Semantische Netze, Ontologien und die verbindende Formale Begriffsanalyse (FBA)

© C. Lübbert / T. Zeh, Arbeitspapier V5.4-kurz, Juni 2016
Vorgetragen auf dem Ernst-Schröder-Seminar, TU-Darmstadt, 18.06.2016

1 Einführung

Die Konzepte **„Semantisches Netz“** und **„Ontologie“** dienen als Mittel zur ***Formalisierung*** (und damit Idealisierung) eines vorhandenen ***Wissensgebietes***, so dass dieses auch von ***Computerprogrammen*** verarbeitet, durchsucht, befragt und weitergereicht werden können soll. Dies wollen wir hier ***„mathemaitsch fundiert“*** darstellen.

Das Werkzeug „**Semantisches Netz**“ ist das einfachere; das der „**Ontologie**“ das komplexere. Letzteres besteht aus zwei Ebenen: Der ***Schema-Ebene*** (O-Schema) und der ***Instanzen-Ebene***. Aber sowohl die Schema-Ebene als auch die Instanzenebene einer Ontologie kann jeweils durch ein *Semantisches Netz* dargestellt werden.

Wir glauben, dass zur Ermittlung der **Schema-Ebene** (O-Schema) einer Ontologie die **Formale Begriffsanalyse (FBA)** ein ***nützliches Bindeglied*** hergibt, und zwar durch eine **„bottom-up" Methode**, die (bei Bedarf) **iterativ** wiederholt wird mit folgendem Zyklus in vier Schritten:

Schritt-1: Erstelle beispielhaft einige einschlägige Info aus dem Wissensgebiet als ein **Semantisches Netz SN1** und nimm SN1 als Ausgangspunkt.

Schritt-2: Aus SN1 erstelle gemäß **FBA** zwei **Formale Kontexte**; einen Kontext **KV** für die „**Knoten**" und ihre „**Attribute**"; einen zweiten Kontext, **KE,** für die **Knotenpaare** und ihre „**Beziehungstypen**".

Schritt-3: Mit Hilfe von **FBA**-Tools erstelle die zugehörigen **Formalen Begriffsverbände** $\mathbb{B}$(**KV**) und $\mathbb{B}$(**KE**); diese geben bereits entscheidende Hinweise auf das **O-Schema** für das Wissensgebiet.

Schritt-4: Erstelle ein **O-Schema** der Ontologie zum Wissensgebiet als ein **SN2**: Aus $\mathbb{B}$(**KV**) liest man bereits die **Klassen** und ihre **Klassentaxonomie** ab und vergibt nur sprechende Namen für die **Klassen**. Aus $\mathbb{B}$(**KV**) liest man die **Relationstypen** und ihre **Relationstypen-Taxonomie** ab.

Prüfung: Dieses O-Schema eignet sich gut zur Überprüfung und ggf, Korrektur / Ergänzung in einem weiteren Iterationsschritt bei der Formalisierung des Wissensgebiets.

Die gefundenen Korrekturen / Ergänzungen führen zu einem **nächsten Zyklus** der Iteration (Schritt-1 bis Schritt-4). – Solange, bis man mit der Version von **SN1** und **SN2** in dem Sinne „zufrieden" ist, dass sie das Wissensgebiet hinreichend angemessen wiedergeben.

Zunächst ist zu klären:

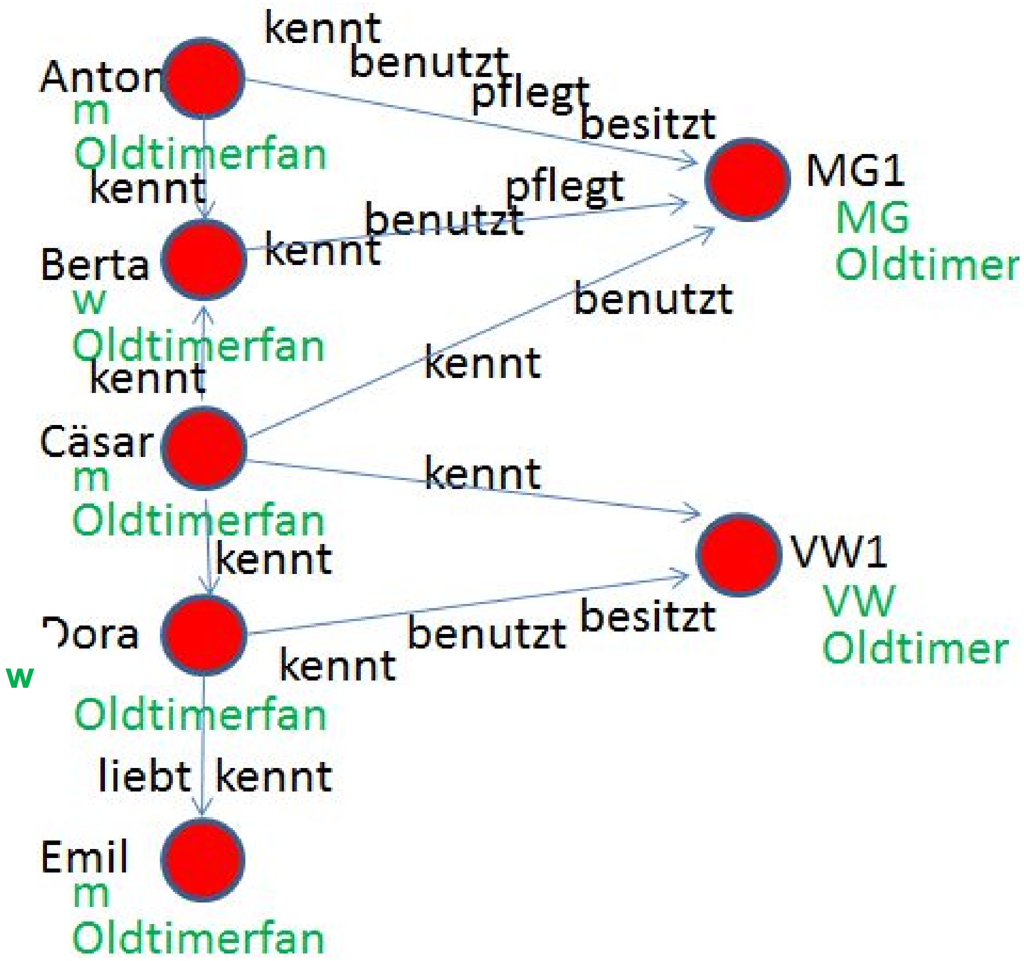
- Was verstehen wir unter einem „Semantischen Netz"?
- Was verstehen wir unter einer „Ontologie"?

Und Schließlich:

- Wie gestaltet sich die verbindende Rolle der **FBA** bei der eben angedeuteten (iterativ anwendbaren) **„bottom-up"-Methode** zur Ermittlung letztendlich akzeptierten **Schema-Ebene** (O-Schema) einer Ontologie des Wissensgebiets?

