

---

# Ontologie-Management

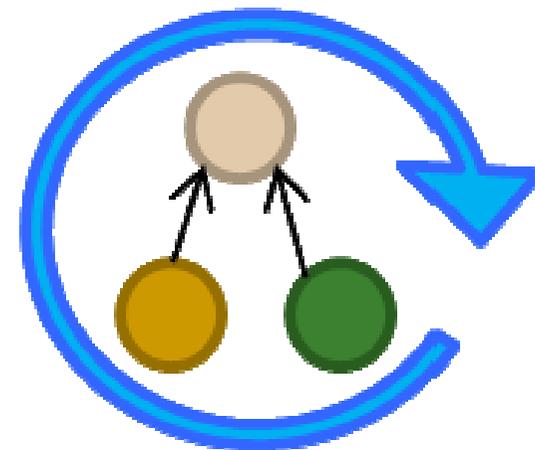
## *Kapitel 4: Erstellung von Ontologien*

Dr. Michael Hartung

Wintersemester 2012/13

---

Universität Leipzig  
Institut für Informatik  
<http://dbs.uni-leipzig.de>



---

# Inhalt

- Methoden des Ontologieentwurfs
  - Allgemeines Vorgehen
  - Methode von Uschold und King
  - Ontology Development 101
  - Weitere Methoden
- Ontology Learning
  - Prinzipielles Vorgehen
  - Ontology Learning im Entwurfsprozess



# Methoden des Ontologieentwurfs

## Definition

Eine Methodologie des Ontologie-Entwurfs beschreibt alle Aktivitäten die zur Konstruktion einer Ontologie notwendig sind

- Warum benötigt man eine formale Methodologie?
  - Entwicklung von konsistenten Ontologien
  - Effiziente Entwicklung komplexer Ontologien
  - Verteilte Entwicklung von Ontologien
- Unterscheidungen *(nach Fernandez-Lopez et. al., 1997)*
  - Ontology management activities
  - Ontology development oriented activities
  - Ontology support activities



---

# Ontology Management Activities

## ■ Scheduling

- Identifikation der durchzuführenden Aufgaben
- Arrangement/Planung der durchzuführenden Aufgaben
- Identifikation der benötigten Ressourcen  
(Zeit, Speicherplatz, etc...)

## ■ Control

- Garantiert korrekte Abwicklung der durchzuführenden Aufgaben

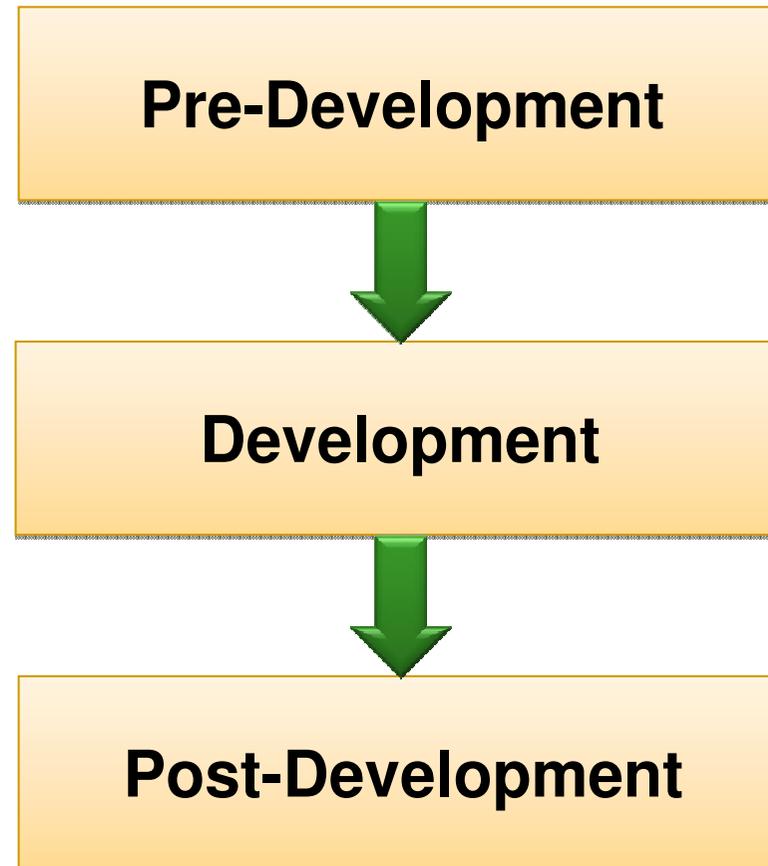
## ■ Quality Assurance

- Qualitätssicherung aller im Entwicklungsprozess anfallender Produkte (Ontologien, Software, Dokumentation)



# Ontology Development Oriented Activities

## Genereller Ablauf



---

# Pre-Development

- **Environment Study**

- Auf welchen Plattformen soll die Ontologie laufen?
- Für welche Anwendungen ist die Ontologie bestimmt?

