Nicola Guarino und Christopher Welty: «a formal ontology of properties»

Vortrag von Tom Pfüller und Matthias Richter

am 8. Mai 2002

im Seminar ≪Ontologiebasierte Wissensmodellierung≫ (B. Heller, H. Herre)

`http://aspra9.informatik.uni-leipzig.de/~mrichter/guarino.pdf.gz`

Worum geht es hier?

Problem:

- taxonomische Struktur bestehender Ontologien „poor and confusing“
- enthaltene Inkonsistenzen machen Ableitungen daraus / Bilden von Relationen damit schwierig

Lösungsansatz:

- Meta-Properties als Grundbausteine definieren
- damit Abbildung von *Identität*, *Einheit*, *Essenz* und *Abhängigkeit*

Ziel: Verbesserung der

- Konstruktion und Verständlichkeit
- Vergleichbarkeit und Integrierbarkeit

von Taxonomien

Ontologien – Probleme

Bemerkung: Bedeutungsunterschied

- *Ontologie* als philosophische Disziplin
- Eine *Ontologie* als Konstruktion, die an Wörter gebundene spezifische Bedeutungen ausdrückt

Probleme

- unklarer Gebrauch von *Relationen*
- undefiniertes Verhalten bei *Subsumption* (Unterordnung)
- Ontologien zu verstehen und zu integrieren schwierig

Lösungsansatz (I): Identität

- Wie kann man verschiedene Instanzen einer Klasse vermittels eindeutiger Merkmale unterscheiden?

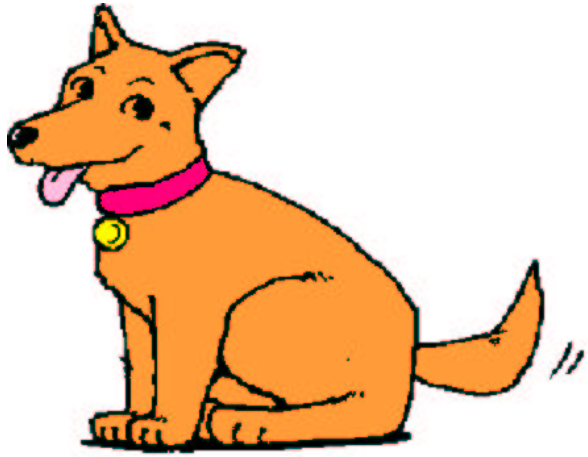


Abbildung 1: Identität: Ist dieser Hund *mein* Hund?

- Gesucht ist dazu eine *characteristic property*
- Sinneseindrücke, soziokulturelle Faktoren, . . . haben Einfluß auf das Identitätskonzept.

Lösungsansatz (II): Einheit

- Wie kann man Teile einer Entität von deren Umwelt abgrenzen?

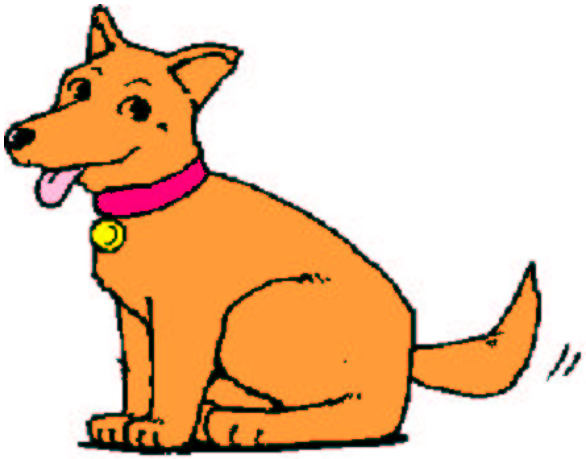


Abbildung 2: Einheit: Gehört das Halsband zum Hund?

- Gesucht ist dazu eine *unifiying relation*

Lösungsansatz (III): Essenz

- Welche Eigenschaften dürfen sich bei einer Entität mit der Zeit ändern?
- Welche Eigenschaften müssen bei einer Entität gleich bleiben?
- Grundlage der Definition von *rigid properties*
- Wie kann man eine Entität (wieder-)erkennen?
- Problem der *synchronen* und *diachronen* Identität

Lösungsansatz (IV): Abhängigkeit

- Manche Eigenschaften von Entitäten sind nur im Vergleich zu den Eigenschaften anderer Entitäten möglich
- Beispiel: a ist größer als b, d ist Kind von c, . . .