2 Was ist ein „Semantisches Netz“?

Das sei eine Struktur ***SN*** **:= (V, E, BT, A_V, $\underline{I}_E$, α, W)** -- Beispiel: „**Oldtimer-Fan-Club**“.



- **V** Menge von „**Knoten**“, z.B. **V**= {Anton,…, Emil, MG1, VW1}. (**rot**).
- **A_V** Menge von „**Knotenattributen**“, z.B. **A_V** = {Oldtimerfan, m, w, Oldtimer, MG1, VW1} (**grün**).
- α:**V**→Pot**A_V** **„Knotenattributierung“,** charakterisiert jeden Knoten x∈**V** durch eine Attributmenge α(x)⊂ **A_V**.
- **W** Menge von „Attribut***werten***“. Ist x∈**V**, a∈α(x), so sei a(x)∈**W** der Wert, der dem zum Knoten x gehörigen Attribut a zukommt.
- **E**⊂V×V Menge gerichteter Kanten ~„Pfeile“ (**blau**) zwischen zwei Knoten.
- **BT** Menge von ***binären*** „**Beziehungstyen**“ zwischen Knoten, z.B. **BT**= {kennt, benutzt, pflegt, besitzt, liebt}.
- **$\underline{I}_E$** ⊂ **E×BT** die „Belegungsrelation“: welcher Pfeil (x, y)∈**E** ist mit welchen Beziehungstypen „belegt“.

3 Was ist eine „Ontologie“?

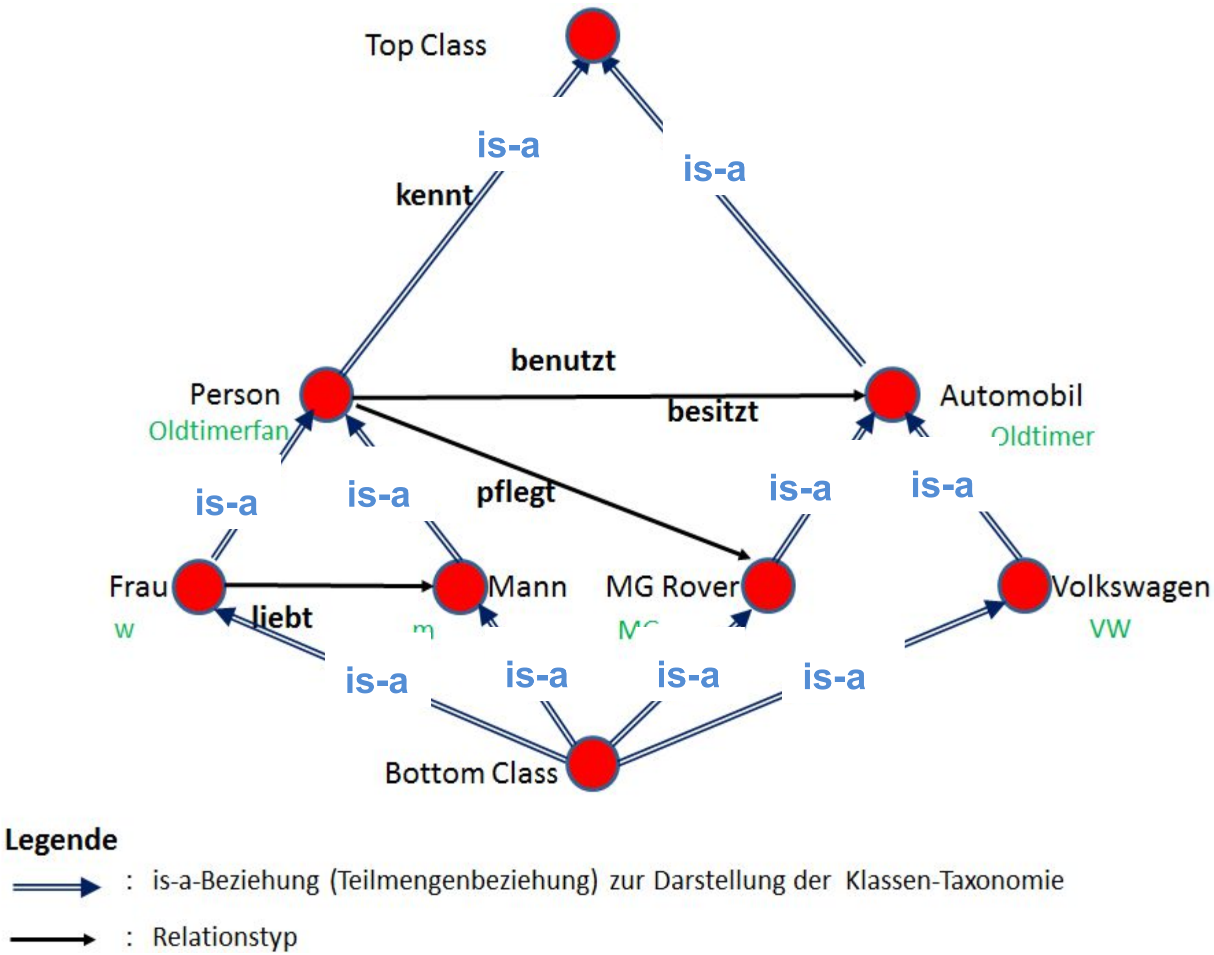
Eine Ontologie erfüllt eine ähnliche Aufgabe wie ein Semantisches Netz. Ihre Struktur ist aber etwas komplexer. Wir teilen sie ein in zwei Teile: die „**System-interna**“ und die „**Systemexterna**“. Die „Systeminterna“ einer Ontologie bestehen aus zwei Ebenen:

- Die *abstraktere* Ebene, die **Schema-Ebene** (kurz: „**O-Schema**“), besteht aus „**Klassen**“ (auch „Typen“ oder „Begriffe“ genannt), ihren „**Attributen**“, sowie weiterhin *binären* „**Relationstypen**“ zwischen Klassen.
- Die *konkretere* Ebene, die „**Instanzen-Ebene**“ besteht aus „**Instanzen**“ (~ Ausprägungen von Klassen) und „**Aussagen**“ über Instanzen (~ Ausprägungen von Relationstypen).

Über den Teil der „System***externa***“, der sozusagen die „*Benutzerschnittstelle*“ einer implementierten Ontologie darstellt, sagen wir hier aus Zeitgründen nichts und beschäftigen uns im Folgenden nur mit den ***Systeminterna*** einer Ontologie.