- **Feasibility Study**

- Kann die Ontologie tatsächlich erstellt werden?
- Ist es überhaupt sinnvoll, die geplante Ontologie zu erstellen?



---

# Development

- **Specification**

- Warum wird die Ontologie erstellt, was ist der beabsichtigte Nutzen und wer sind die End-Anwender?

- **Conceptualization**

- Strukturiert Domain-Wissen in konzeptuellem Modell

- **Formalization**

- Formalisiert konzeptuelles Modell in semiberechenbarem Modell

- **Implementation**

- Konstruktion eines berechenbaren Modells in einer Ontologiesprache → *Kapitel 2*



---

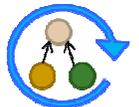
# Post-Development

- **Maintenance**

- Update und Korrektur der Ontologie (falls nötig)  
→ *Kapitel 6: Dynamik in Ontologien*

- **Use / Reuse**

- Einsatz der Ontologie in den geplanten Anwendungen  
→ *Kapitel 3*
- Verwendung in anderen Ontologien



---

# Ontology Support Activities

- **Knowledge Acquisition**

- Wissen von Experten (semi-)automatisch gewinnen (Ontology Learning)

- **Evaluation**

- Technische Überprüfung der Ontologien in jeder Stufe der Entwicklung

- **Integration**

- Wiederverwendung bereits existierender Ontologien (Ontology Reuse)

- **Merging**

- Konstruktion einer neuen Ontologie aus bereits existierenden innerhalb einer bestimmten Domain → *Kapitel 7*



---

# Ontology Support Activities (II)

- **Matching / Alignment**

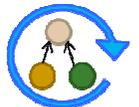
- Mapping zwischen den beteiligten Ontologien → *Kapitel 5*

- **Documentation**

- Jede Stufe der Ontologie-Entwicklung wird akkurat dokumentiert

- **Configuration Management**

- Verwaltet alle Versionen der Dokumentation und der entwickelten Ontologie → *Kapitel 6*



