

Knowledge Representation in Cyc

**Quelle: R. V. Guha, D. B. Lenat, Cyc: A Midterm Report.
und andere**

**Seminar „Wissensrepresentation und Bereichsontologien“
Saarbrücken, 15.06.2004
Sebastian Blohm**

Idee | Architektur | Upper Level | Zerteilung | Ende

Outline

- **Grundidee und Geschichte**
- **Architektur**
- **Aufteilung der Top- Level**
- **Zerteilung**
- **Zusammenfassung**

Idee | Architektur | Upper Level | Zerteilung | Ende

Pinguin und Co.

Probleme, die Angesprochen werden

- **Nicht-Monotonie (Pinguin)**
 - **Defaults und Microtheories**
- **Veränderung (Friseurbesuch)**
 - **SubAbstractions**
- **Identität und Teilung (Apfel und Apfelmus,)**
 - **Substance vs. Object**
 - **Granule**
 - **Structure**

Idee | Architektur | Upper Level | Zerteilung | Ende

Grundidee: Common Sense Knowledge

Wissen als Grundlage von Intelligenz:

“Every normal child learns millions of different fragments of knowledge and skill that no computer yet comprehends, but which people regard as ‘obvious’.” – M. Minsky

Algorithmen ersetzen nicht die Wissensdatenbank:

“We believe that there is no shortcut to being intelligent.” – D. Lenat

Skalierbarkeit ist eines der größten Probleme von AI Ansätzen:

Cyc = Ontologisches Wissen + passende Inferenzmechanismen

Idee | Architektur | Upper Level | Zerteilung | Ende

Geschichte von Cyc

Die Bemühungen von Cyc-Gründer Douglas B. Lenat, Commonsense-Knowledge zu kodieren begannen vor 1984 am Atari Research Center.

Von 1984 bis 1994 war Cyc Projekt der Firma MCC in Austin Texas.

Seitdem besteht die Firma Cycorp, in der Lenat selbständig Cyc weiterentwickelt.

Cycorp hat heute nach eigenen Angaben ca. 60 Mitarbeiter und versucht seit 2000, Cyc als Produkt zu vermarkten. Seit 2001 gibt es OpenCyc, so dass jeder spezifisches Wissen hinzufügen kann.

Fördervolumen (nach Presseangaben) 60 Millionen \$US, davon 25 vom US-Militär.

Idee | Architektur | Upper Level | Zerteilung | Ende

Linguistische Anwendungen

Ontologien können im Zusammenhang mit Sprachverarbeitung verwendet werden:

- **Disambiguierung**
- **Anaphernresolution**
- **Heuristische Bewertung von Aussagen**

Aber:

- **Kaum Anwendungen veröffentlicht.**
- **Cyc ist keine „lexical Database“**

Cyc-Aufbau

- **Cyc hat seine eigene Sprache CycL zum notieren von Wissen (typisierter FOPC)**
- **Ein „Cyc Knowledge Server“ verfügt über die Wissensdatenbank und Inferenzmechanismen und stellt diese Anwendungen zur Verfügung.**
- **Anwendungen können die Wissensdatenbank ändern und Anfragen stellen.**
- **Für Ein- /Ausgabe ohne CycL steht (angeblich) ein „Semantic Integration Bus“ und ein Natural Language Subsystem zur Verfügung.**

Formalismus – Instanz und Collection

Cyc kann in Frames dargestellt werden.

Die wichtigste Ordnung in der Ontologie ist die Spezialisierungshierarchie:

- **Alle möglichen Entitäten: Instanzen mit Eigenschaften.**
- **Collections: Mengen von Instanzen.**
- **Alle Ontologischen Beziehungen sind Relationen zwischen Collections.**
 - Gens: **Obermengen** \subseteq
 - Specs: **Teilmengen** \supseteq
- **Relation zwischen Instanz und Menge:**
 - **IsA: Element** \in

| | |
|---------------------------------------|-----------------|
| Oskar | Instance |
| IsA: Saarländer | |
| size: around170cm liveIn: Saarland | |

| | |
|---|-------------------|
| Saarländer | Collection |
| IsA: PersonType Gens: SpatialThing Specs: ... | |
| AllInstances.liveIn: ... | |

Formalismus – Types, Collections of Collections

Ein Typ benennt gleichartige Instanzen.