3.1 Was ist die „Schema-Ebene“ (O-Schema) einer Ontologie?

Das sei eine Struktur **$\mathbf{OS}$:= ($\mathbb{C}$, $\leq_c$, $\mathfrak{A}$, *att*, $\mathbb{RT}$, *prop*, *Vererbung*)** – Beispiel: ein **„Oldtimer-Fan-Club“.**



- $\mathbb{C}$ Menge von „**Klassen**“ („Typen“) (**rot**).
- $\leq_c$ eine *vorab* gegebene ***Ordnung*** auf $\mathbb{C}$, genannt „**Klassentaxonomie**“, auch mit „***is_a***“ bezeichnet.
- $\mathfrak{A}$ Menge von „**Attributen**“ (**grün**).
- ***att***: $\mathbb{C} \rightarrow \mathrm{Pot}\mathfrak{A}$ charakterisiert jede Klasse C durch die Menge $\mathrm{attC} \in \mathfrak{A}$ ihrer **Attribute**.
- $\mathbb{RT}$ Menge von weiteren *binären* „**Relationstypen**“ zwischen Klassen.
- ***prop*** : $\mathbb{RT} \rightarrow \mathbb{C} \times \mathbb{C}$ die „**Aufhängungs-funktuon"**, die jedem Relationstyp $r \in \mathbb{RT}$ das „minimale“ Klassenpaar (C, D) = ***prop***(r) zuordnet. (Genaueres → später)
- **Vererbung** von Attributen u. Relationstypen.

Vererbung von Attributen: Attribute werden entlang der Klassentaxonomie „vererbt“, das soll heißen: Ist C eine echte Unterklasse von D, $C <_c D$, und $a \in attD$, so auch $a \in attC$. Attribute benutzen wir ***ausdrücklich*** dazu, um Klassen im geordneten System ($\mathbb{C}$, $<_c$) zu **unterscheiden**, d.h., wir fordern:

(1) attC = attD ⇒ C=D und **$C <_c D \Rightarrow attD \subset attC$.**

Wir fordern sogar folgende „Vollständigkeitseigenschaft“ des Systems ($\mathbb{C}$, $<_c$):

(2) Zu je zwei Klassen C, D$\in\mathbb{C}$ gibt es eine Klasse E$\in\mathbb{C}$, für die attE = attC ∩ attD gilt. (Oft ist E einfach die „leere Klasse“, d.h. die nichtssagende BottomClass.)

Anmerkung: Die Eigenschaften (1), (2) werden *nicht von allen Ontologie-Entwicklern anerkannt!* Attribute scheinen bei ihnen eher „nur so nebenher“ aber nicht immer zur Unterscheidung der Klassen eingeführt zu werden. Eine Unterklasse dürfte bei ihnen dieselben Attribute wie eine ihrer echten Oberklassen haben. Da fragen wir uns aber: durch was sollen sich denn die Klassen im O-Schema der System-interna einer Ontologie unterscheiden?

Vererbung von Relationstypen: Auch Relationstypen werden entlang der Klassentaxonomie „vererbt“. Ist zwischen den Klassen C, D$\in\mathbb{C}$ ein Relationstyp r $\in$ $\mathbb{RT}$ *aufgehängt*, also prop(r) = (C, D), und sind $C' \leq_c C$, $D' \leq_c D$ jeweils (echte oder unechte) Unterklassen, so gibt es einen Relationstyp r‘ $\in \mathbb{RT}$ mit prop(r‘) = (C‘, D‘). r‘ ist dann der von r her „vererbte“ Relationstyp. Genaueres: → Instanzenebene.

Relationstypen-Taxonomie: Mit Hilfe der Klassentaxonomie $<_{\mathbb{C}}$ und der Vererbung ergibt sich auch für $\mathbb{RT}$ eine ***Ordnung*** $\leq_{\mathbb{RT}}$, die wir die **Relationstypen-Taxonomie** nennen. Genaueres: → Instanzenebene.

Anmerkung: Die Relationstypen-Taxonomie $\leq_{\mathbb{RT}}$ wird bei konventionellen Ontologie-Entwürfen meist nicht beachtet. Wir sehen sie aber – sofern überhaupt eine Menge $\mathbb{RT}$ im O-Schema von Bedeutung ist – als ebenso relevant an wie die Klassentaxonomie $\leq_{\mathbb{C}}$ (**is-a**)

3.2 Was ist die Instanzenebene zum O-Schema?

Das ist eine Struktur $\mathbb{OI}$:= ($\mathbb{OS}$, **IN**, *inst*, *instr*, *instw,* **W**) ($\mathbb{OS}$ = O-Schema wie oben)

• **IN** sei die Menge aller in der Ontologie ermittelbaren **Instanzen** (~ Ausprägungen von Klassen); zur meist nichtssagenden TopClass gehören ***alle*** Instanzen von **IN**.

• *inst:* $\mathbb{C} \rightarrow$ Pot**IN** ordnet jeder Klasse C $\in \mathbb{C}$ die Menge $\underline{C}$:=inst(C) der **Instanzen** zu; das sind die Ausprägungen der Klasse C. Für die Instanziierung der Klassen des O-Schemas gelten bei den Ontologie-Entwerfern folgende Regeln meist unwidersprochen:

(3) $\quad$ **C = D** $\Leftrightarrow$ ***inst*(C) = *inst*(D)** $\quad$ und $\quad$ **C<$_c$ D** $\Leftrightarrow$ ***inst*(C) $\subset$ *inst*(D).**

Das heißt: Zwei Klassen sind als gleich anzusehen, wenn sie dieselben Instanzen haben. Ist C echte Unterklasse von D, so ist C „spezieller“ als D und hat daher echt *weniger* Instanzen als D. Gilt C<$_c$ D, so gehört jede Instanz der Klasse C auch zur Klasse D. Jede Instanz x$\in\underline{D}$ hat alle Attribute der Klasse D; gehört x aber auch zu einer echten Unterklasse von D, so hat -- wegen (1) -- x *mehr* Attribute als die von D. Mit **att**(x) bezeichnen wir die Menge ***aller*** Attribute der Instanz x. **att**(x) ist die Attributmenge der ***kleinsten*** Klasse (im Sinne der Klassentaxonomie <$_c$), zu der die Instanz x noch gehört. Mit der Forderung (2) ergibt sich schließlich:

(4) $\quad$ **Zu je zwei Klassen C, D$\in\mathbb{C}$ gibt es eine Klasse E$\in\mathbb{C}$, für die inst(E) = inst(C) $\cap$ inst(D) gilt.**

• *instr:* $\mathbb{RT} \rightarrow$ Pot(**IN×IN**) ordnet jedem Relationstyp r$\in\mathbb{RT}$ die **Relation $\underline{\mathbf{r}}$**:=instr(r) = {(x, y)$\in$**IN×IN** | r°(x, y)} zu, wobei r° die zu r gehörige ***Aussageform*** ist. Es gilt also (x, y) $\in$ $\underline{\mathbf{r}}$ genau dann, wenn die ***Aussage*** r°(x, y) für die Instanzen x, y$\in$**IN** ***zutrifft***.