Lösungsansatz (V): Beispiele

Beschreibungslogik:

- Hinreichende und notwendige Kriterien für *Mitgliedschaft in einer Klasse*
- Keine Identität, sondern nur das Zutreffen spezifischer Eigenschaften
- Frage: „ist das ein Hund?“

Objektorientierte Modellierung:

- Globally Unique IDs
- Keine Identität, da mehrere Beschreibungen einer Entität möglich sind (mehrere unverbundene Datensätze zu einer Entität)
- Frage: „Sind A und B Beschreibungen desselben Hundes?“

Lösungsansatz (V): Beispiele (II)

Primärschlüssel:

- basieren meist auf *extrinsischen Eigenschaften*:
  - Eigenschaften, die nicht einem Individuum inhärent sind, sind relationaler Natur
  - Beispiel: Sozialversicherungsnummer zum *Erzwingen* der Eindeutigkeit in einem Informationssystem
  - gut zur Beschreibung geeignet
- *Identität* basiert auf *intrinsischen Eigenschaften*
  - Eigenschaften, die einem Individuum inhärieren und unabhängig von anderen Individuen sind.
  - Beispiel: Das Gehirn eines Menschen
  - Können oft nicht in einem System implementiert werden

Meta-Eigenschaften (I): Rigidity

Definition: Eine Eigenschaft ist

- eine *rigid property (+R)*: $\forall x\ \Phi(x) \rightarrow \Box\Phi(x)$
- eine *non-rigid property (−R)*: $\exists x\ \Phi(x) \wedge \neg\Box\Phi(x)$
- eine *anti-rigid property (∼R)*:$\forall x\ \Phi(x) \rightarrow \neg\Box\Phi(x)$
- eine *semi-rigid property (¬R)*, wenn sie *non-rigid* aber nicht *anti-rigid* ist.

NB: *rigidity* wird nicht an Untereigenschaften vererbt.
NB: *anti-Rigidity* als zwingend nicht-essentielle Eigenschaft

Meta-Eigenschaften (I): Rigidity (II)

Beispiel: Jemand kann Student werden und wieder aufhören, Student zu sein. Sie kann aber nicht aufhören, Person zu sein. Student ist also *non-rigid*, wohingegen Person *rigid* ist. Student ist sogar *anti-rigid*, weil es keinen Studenten gibt, für den sein Student-sein (als Entität) essentiell ist.

Eine *semi-rigid* Eigenschaft entsteht zum Beispiel, wenn man eine Mischeigenschaft bildet, etwa „Mensch-oder-Freund“.

Meta-Eigenschaften: Identität

Identität normalerweise: $\Phi(x) \wedge \Phi(y) \rightarrow (\rho(x,y) \leftrightarrow x=y)$

Probleme:

- Einschränkungen auf *extrinsische* Eigenschaften bleiben gültig
- Kein Unterschied zwischen Identität *tragen* und *spenden*
- Keine Berücksichtigung des Zeitfaktors
- bisweilen Finden einer *notwendigen* und *hinreichenden* Identitätsbedingung schwierig

Lösung:

- Beschränkung der IC auf *rigid properties*
- Explizite Berücksichtigung der Zeit in ρ mittels E(*var*,*zeitpunkt*)
- Ein IC soll entweder *notwendig* oder *hinreichend* sein

Identitätsbedingungen

Definition: Eine *rigid property* Φ trägt die **notwendige Identitätsbedingung** $\Gamma(x,y,t,t')$, wenn Γ nur x,y,t,t' als freie Variablen enthält und

1. $\neg\forall xytt'(\Gamma(x,y,t,t') \leftrightarrow x=y)$
2. $E(x,t) \wedge \Phi(x,t) \wedge E(y,t') \wedge \Phi(y,t') \wedge x=y \rightarrow \Gamma(x,y,t,t')$
3. $\neg\forall xy(E(x,t) \wedge \Phi(x,t) \wedge E(y,t) \wedge \Phi(y,t') \rightarrow \Gamma(x,y,t,t'))$

Definition: Eine *rigid property* Φ trägt die **hinreichende Identitätsbedingung** $\Gamma(x,y,t,t')$, wenn Γ nur x,y,t,t' als freie Variablen enthält und

1. $\neg\forall xytt'(\Gamma(x,y,t,t') \leftrightarrow x=y)$
2. $E(x,t) \wedge \Phi(x,t) \wedge E(y,t') \wedge \Phi(y,t') \wedge \Gamma(x,y,t,t') \rightarrow x=y$
3. $\exists xytt'\ \Gamma(x,y,t,t')$