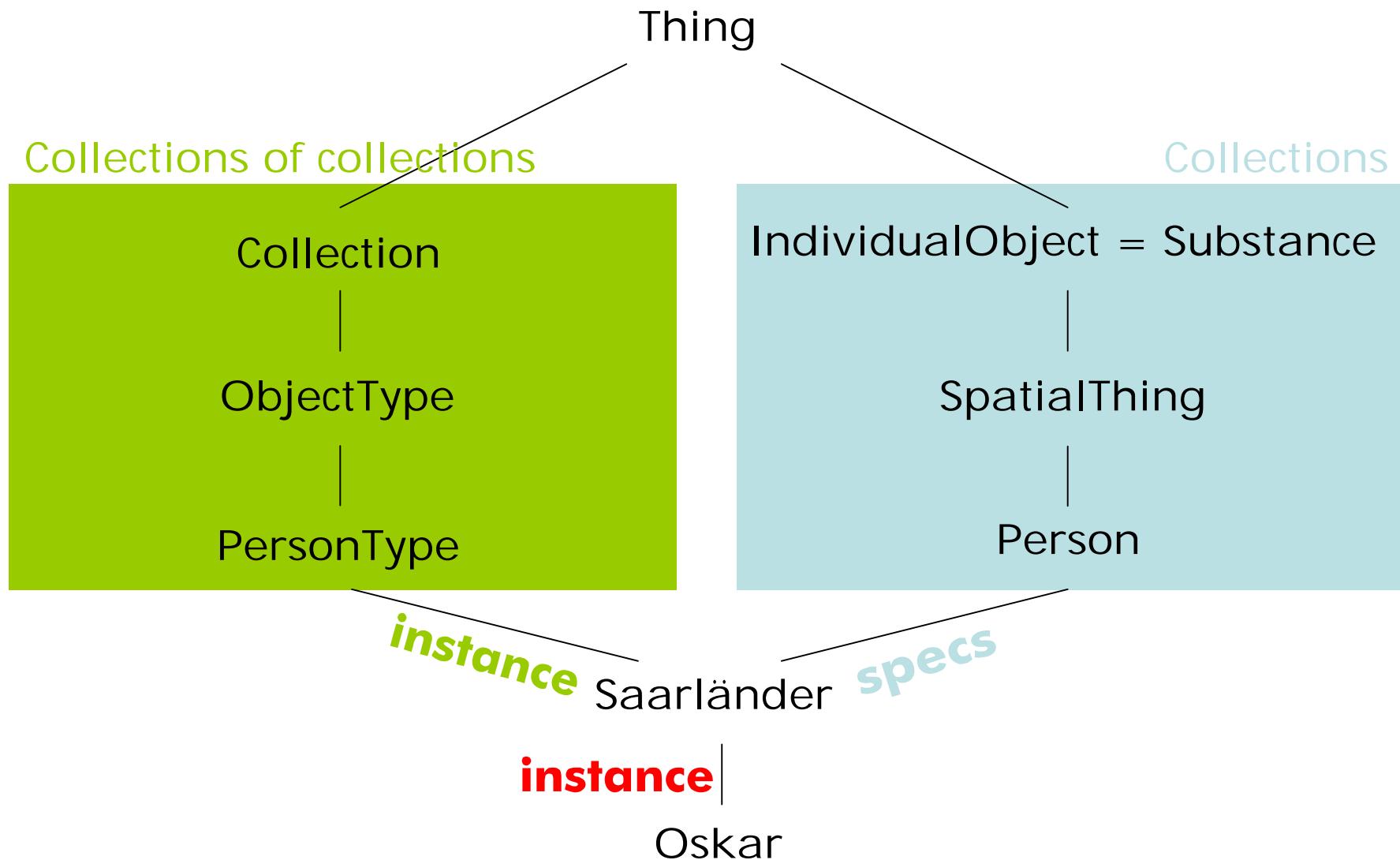
Typen von Collections können beschrieben werden indem man sie in Collections von Collections einordnet:

PersonType ist die Menge aller Mengen, die nur aus Personen bestehen.