• *instw:* $\mathbb{A}\times\mathrm{inst}(\mathbb{C})\rightarrow\mathbf{W}$ leistet folgendes: ist $\mathbf{a}\in \mathrm{att}C\subseteq\mathbb{A}$ ein Attribut der Klasse C und $x\in\underline{C}$ eine Instanz der Klasse C, so ist $\mathbf{a}(x) := \mathrm{instw}(\mathbf{a}, x) \in \mathbf{W}$ der ***Wert***, den das Attribut a für die Instanz x annimmt.

Mit Hilfe der Instanzenebene können wir nun auch die **Relationstypen-Taxonomie** $\leq_{\mathbb{RT}}$ und die **Aufhängugsfunktion *prop*** genauer definieren:

(i) Zur Menge $\mathbb{RT}$ der Relationstypen bilden wir die ***Relationen***menge $\underline{\mathbb{RT}} \coloneqq \mathrm{instr}(\mathbb{RT}) \coloneqq \{\underline{\mathbf{r}} \mid r\in\mathbb{RT}\}$. Diese ist durch die Mengeninklusion $\subseteq$ in natürlicher Weise geordnet. Übertragen wir diese Ordnung auf $\mathbb{RT}$ selbst und nennen sie dort „$\leq_{\mathbb{RT}}$“, so ist $(\mathbb{RT}, \leq_{\mathbb{RT}})$ ebenfalls eine geordnete Menge, und „$\leq_{\mathbb{RT}}$“ ist die gewünschte **Relationstypen-Taxonomie.**

(ii) Bilden wir zu jedem Relationstyp $r\in\mathbb{RT}$ die Mengen

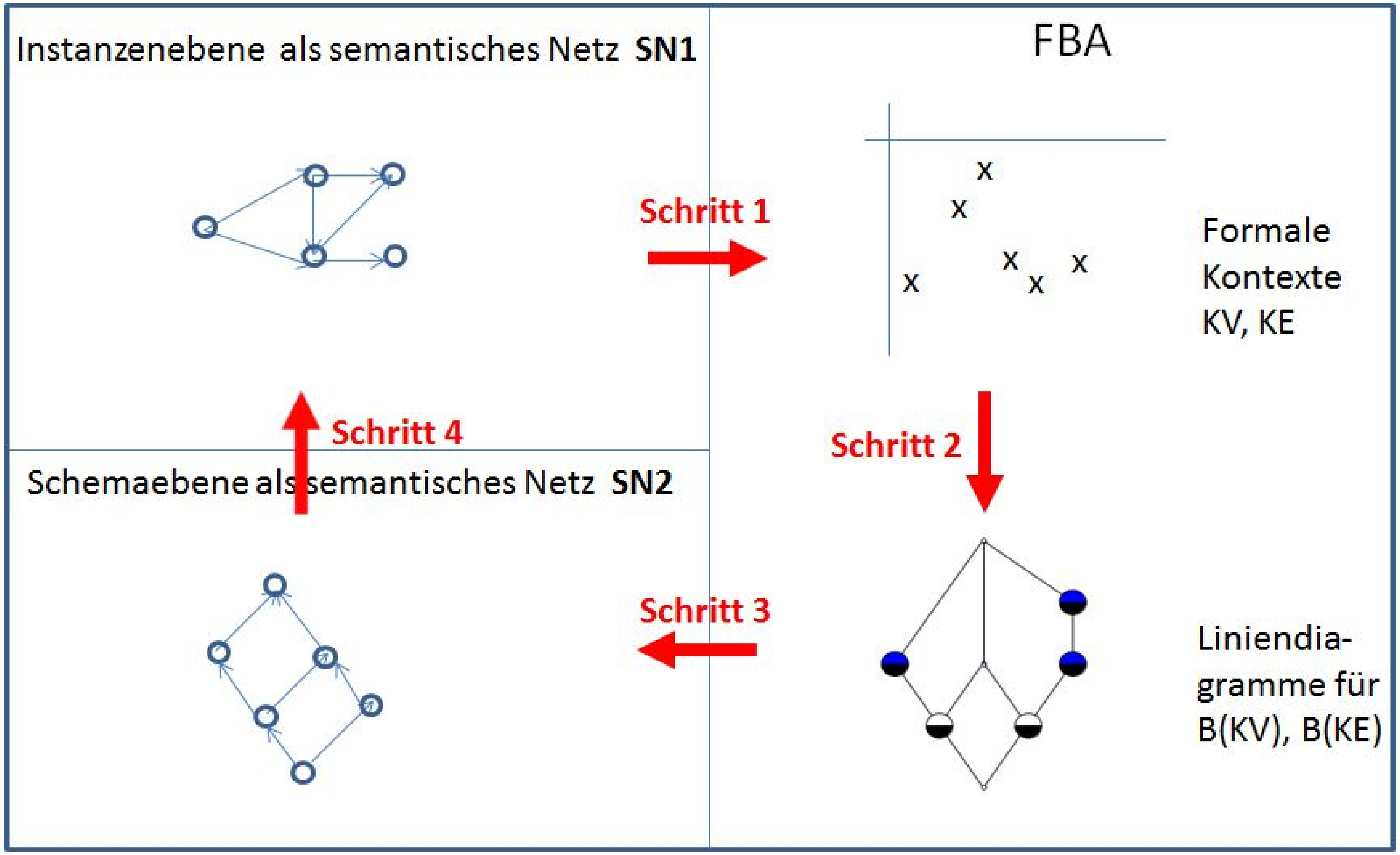
$$dom(\underline{\mathbf{r}}) := \{x\in\mathbf{IN} \mid \exists y\in\mathbf{IN}: {}^\circ r(x, y)\}, \quad range(\underline{\mathbf{r}}) := \{y\in\mathbf{IN} \mid \exists x\in\mathbf{IN}: {}^\circ r(x, y)\},$$

so bedeutet die Gleichung *prop*(r)=(C, D) Folgendes:

(5) *prop*(r) = (C, D) :⇔ im Sinne der Klassentaxonomie $\leq_C$ ist C die *kleinste* Klasse mit dom($\underline{r}$) $\subseteq$ $\underline{C}$ und D die *kleinste* Klasse mit range($\underline{r}$) $\subseteq$ $\underline{D}$. Damit ist die „Aufhängungsfunktion“ ***prop*** präzise definiert.