Identität tragen und spenden

Definitionen:

- Eine *non-rigid property* **trägt** eine Identitätsbedingung Γ, wenn sie von einer *rigid property* untergeordnet wird, die Γ trägt.
  Bezeichnung: $+$I (sonst: $-$I)
- Eine Eigenschaft Φ **spendet** eine Identitätsbedingung Γ, wenn sie *rigid* ist und Γ trägt und keine der Φ unterordnenden Eigenschaften Γ tragen.
  Bezeichnung: $+$O (sonst $-$O)
  NB: $+$O impliziert $+$I und $+$R
- Eine Eigenschaft, auf die $+$I zutrifft, heißt *sortal*.

Meta-Eigenschaften (V): Abhängigkeit

- Eine Eigenschaft Φ ist *äußerlich abhängig* von einer Eigenschaft Ψ, wenn für alle ihre Instanzen x notwendig ein Ψ existiert, das weder Teil noch Konstituente von x ist: $\forall x \Box(\Phi(x) \rightarrow \exists y \Psi(y) \wedge \neg P(y,x) \wedge \neg C(y,x))$
- Eine *äußerlich abhängige* Eigenschaft wird mit $+D$ bezeichnet (sonst $-D$)
- Beispiel: ELTER ist äußerlich abhängig von KIND weil die Elternrolle an die Existenz des Kindes gebunden ist
- Beispiel: PERSON ist nicht äußerlich abhängig von KÖRPER oder HERZ, weil jede Person ein Herz als Körperteil hat und der Körper die Person konstituiert.