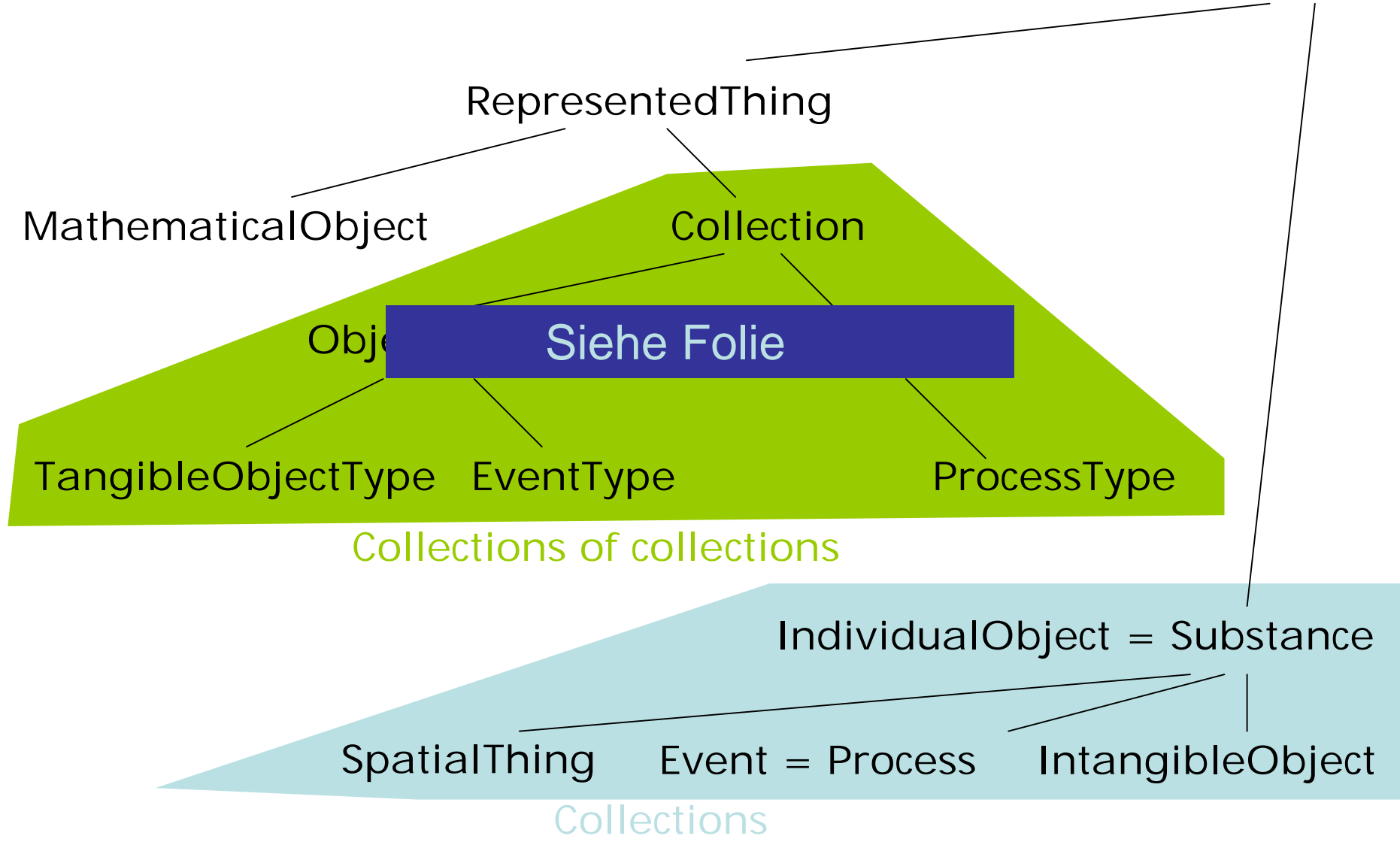
| | |
|-------------------------------|------------|
| Saarländer | Collection |
| IsA: PersonType Specs: ... | |
| AllInstances.liveIn: ... | |

| | |
|--|-------------------|
| PersonType | Collection of Cs. |
| Gens: TangibleObjectType Specs: Saarländer, VielZuVieleFolienBenutzer | |
| AllInstances.allInstances: human... | |