4 Was ist die verbindende Rolle der FBA?

Wir demonstrieren das anfangs angedeutete **bottom-up-Verfahren** zur Ermittlung des **O-Schemas** einer **Ontologie** nun im Detail. Hier noch einmal ein Übersichtsbild für die iterierbaren Schritte 1 bis 4:



4.1 Schritt-1

SN1:

Ermittle beispielhaft einschlägige Info aus dem „Oldtimer Club“:

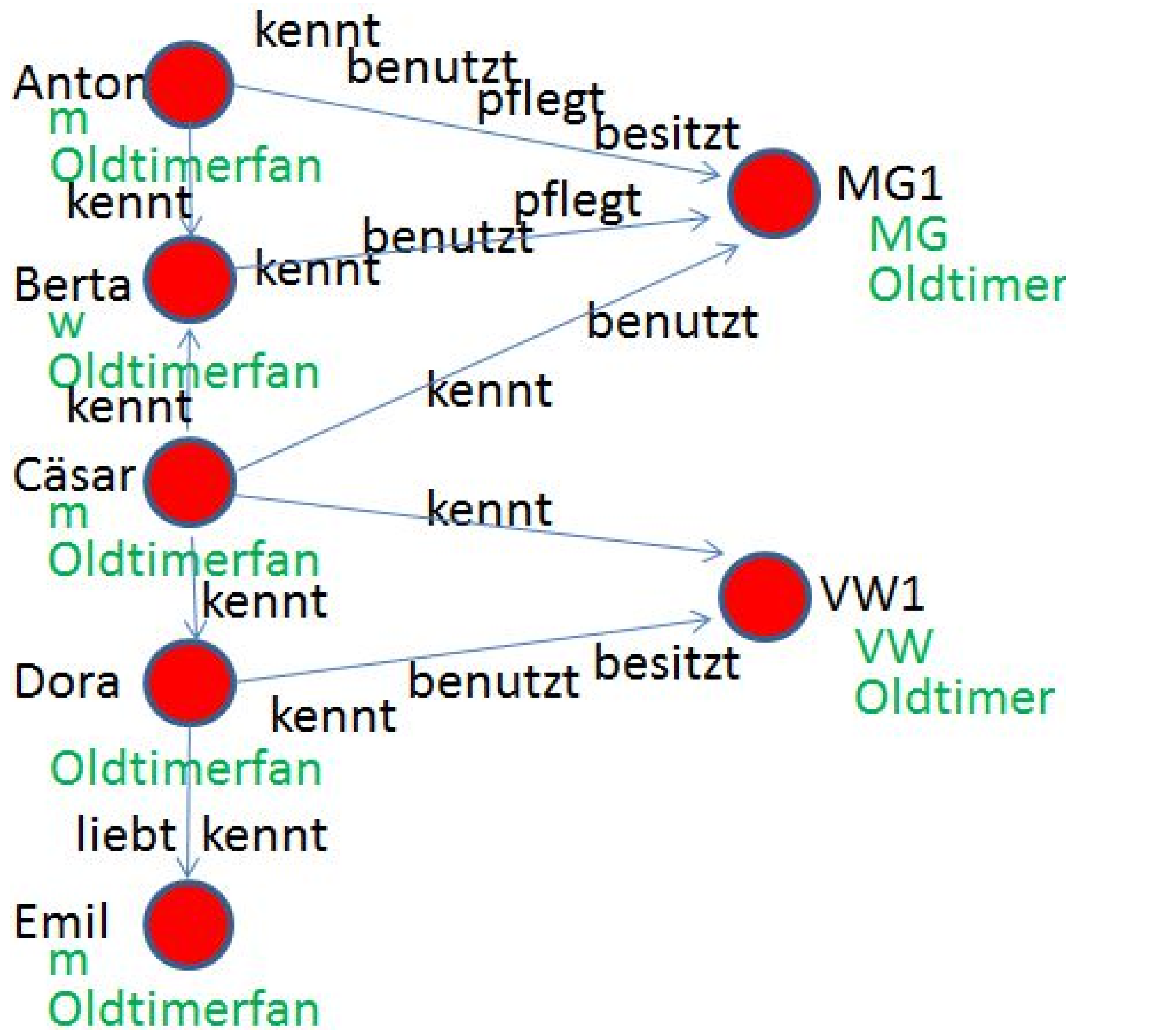
Knotenmenge:
V = {Anton,…, Emil, MG1, VW1},

Attributmenge:

$\mathbf{A_v}$ = {Oldtimerfan, m, w, Oldtimer, MG, VW},

Beziehungstypenmenge:
BT = {kennt, benutzt, pflegt, besitzt, liebt},

und erstelle damit das Semantische Netz **SN1,** welches die **Instanzenebene** beispielhaft repräsentiert.



4.2 Schritt-2

Erstelle den formalen **Knotenkontext**

KV = (**V**, $\mathbf{A}_V$, $\underline{\mathbf{I}}_V$)
mit der Inzidenzrelation $\underline{\mathbf{I}}_V$ definiert durch
$(x, a) \in \underline{\mathbf{I}}_V \Leftrightarrow a \in \alpha(x)$
für $x \in \mathbf{V}$, $a \in \mathbf{A}_v$

A	B	C	D	E	F	G
	Oldtimerfan	m	w	Oldtimer	MG	VW
Anton	X	X				
Berta	X		X			
Cäsar	X	X				
Dora	X		X			
Emil	X	X				
MG1				X	X	
VW1				X		X

Erstelle den formalen **Kantenkontext**

KE := (**E**, **BT**, $\underline{\mathbf{I}}_E$) mit der Inzidenzrelation $\underline{\mathbf{I}}_E$ definiert durch
$(x, y; r) \in \underline{\mathbf{I}}_E \Leftrightarrow r(x, y)$
für $(x, y) \in \mathbf{E}$, $r \in \mathbf{BT}$