# Überblick über Aktivitäten

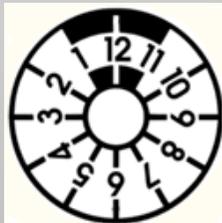
## Management



scheduling



control



quality assurance

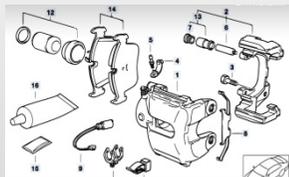
## Development Oriented



environment study



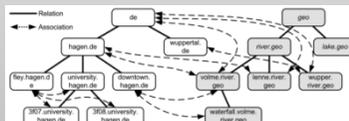
feasibility study



specification



conceptualization



formalization



implementation



maintenance



use / reuse

## Support



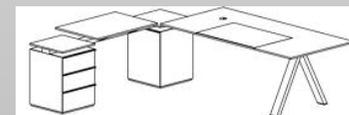
knowledge acquisition



evaluation / integration



documentation / merging

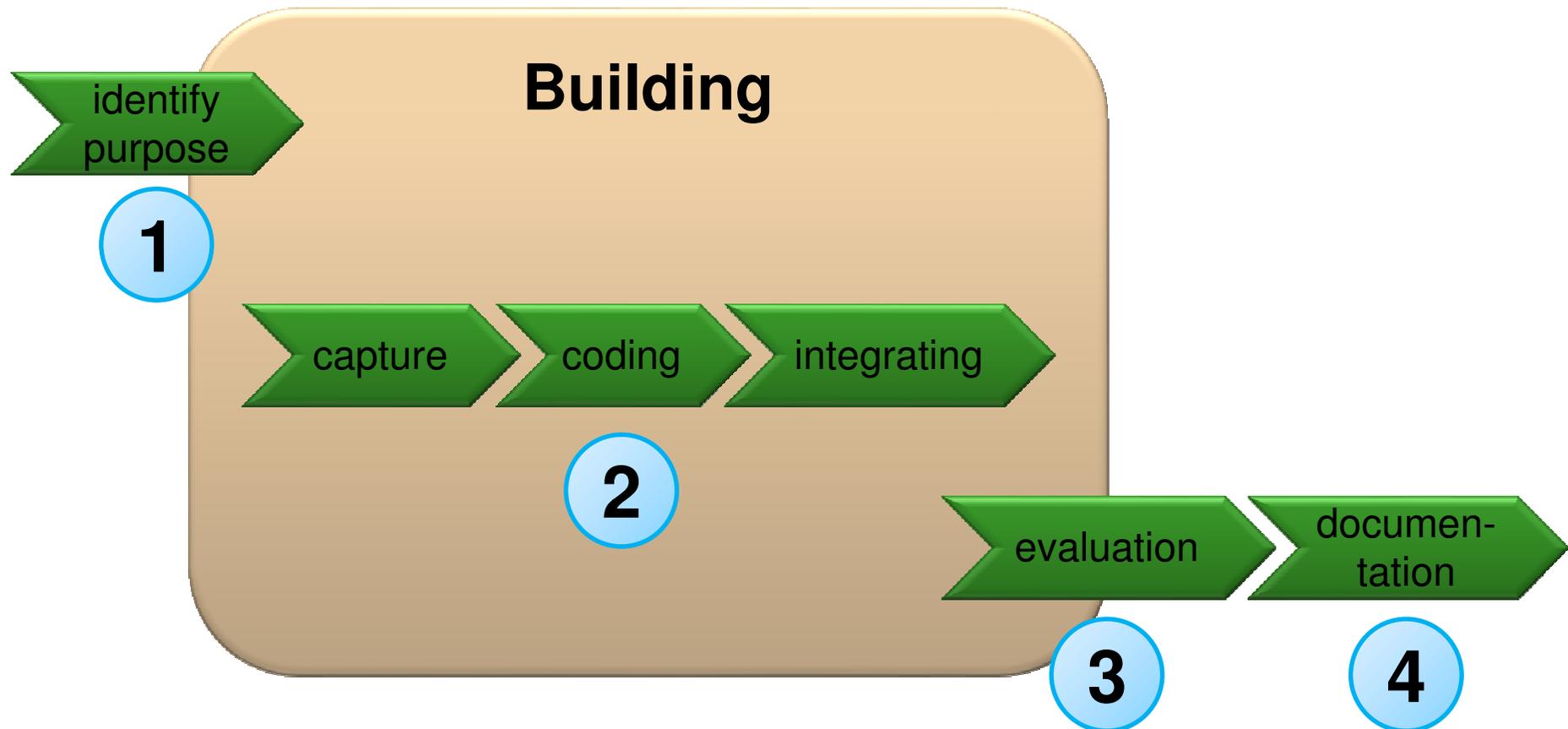


configuration management / matching



# Methode nach Uschold und King

- Prozessbasierte Entwicklung



*M. Uschold, M. King: Towards a Methodology for Building Ontologies, 1995.*



# Identify Purpose

## Identifiziere Zweck und Anwendungsgebiet

- ❑ Warum wird Ontologie benötigt?
- ❑ Vorgesehene Verwendung / Applikation
  - Simple use, reuse, share, Teil einer Wissensbasis, ...
- ❑ Identifikation relevanter Begriffe

### Beispiel: *Reise-Ontologie*

- Aufbau eines gemeinsamen Wissensmodell über das Wissensgebiet Reisen, das in Reisebüros genutzt werden soll
- Ontologie könnte auch für andere Anwendungsgebiete genutzt werden, z.B. um einen Katalog für Unterkünfte oder Transportmöglichkeiten zu entwickeln
- relevante Begriffe z.B.: **Orte, Typen von Orten, Unterkünfte, Arten von Unterkünften (Hotel / Motel / Camping / ...), Bahn, Busse, U-Bahn, ...**



# Building – Ontology Capture

## Ontologie Aufbau

- Identifiziere Schlüsselkonzepte (Klassen) und Beziehungen (Relationen) des betreffenden Wissensgebiets und gebe diese in textueller Form an

### Beispiel: *Reise-Ontologie*

- *Transportmittel* ist eine Klasse. Jeder Transport besitzt einen *Startpunkt*
- *Bus* ist eine Klasse. Bus ist ein bestimmtes *Transportmittel*.
- *Stadtbus* ist eine Klasse. Ein Stadtbus ist ein Bus, dessen *Start-* und *Zielpunkt* sowie dessen *Zwischenstopps* in derselben *Stadt* liegen.