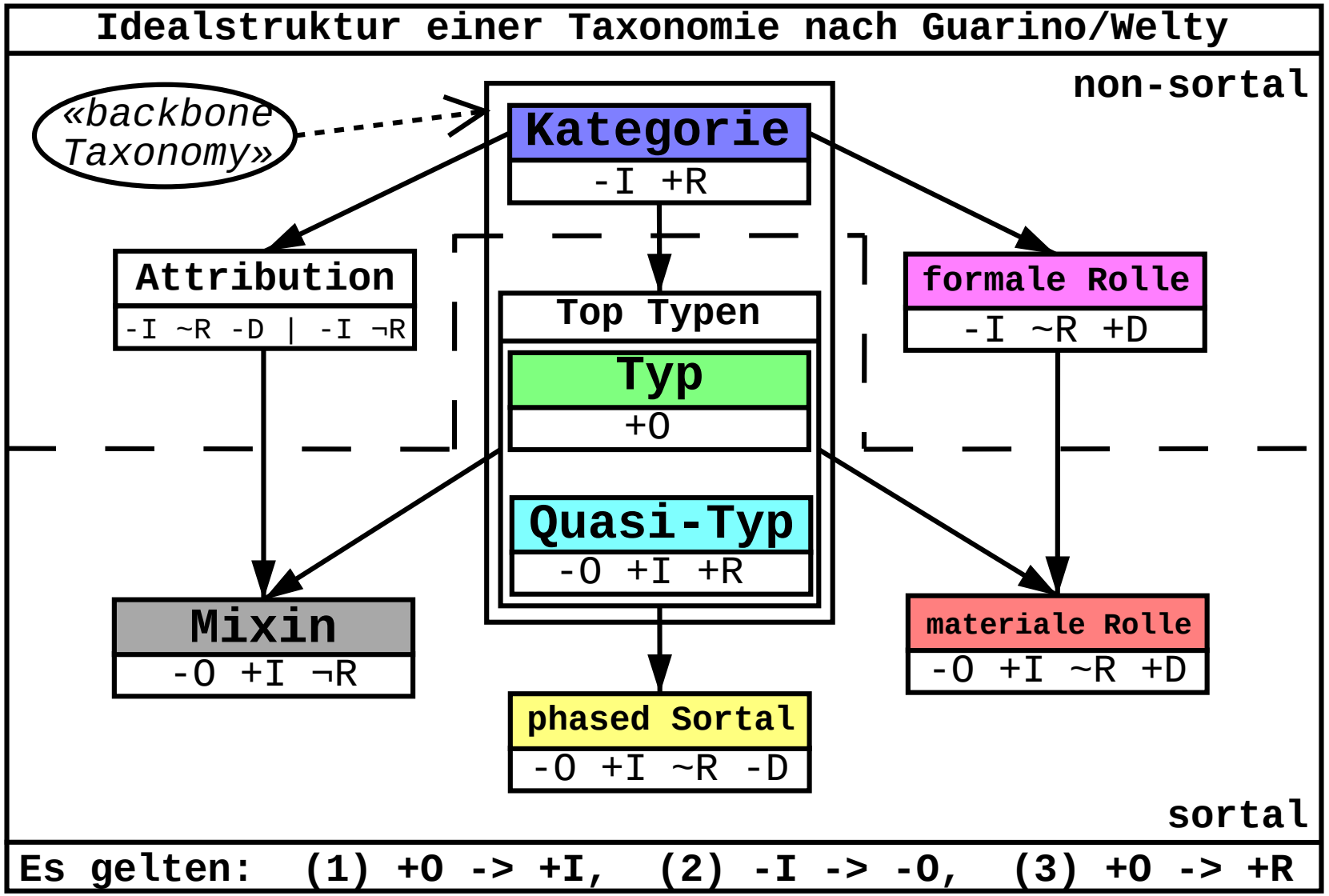
Einschränkungen

- Eine *anti-rigid* Eigenschaft kann keine *rigid* Eigenschaft unterordnen
- Eine Eigenschaft, die Identität trägt, kann keine Eigenschaft, die keine Identität trägt, unterordnen
- Eigenschaften mit *inkompatiblen Identitätsbedingungen* sind unverbunden
- Eine *äußerlich abhängige* Eigenschaft kann keine äußerlich nicht abhängige Eigenschaft unterordnen
- *Sortal Individuation:* Jedes Domänenelement muß Instanz einer Eigenschaft sein, die Identität trägt (Quine: „no entity without identity“)
- *Sortal Expandability:* Wenn zwei Entitäten gleich sind, so müssen sie Instanzen einer *rigid* Eigenschaft sein

Eigenschaften: Klassifikation

- Basiseigenschaften setzen sich aus den Metaeigenschaften zusammen
- Basiseigenschaften werden dadurch *klassifiziert*
- Kombinatorisch: 24 Möglichkeiten, davon aber 10 inkohärent
- Aus den verbleibenden 14 Kombinationen lassen sich 8 Klassen von Basiseigenschaften bilden.

Übersicht


Idealstruktur einer Taxonomie nach Guarino/Welty
non-sortal
«backbone Taxonomy»
Kategorie
-I +R
Attribution
-I ~R -D | -I ¬R
Top Typen
Typ
+O
Quasi-Typ
-O +I +R
formale Rolle
-I ~R +D
Mixin
-O +I ¬R
materiale Rolle
-O +I ~R +D
phased Sortal
-O +I ~R -D
sortal
Es gelten: (1) +O -> +I, (2) -I -> -O, (3) +O -> +R

Eigenschaften (I): Kategorie

- Eigenschaften: +R -I
- zum Teilen der Domäne in handhabbare Segmente
- kann unterordnen: alle Arten von Eigenschaften
- kann untergeordnet werden zu: Kategorien, Attributionen
- Bildet normalerweise Bäume
- Zumindest die obersten Kategorien sollten unverbunden sein
- Beispiel: ENTITY als Oberkategorie, die CONCRETE ENTITY und ABSTRACT ENTITY unterordnet

Eigenschaften (II): Typ

- Eigenschaften: +R +O
- sind die wichtigsten Elemente einer Ontologie, weil nur die Typen Identität spenden können
- jedes Domänenelement muss mindestens einen Typen instantiieren
- sollten die Haupteigenschaften repräsentieren und daher in der mittleren Ebene der Ontologie vorkommen
- kann unterordnen: alle *sortal* Eigenschaften
- kann untergeordnet werden zu: Kategorien, (Quasi-)Typen und Attributionen
- sollte unter mindestens einer Kategorie stehen
- Beispiel: PERSON, KATZE und WASSER

Eigenschaften (III): Quasi-Typ

- Eigenschaften: +R -O +I
- sie gruppieren Entities basierend auf nützlichen Kombinationen von Eigenschaften, die nicht die Identität beeinträchtigen
- Stellen bereit: notwendige und hinreichende Bedingungen für die Mitgliedschaft in einer Klasse
- kann unterordnen: alle *sortal* Eigenschaften
- kann untergeordnet werden unter: Kategorien, Typen, Attributionen und Mixins
- müssen von mindestens einem Typ untergeordnet werden
- sollten nicht untergeordnet werden von: Mixins und zu vielen Attributionen
- Beispiel: WIRBELLOSE TIERE, PFLANZENFRESSER

backbone properties

- *rigide Eigenschaften* (Kategorien und (Quasi-)Typen) bilden das *Rückgrat* einer Ontologie.
- unveränderliche Eigenschaften
- bilden eine Untermenge aller Eigenschaften in der Ontologie
- sind wichtig für das Verständnis einer Ontologie, weil alle relevante strukturale Information in ihnen steckt