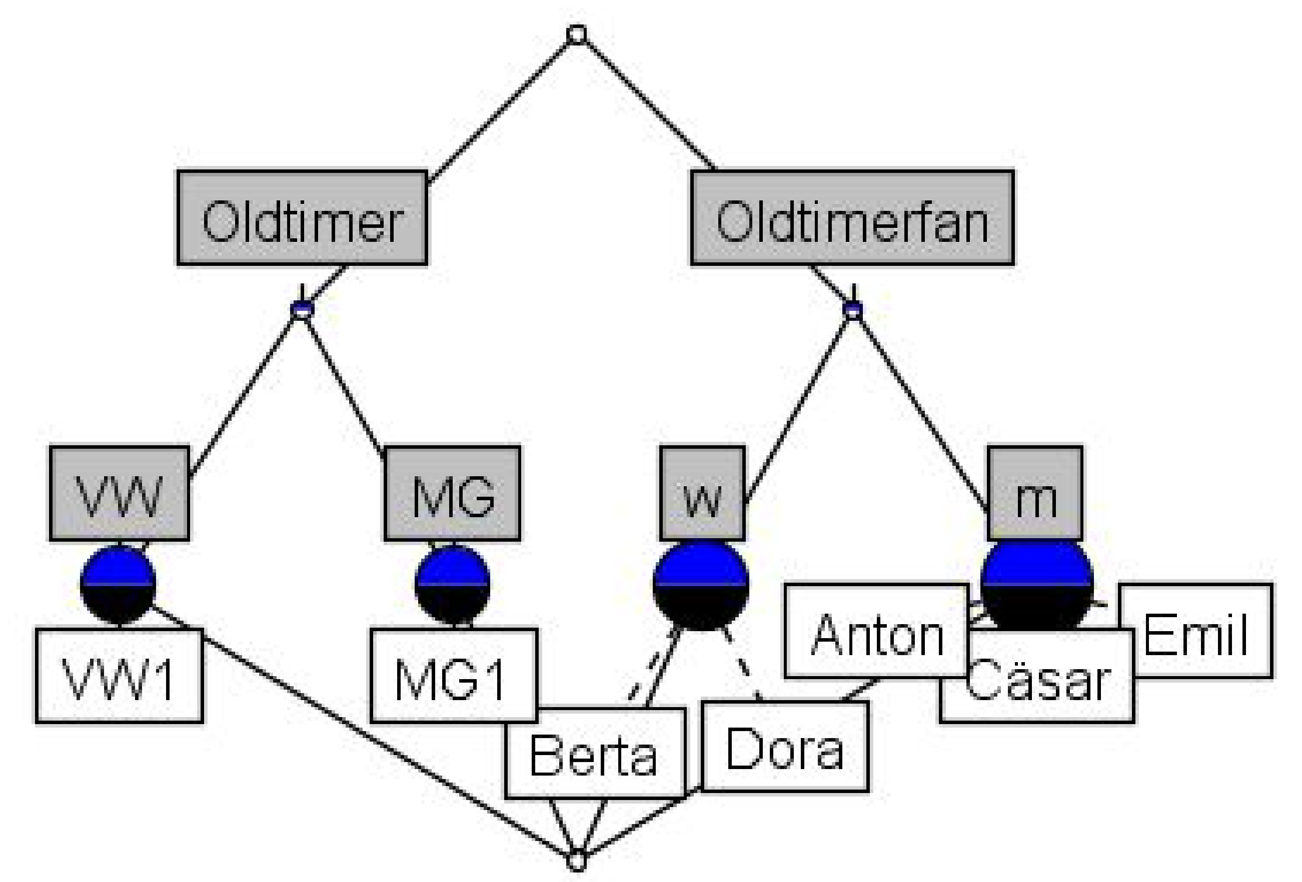
A	B	C	D	E	F
	kennt	benutzt	besitzt	pflegt	liebt
Anton-MG1	X	X	X	X	
Berta-MG1	X	X		X	
Cäsar-MG1	X	X			
Cäsar-VW1	X				
Dora-VW1	X	X	X		
Anton-Berta	X				
Cäsar-Berta	X				
Cäsar-Dora	X				
Dora-Emil	X				X

4.3 Schritt-3

Mit bewährten FBA-Tools erstelle aus dem formalen **Knotenkontext KV** den formalen **Begriffsverband** $\mathbb{B}$(**KV**): Jeder Liniendiagramm-Knoten ist ein formaler Begriff von **KV**.

Aus dem Liniendiagramm von $\mathbb{B}$(**KV**) ergeben sich im Schritt 4 – wegen der Regeln (1), (2) die **Klassen** und die **Klassen-taxonomie** ($\mathbb{C}$, $\leq_c$) des O-Schemas.

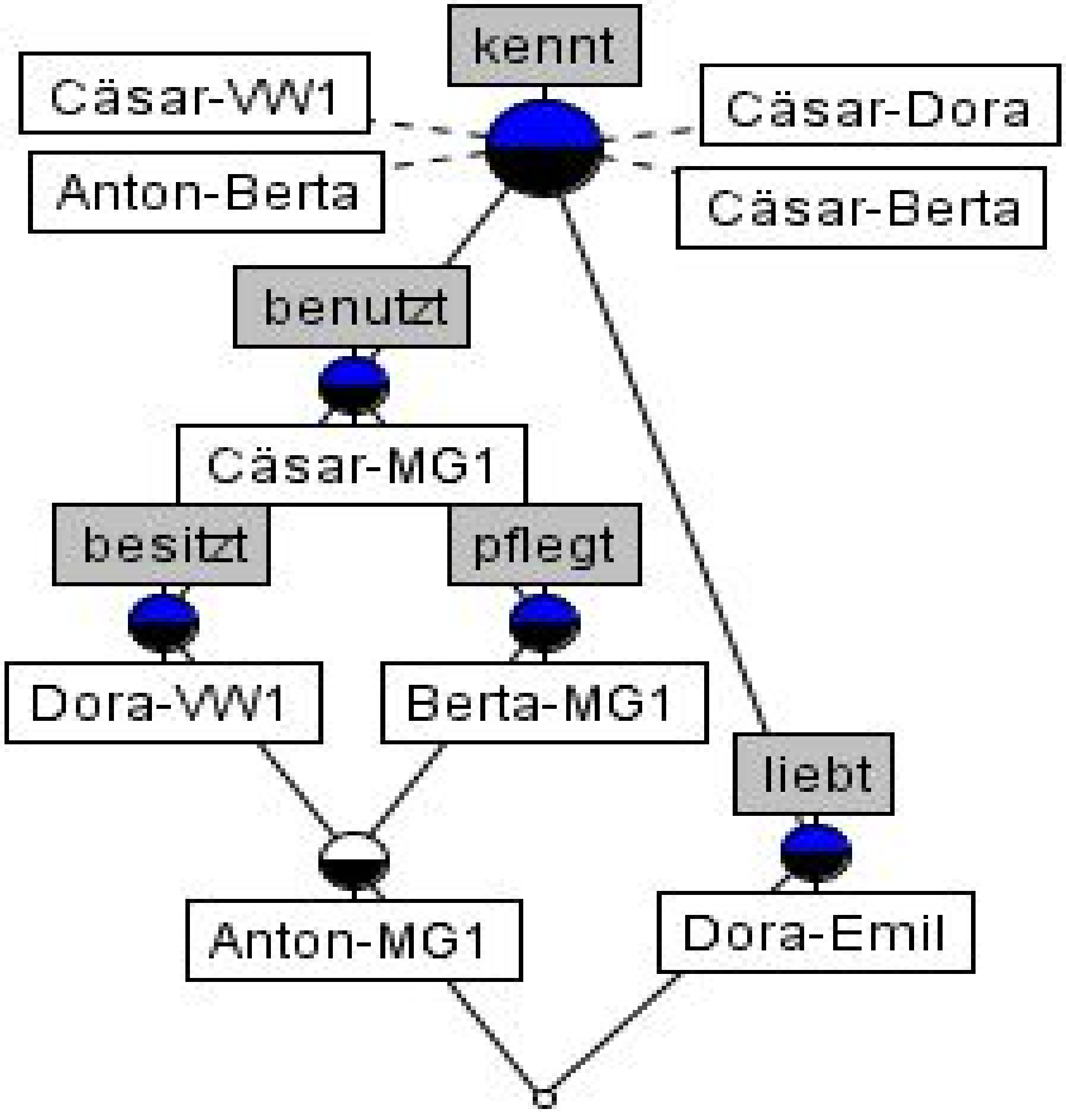
Liniendiagramm 3.1 ➔



Mit bewährten FBA-Tools erstelle aus dem formalen **Kantenkotext KE** den formalen Begriffsverband $\mathbb{B}$(**KE**): Jeder Liniendiagramm-Knoten ist ein formaler Begriff von **KE**.