- Identifikation der Ontologiekonzepte
  - Bottom-Up / Top-Down / Middle-Out



# Building – Ontology Capture

- **Bottom-Up Identifikation von Ontologiekonzepten**
  - Erhöhter Gesamtaufwand
  - Schwierige Findung von Gemeinsamkeiten zwischen verwandten Konzepten
  - Erhöhtes Risiko von Inkonsistenzen → erfordert wiederum Überarbeitung (steigender Aufwand)

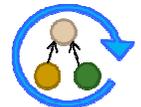
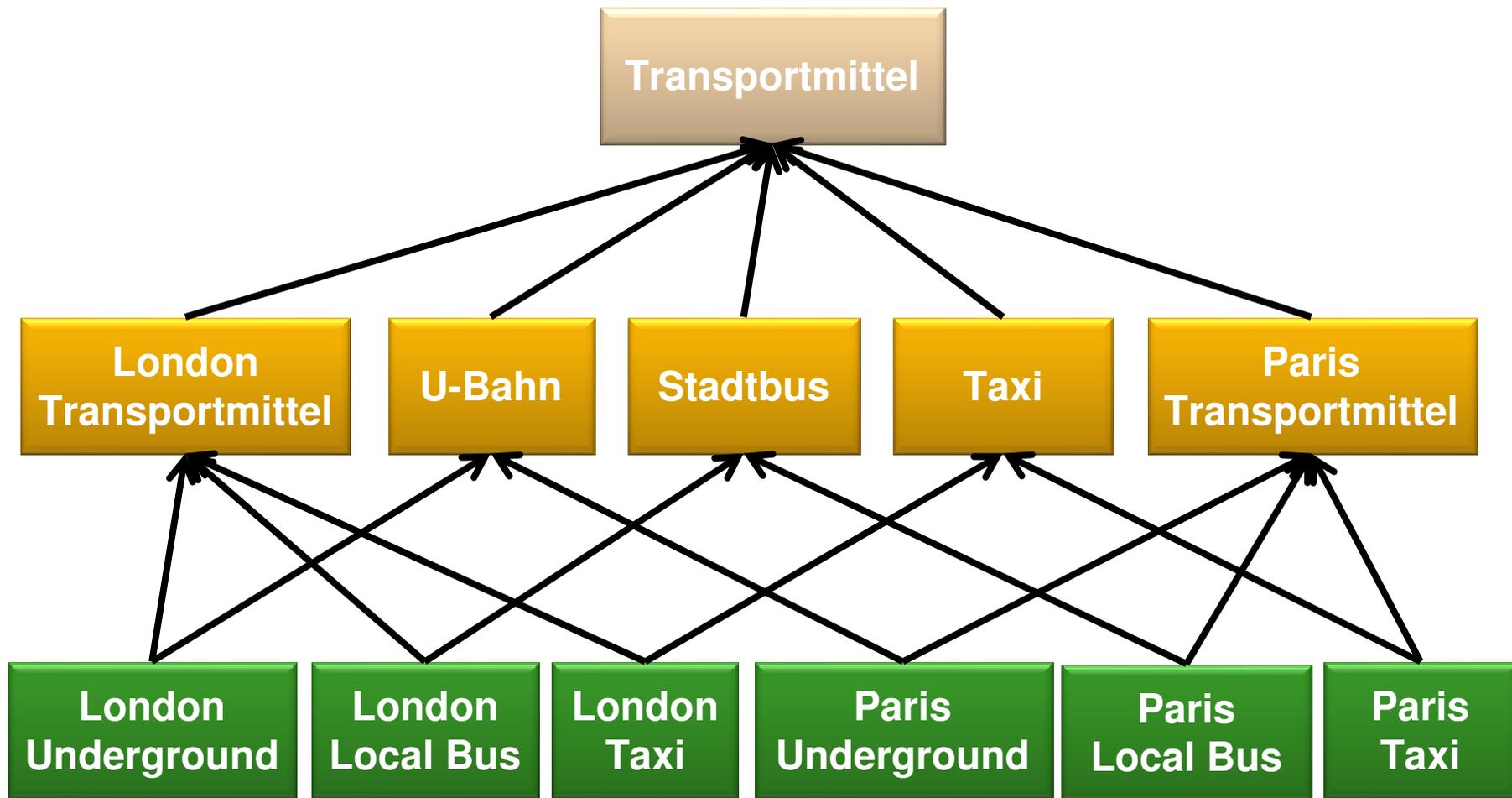
Beispiel: ***Reise-Ontologie***

