2022_14. Präsentation auf dem AKWI 2022: http://jbusse.de/gendifs/akwi2022.html.

Cimiano, P. / McCrae, J. P. / Buitelaar, P. (2016): Lexicon Model for Ontologies: Community Report, Final Community Group Report 10 May 2016 https://www.w3.org/2016/05/ontolex/.

Gruber, T. R. (1995): „Toward Principles for the Design of Ontologies Used for Knowledge Sharing". In: International Journal Human-Computer Studies, 43(5-6):907-928, 1995.

Gruber, T. (2008): „Stichwort Ontology". In: Liu, L. / Özsu, M. T. (eds.): Encyclopedia of Database Systems, Springer 2020. <DOI 10.1007/978-1-4899-7993-3_1318-2>, online auch: https://tomgruber.org/writing/ontology-in-encyclopedia-of-dbs.pdf.

Guarino, N. (2006): „Ontology and Terminology: how can formal ontology help concept modeling and terminology?". In: EAFTNordTerm on Terminology. Concept Modeling and Ontology. Vaasa, February 10th, 2006, slide no 13.

Biagetti, M. T. (2021): „Ontologies as knowledge organization systems". In: Knowledge Organization 48, no. 2: 152-176, 2021. Also available in ISKO Encyclopedia of Knowledge Organization, eds. Birger Hjørland and Claudio Gnoli, <https://www.isko.org/cyclo/ontologies>.

Smith, A. (2022): Simple Knowledge Organization System (SKOS). 2022-07-18 <https://www.isko.org/cyclo/skos>.

McCusker, J. et. Al. (2012): „Functional Requirements for Information Resource Provenance on the Web". 52-66. 10.1007/978-3-642-34222-6_5. <https://www.researchgate.net/publication/262369023_Functional_Requirements_for_Information_Resource_Provenance_on_the_Web (pdf)>.

Siegel, M. (2018): „Odenet. Ein deutscher Beitrag zur „Multilingual Open Wordnet Initiative". <https://ontologenkreis.org/docs/2018-01/201801_OdeNet.pdf>.

Siegel, M. / Bond, F. (2021): „OdeNet - Compiling a German Wordnet from other Resources." 11th International Global Wordnet Conference (GWC2021)At: South Africa, January 2021, https://www.aclweb.org/anthology/2021.gwc-1.22

W3C WordNet = van Assem, M. et al. (2006: RDF/OWL Representation of WordNet, W3C Working Draft 19 June 2006. https://www.w3.org/TR/wordnet-rdf/.

Der vorliegende Aufsatz liegt unter http://jbusse.de/dtt2023/ auch in einer erweiterten und aktualisierten Web-Fassung vor, die auch weitere Quellen, ein Glossar sowie vertiefende Diskussionen enthält.

Prof. Dr. Johannes Busse
HAW Landshut
Am Lurzenhof 1
84036 Landshut
busse@haw-landshut.de