Aus dem Liniendiagramm von $\mathbb{B}$(**KE**) ergeben sich im Schritt 4 – wegen der Regeln (3), (4), (5) – die Relationstypen, die Relationstypen-Taxonomie ($\mathbb{RT}$, $\leq_{RT}$) und die *richtige Aufhängung* der Relationstypen des O-Schemas.

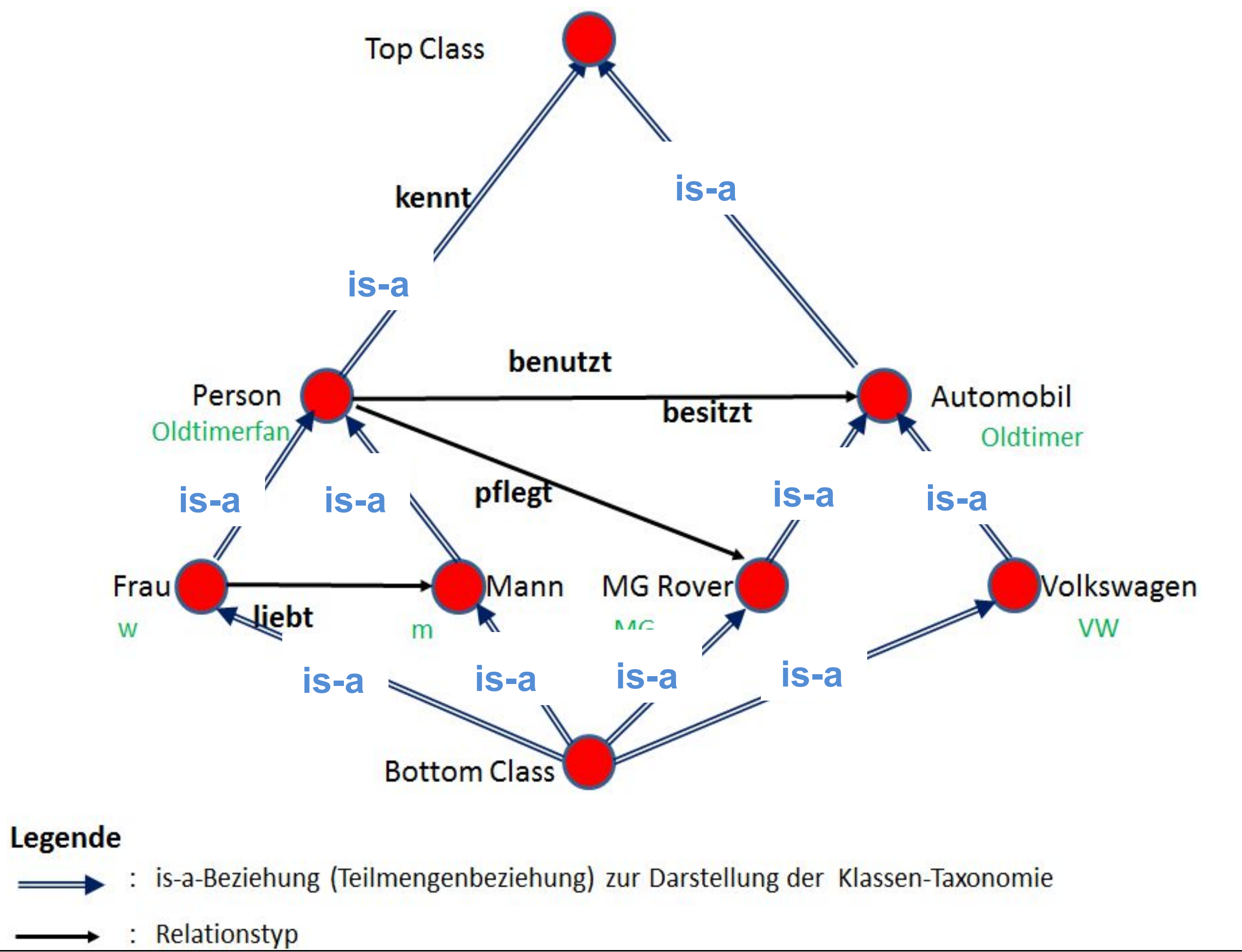
Liniendiagramm 3.2 ➔



4.4 Schritt 4

SN2:

- Im Schritt-4 muss man nur den Knoten des Liniendiagramms 3.1 neue ***Namen*** geben, um sie als **Klassen des O-Schemas** zu kennzeichnen. Beispiel: Die Instanzenmenge {>Berta, >Dora} bekommt den Klassennamen **^Frau** usw….
- Die (meist nichtssagende) Klasse, die **alle** Instanzen enthält, bekomme den Namen **^TopClass**, die (meist leere) Klasse, welcher *alle* **Attribute** zugesprochen werden, bekommt den Namen **^BottomClass.**