•Transportmittel sollen in einer Bottom-Up Strategie konzeptualisiert werden



# Building – Ontology Capture

- Bottom-Up Identifikation von Ontologiekonzepten



# Building – Ontology Capture

- **Top-Down Identifikation von Ontologiekonzepten**
  - bessere Kontrolle des Detaillierungsgrades
  - möglicherweise werden aber abstrakte Kategorien gar nicht benötigt
  - geringere Stabilität des Modells → dann Überarbeitung notwendig (mehr Aufwand)

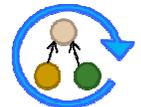
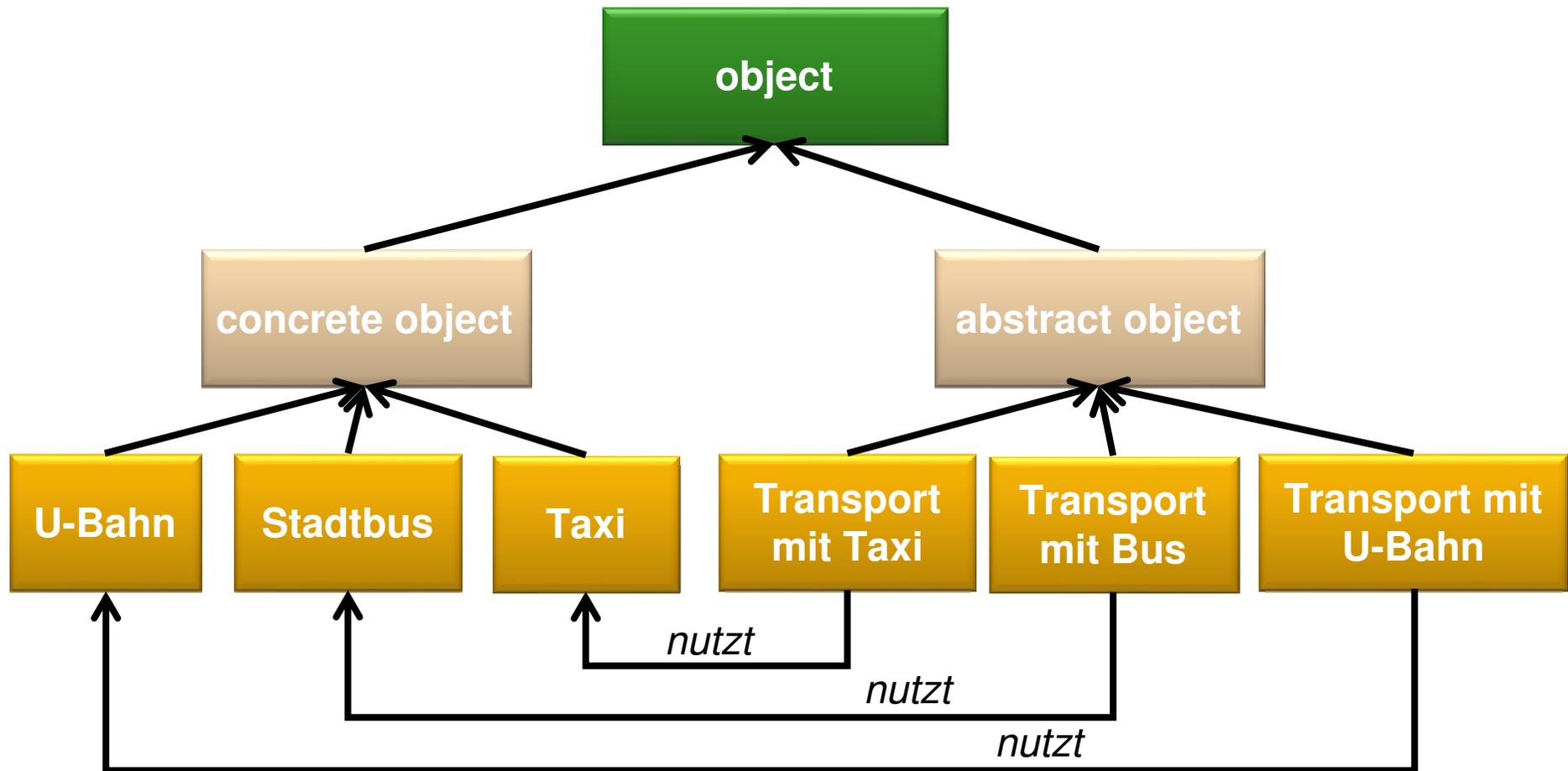
Beispiel: ***Reise-Ontologie***