Eigenschaften (IV): formale Rolle

- Eigenschaften: -I ∼R +D
- Ausdruck für die Rolle, die eine Entität im Rahmen eines Ereignisses spielt
- abstrakte Relationen zwischen zwei oder mehr Entitäten
- Es gibt eine Rollenhierarchie, die allgemeinsten Rollen stehen im oberen Teil der Ontologie
- kann unterordnen: *non-rigid* Eigenschaften, die *abhängig* sind: abhängige Attribution, Mixins, materiale Rollen
- kann untergeordnet werden unter: andere formale Rollen, Attributionen oder Kategorien
- sollten nur dazu verwendet werden, die Rollenhierarchie zu *ordnen*, d.h. sollten keine Mixins und Attributionen unterordnen
- Beispiel: OBJEKT (EINER HANDLUNG), INSTRUMENT

Eigenschaften (V): materiale Rolle

- Eigenschaften: -O +I ∼R +D
- Rollen, die auf *bestimmte Arten* von Entitäten beschränkt sind
- Ursprung in einem *konkreten* Ereignis
- kann unterordnen: materiale Rollen, abhängige Mixins
- kann untergeordnet werden unter: alle Eigenschaften
- muss unter einem Typ untergeordnet werden
- sollte keine Mixins unterordnen
- sollte nur unter Rollen oder *backbone properties* untergeordnet werden
- Beispiel: STUDENT (eingeschrieben), VERHEIRATET (Hochzeit), NAHRUNG

Eigenschaften (VI): phased sortals

- Eigenschaften: -O +I ∼R -D
- stellen *lokale* ICs bereit, sind für einen bestimmten Zeitraum gültig („Zustände einer Entität“)
- kann unterordnen: alle *non-rigid* Eigenschaften
- kann untergeordnet werden unter: alle unabhängigen Eigenschaften
- muss unter einem Typ untergeordnet werden
- sollen nur unter *backbone properties* untergeordnet werden
- sollen andere *phased sortals* und *materiale Rollen* unterordnen

Eigenschaften (VI): phased sortals

- sollen gemeinsam mit den übrigen *phased sortals* einer Entität unter einen eigenen, gemeinsamen Typen oder Quasi-Typen untergeordnet werden
- Beispiel: RAUPE und SCHMETTERLING, FARBEN EINES APFELS, eigentlich nur in biologischem Zusammenhang