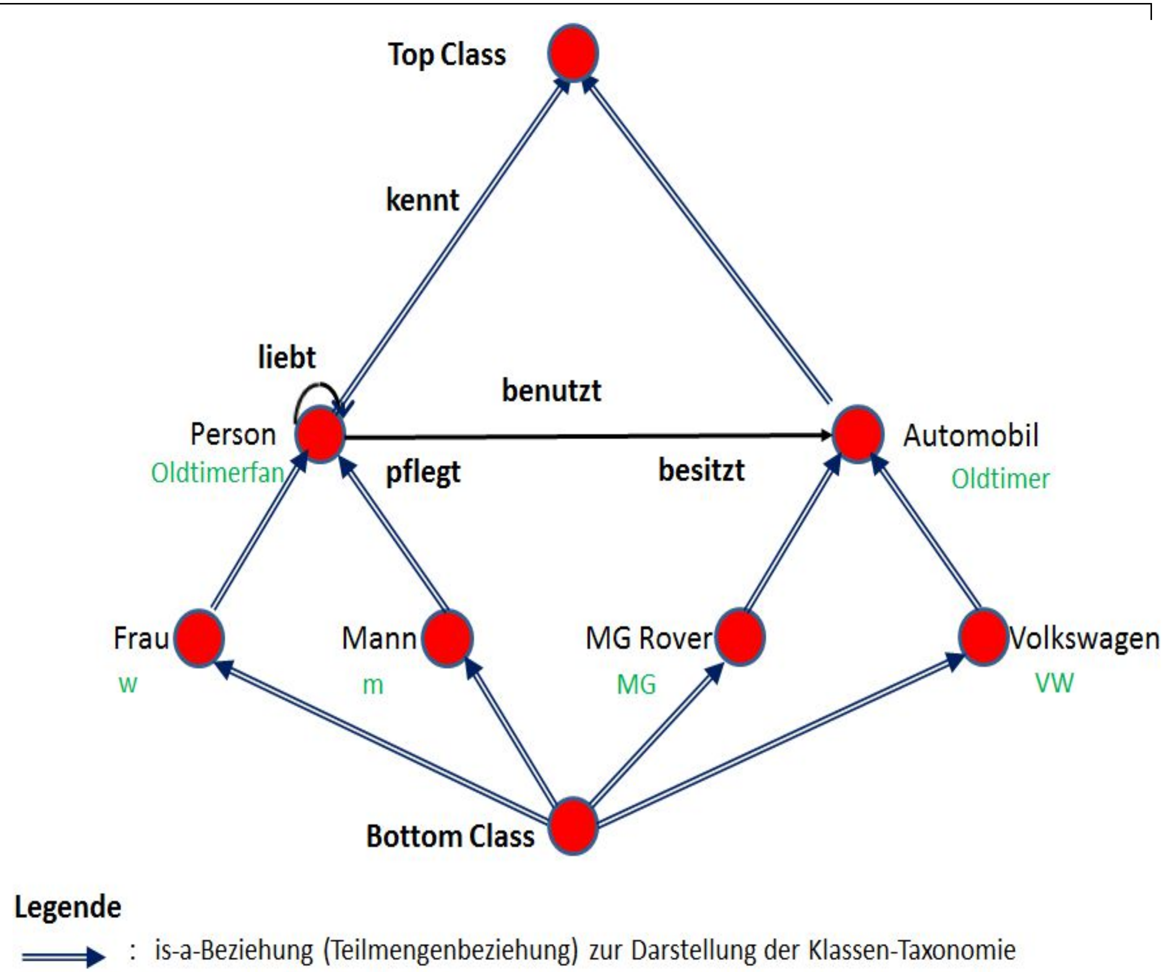


4.5Überprüfung des O-Schems und eventuelle Korrektur

Das bislang generierte O-Schema **SN2** gestattet (viel schneller als das **SN1**) eventuelle schematische Fehler zu entdecken.

Beispiele für mögliche Korrekturen:

- Das automatische Verfahren liefert bislang für den Relationstyp **<>liebt** die Aufhängung *prop*(<>liebt) =(^Frau, ^Mann) . Falls es sich herausstellt, dass im Oldtimer-Club auch ein Mann eine Frau liebt, dann müsstee für <>liebt die Aufhängung geändert werden zu *prop*(<>liebt) = (^Person, ^Person).
- Das automatische Verfahren liefert bislang für den Relationstyp **<>pflegt** die Aufhängung *prop*(<>pflegt)= (^Person, ^MG). Falls es sich herausstellt, dass auch Oldtimer eines anderen Typs gepflegt werden, müsste für <>pflegt die Aufhängung etwas allgemeiner gemacht werden zum Beispiel durch: „>Dora <>pflegt >VW1“).



5 Literatur

[0] ***Christoph Lübbert, Thomas Zeh:*** „Semantische Netze, Ontologien und Formale Begriffsanalyse“, Arbeitspapier, V5.3, Juni 2016. Basis des vorliegenden Vortrags V5.4-kurz.

[1] ***Bernhard Ganter, Rudolf Wille:*** „Formale Begriffsanalyse“ (FBA); Springer 1996

[2] ***Gregor Pickert:*** „Einführung in Ontologien“, Humbold-Universität Berlin, Feb. 2011 http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0203/SemWeb/artikel/2/Pickert_Ontologien_final.pdf

[3] ***Alexander Maedche, Valentin Zacharias:*** Clustering Ontology-based Metadata in the Semantic Web“. Universität Karlsruhe, Research Group WIM, 2002/2003 http://www.valentinzacharias.de/papers/clustering.pdf

[4] Wikipedia-Artikel: „Semantisches Netz“, https://de.wikipedia.org/wiki/Semantisches_Netz

[5] Wikipedia-Artikel: „Ontologie (Informatik)“, https://de.wikipedia.org/wiki/Ontologie_%28Informatik%29

[6] Wikipedia-Artikel: „Formale Begriffsanalyse“, https://de.wikipedia.org/wiki/Formale_Begriffsanalyse

[7] ***Christoph Lübbert:*** „Ontologiedefinition auf Basis von FBA“, V3.14, 27.08.2013 – vorgetragen im „Ontologie-Kreis“ Hochschule Darmstadt im Mai 2013.

[8] ***Christoph Lübbert:*** „Verfeinerung der Ontology-Sprache durch die FBA-Sprache“, V4f, Juni 2014 – vorgetragen im „Ontologie-Kreis“ Hochschule Darmstadt im Mai 2014.

[9] ***Hermann Bense:*** „Schematik.de“, Website gestartet im Jan 2014; siehe z.B. http://www.schematik.de/s1/Terminologie/index.html und „Ontology4us: Notations“ http://www.ontology4.us/english/Ontology4/Notations/index.html

[10] ***Gerd Stumme, Alexander Maedche***: „FCA-MERGE: Bottom-Up Merging of Ontologies”, University of Karlsruhe, 2001 http://www.aifb.kit.edu/images/9/94/2001_483_Stumme_FCA-Merge_Bott_1.pdf

[11] ***Rudolf Wille:*** Conceptual Graphs and Formal Concept Analysis.
In: D. Lukose, H. Delugach, M. Keeler, L. Searle, J.F. Sowa (eds.):
Conceptual Structures:
Fulfilling Peirce's Dream. LNAI 1257. Springer, Heidelberg 1997, 290–303.

[12] Beschreibungen und Downloadmöglichkeiten von Software zur FBA
http://www.ernst-schroeder-zentrum.de/software/

[13] ***Hesse, Wolfgang:*** FB Mathematik und Informatik, Philipps-Universität Marburg
Ontologie(n). In Informatiklexikon der GI
https://www.gi.de/service/informatiklexikon/detailansicht/article/ontologien.html

[14] ***Friedrich Steimann:*** „Formale Modellierung mit Rollen", Habiltationsschrift, Universität Hannover, Frühjahr 2000
https://www.fernuni-hagen.de/ps/pubs/HabilitationsschriftSteimann.pdf