Baumdarstellung



Thing



Übersetzung in Prädikate

isA (Oskar, Saarländer)

size (Oskar, around170cm)

likes(Oskar, IceCream)

| | |
|---------------------------------------|----------|
| Oskar | Instance |
| IsA: Saarländer | |
| size: around170cm likes: Ice-cream | |

Dabei sind alle Prädikate streng getypt:
likes \in **IndividualObject** \rightarrow **Thing**
size \in **SpatialThing** \rightarrow **SizeInterval**

Inferencing

Mit Hilfe von Regeln (die auch zum Wissen von Cyc gehören) können nun weitere Schlussfolgerungen gezogen werden.

Beispielregel:

$\forall X:$

**weight(X, lessThan80Kilograms) \wedge
size(X, moreThan190cm)**

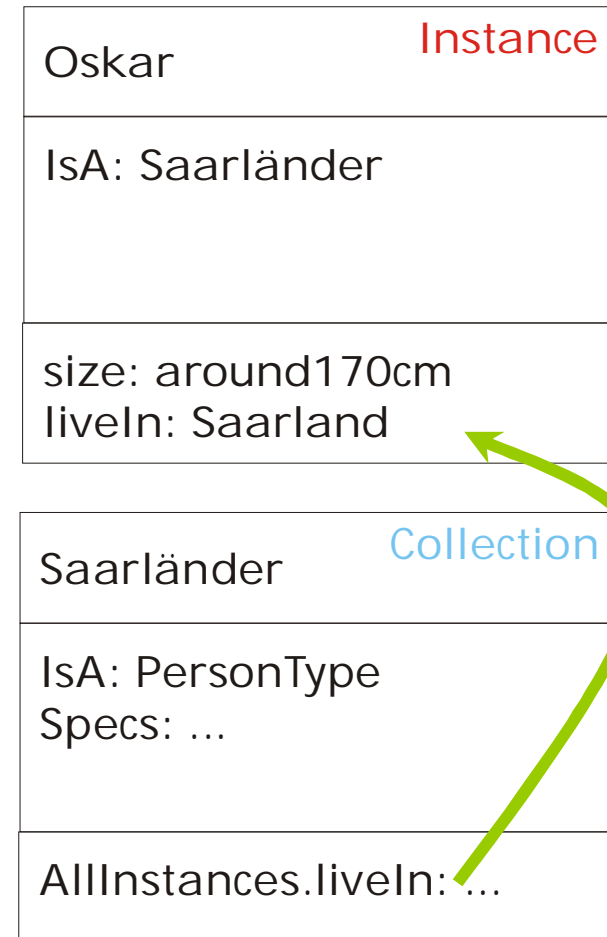
\Rightarrow

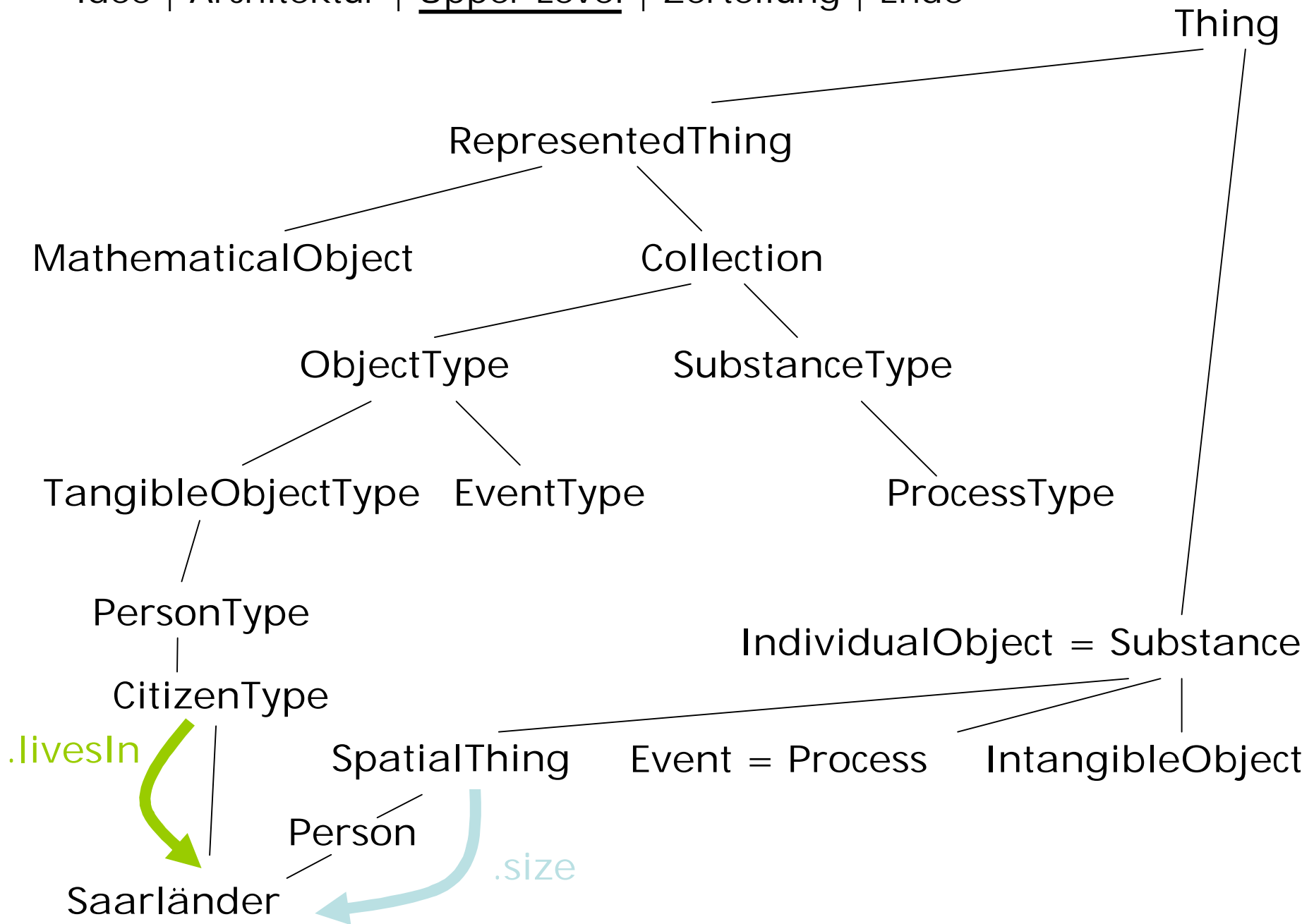
slim(X, true)

**Große Ontologie \Rightarrow viele Prädikate \Rightarrow Bedarf an
Effizienten Inferenzmechanismen. (siehe Mark)**

Vererbung

Specs erweitern die Eigenschaften ihrer Genls.
Specs erben die Eigenschaften ihrer Genls.
Jede Collection kann mehrere Genls haben, erbt also von verschiedenen Collections: Mehrfachvererbung.





Zerteilung als wichtiges Kriterium

Neben Spezialisierung spielt die Teil- Ganzes- Beziehung eine wichtige Rolle im Design der Cyc Ontologie. Die Top- Level ist so gestaltet, dass sich gewisse Aufteilungen gut modellieren lassen:

- **Klasse von Bestandteilen:**
 - Substance
 - Object
- **Bestandteile:**
 - Structure (Objects)
 - Granule (Substance)
- **Zeitliche Aufteilung:**
 - Structure (Events)
 - SubAbstractions (Objekte zu unterschiedlichen Zeitpunkten)