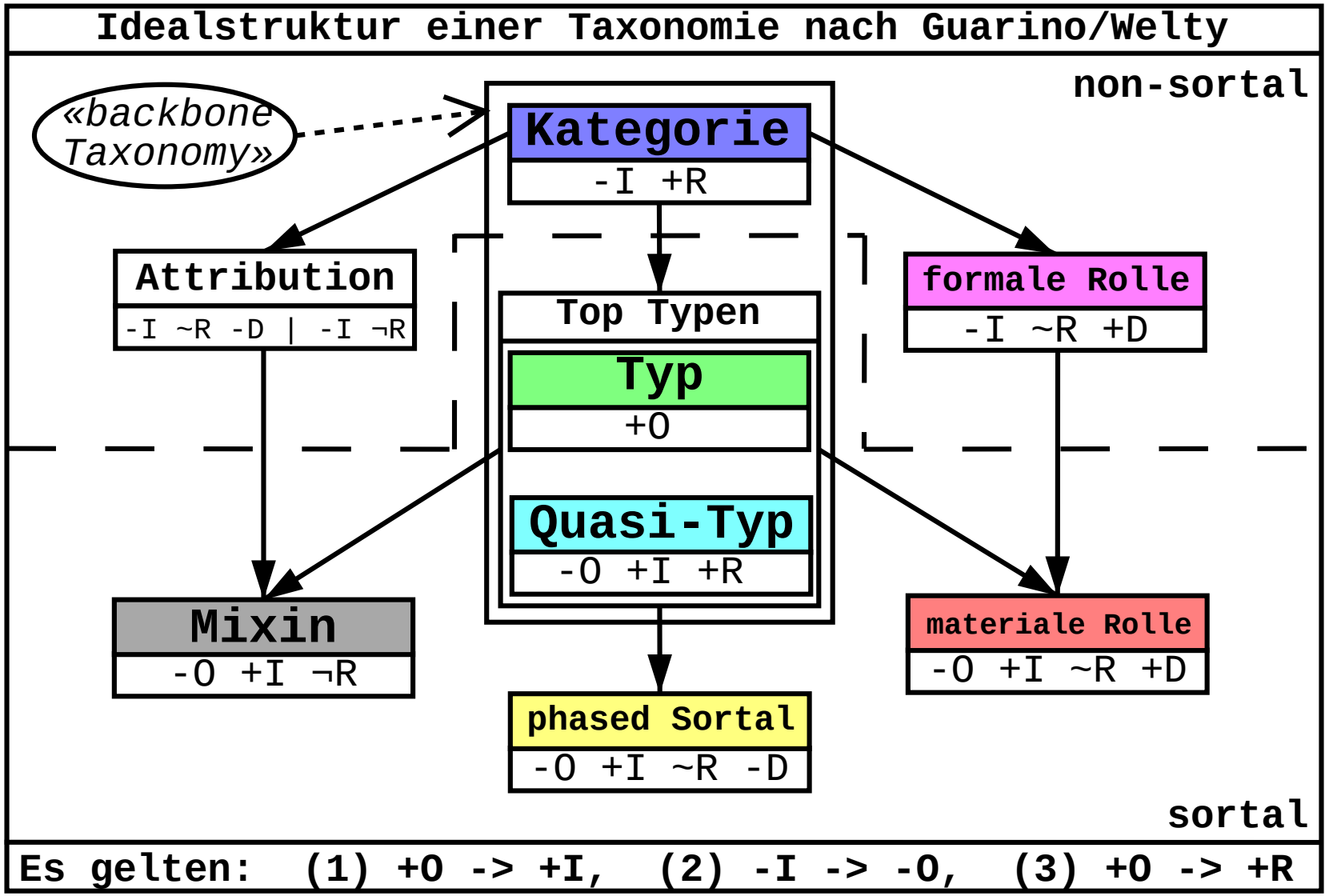
Eigenschaften (VII): attribution

- Eigenschaften: -I $\sim$R -D oder -I $\neg$R
- Werte von Attributen oder Qualitäten, wie Farbe oder Form
- sollten wohl $\sim$R sein (offene Frage)
- kann unterordnen: alle Eigenschaften
- kann untergeordnet werden unter: *non-sortal* Eigenschaften
- sollten nur Mixins oder andere Attributionen unterordnen
- sollten nur von Kategorien untergeordnet werden
- Beispiel: ROT, DREIECKIG, MÄNNLICH

Eigenschaften (VIII): mixins

- Eigenschaften: -O +I ¬R
- Disjunktionen und Konjunktionen aus *rigid* und *non-rigid* Eigenschaften
- kann unterordnen: alle *sortal* Eigenschaften
- kann untergeordnet werden unter: alle Eigenschaften
- müssen von einem *sortal* untergeordnet werden
- sollten keine *rigid* Eigenschaften unterordnen
- sollten soweit wie irgend möglich vermieden werden, weil sie so unspezifisch sind
- können zur Organisation großer Ontologien dienen
- Beispiel: KATZE-ODER-WAFFE

Idealfall


Idealstruktur einer Taxonomie nach Guarino/Welty
non-sortal
«backbone Taxonomy»
Kategorie
-I +R
Attribution
-I ~R -D | -I ¬R
Top Typen
Typ
+O
Quasi-Typ
-O +I +R
formale Rolle
-I ~R +D
Mixin
-O +I ¬R
phased Sortal
-O +I ~R -D
materiale Rolle
-O +I ~R +D
sortal
Es gelten: (1) +O -> +I, (2) -I -> -O, (3) +O -> +R

Fazit

- Methodik zur Strukturierung von Ontologien mit Hilfe von Basis-Eigenschaften, die aus Meta-Eigenschaften konstruiert werden
- klarere Taxonomie (aufgrund der semantischen Beschränkungen)
- Identifizierung von *backbone properties*
- zwingen den Ersteller der Ontologie dazu, seine Klassifikation zu explizieren.
- können zu besser wiederverwendbaren Ontologien führen