Substance Type vs. Object Type

Zerteile ich ein Stück Holz, **erhalte ich mehrere Stücke Holz.**

Zerteile ich einen Tisch, **erhalte ich nicht mehrere Tische.**

Daran lassen sich unterscheiden:

- **Substanz: Teile ihrer Instanzen sind vom gleichen Typ (Holz, Wasser, Spaziergang)**
- **Objekt: Wird beim Zerteilen in Bestandteile zerlegt. (Tisch, Getränk, Theateraufführung)**

Das gleiche Kriterium kann zur Unterscheidung von Prozess (wie Spaziergang) oder Ereignis (Theateraufführung) verwendet werden.

Eigenschaften heißen intrinsisch, wenn sie Teilung „überleben“ (üblich für Eigenschaften von SubstanceType) oder extrinsisch, wenn nicht (üblich für Eigenschaften von ObjectType)

Der Structure Slot

- **Im Structure Slot können** Bestandteile von Objekten **angegeben werden.**
- **Der Structure Slot kann je** der Kategorie **entsprechend** gefüllt werden.
- **Auch Events haben eine** Struktur. Sie legt die **Reihenfolge von Sub-Events** und **beteiligte Actors fest.**

| | |
|----------------------------|----------|
| MyTable | Instance |
| IsA: Table | |
| structure: TableStructure1 | |

| | |
|---|----------|
| TableStructure1 | Instance |
| IsA: TableStructure | |
| legs: myTablesLeg1, ... top: TableTop1 | |

Granule

Analog zu Structure bei Objekten gibt es Granules von Substances.

- **Granules sind Bestandteile von Substanzen. Jede Substanz setzt sich danach aus einzelnen (gleichartigen) Objekten zusammen.**
- **Beispiele:** Menschenmasse – Mensch, Mensch – Zellen, Zellen – Zellmaterial, Zellmaterial – Kohlenwasserstoffmoleküle.
- **Die Abwechslung Objekt- Substanz in diesem Beispiel ist typisch.**

SubAbstractions

Objekte haben die Tendenz sich zu verändern, d.h. Eigenschaften zu wechseln. Es gibt deshalb Objekte wie SebastianGestern. Also zeitliche „Abschnitte“ eines Objekt. Die Sub-Abstractions.

Es gibt beliebig viele Sub- Abstractions jedes räumlich- zeitlichen Objekts. Sie werden erzeugt durch.

- **Vergehen von Zeit (SebastianImAlterVon3Jahren).**
- **Events. (SebastianVorFriseurbesuch33, ...während...,...nach...)**

Idee | Architektur | Upper Level | Zerteilung | Ende

Zusammenfassung

- **Mehrfachvererbung**
- **Inferenzen auf Prädikaten: size (Oskar, around 170cm)**
- **Collections und Collections von Collections**
- **Teilbarkeit: Object/Substance, Structure und Granule, Sub- Abstractions**

Cyc gescheitert?

- **Der „Mid- Term Report“ ist 1990 erschienen und seitdem kein umfassender Bericht mehr.**
- **CSC- Bericht: Viel zu schwer an Infos zu kommen.
„Cyc has still a long way to go“**
- **CSC- Bericht: Cycorp hat kein vernünftiges Marketing- Personal und kein zur Vermarktung aufbereitetes Produkt.**
- **Das Sourceforge Team von OpenCyc hat nur 11 Mitwirkende.**
- **Vapour- Ware Award des Wired Magazins 2002.
Stellvertretend für die gesamte KI.**

References

- Lenat, D. B. and Guha, R. V. (1990). Cyc: A mid-term report.
- Lenat, D. B. and Guha, R. V. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Addison-Wesley, Reading, Massachusetts.
- <http://www.cyc.com>, <http://www.opencyc.org>,
<http://www.opencyc.org/doc/tut>
- Sanguino, R. (2001). Evaluation of Cyc ©CSC, Miami, Florida.

Warnung: Hierarchie und Benennung von Kategorien unterscheiden sich zwischen den Publikation und zu OpenCyc. Die hier verwendeten können von den tatsächlich aktuell implementierten abweichen.