- Transportmittel sollen in einer **Top-Down** Strategie konzeptualisiert werden



# Building – Ontology Capture

- Top-Down Identifikation von Ontologiekonzepten



# Building – Ontology Capture

- **Middle-Out Identifikation von Ontologiekonzepten**
  - Starte mit Kern / Grundbegriffe, dann Spezialisierung / Generalisierung
  - ausbalanciert (bzgl. Detaillierungs-/Abstraktionsgrad)
  - stabiler als die beiden vorherigen Verfahren

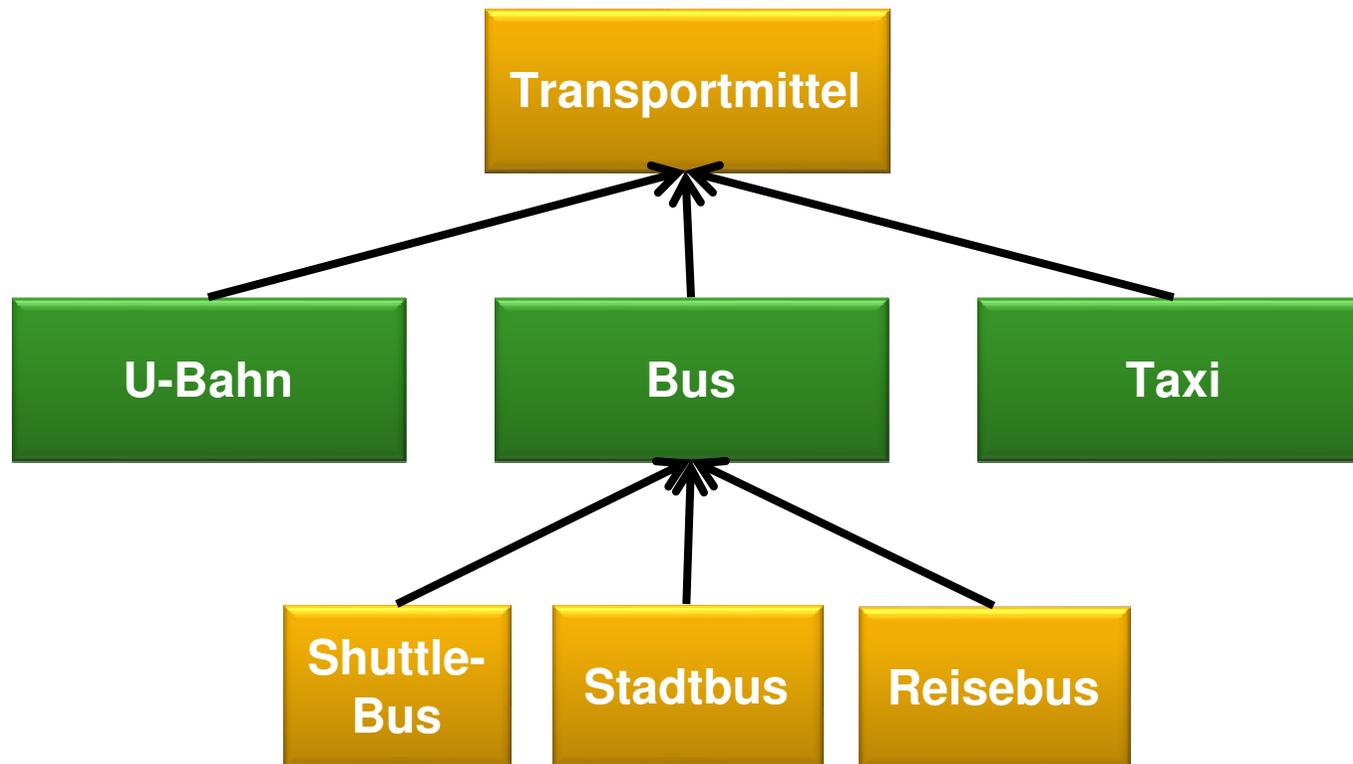
Beispiel: ***Reise-Ontologie***

- Transportmittel sollen in einer **Middle-Out** Strategie konzeptualisiert werden



# Building – Ontology Capture

- Middle-Out Identifikation von Ontologiekonzepten



---

# Evaluation & Documentation

## Evaluation

- Technische Überprüfung der Ontologien und der damit assoziierten Software in jeder Stufe der Entwicklung

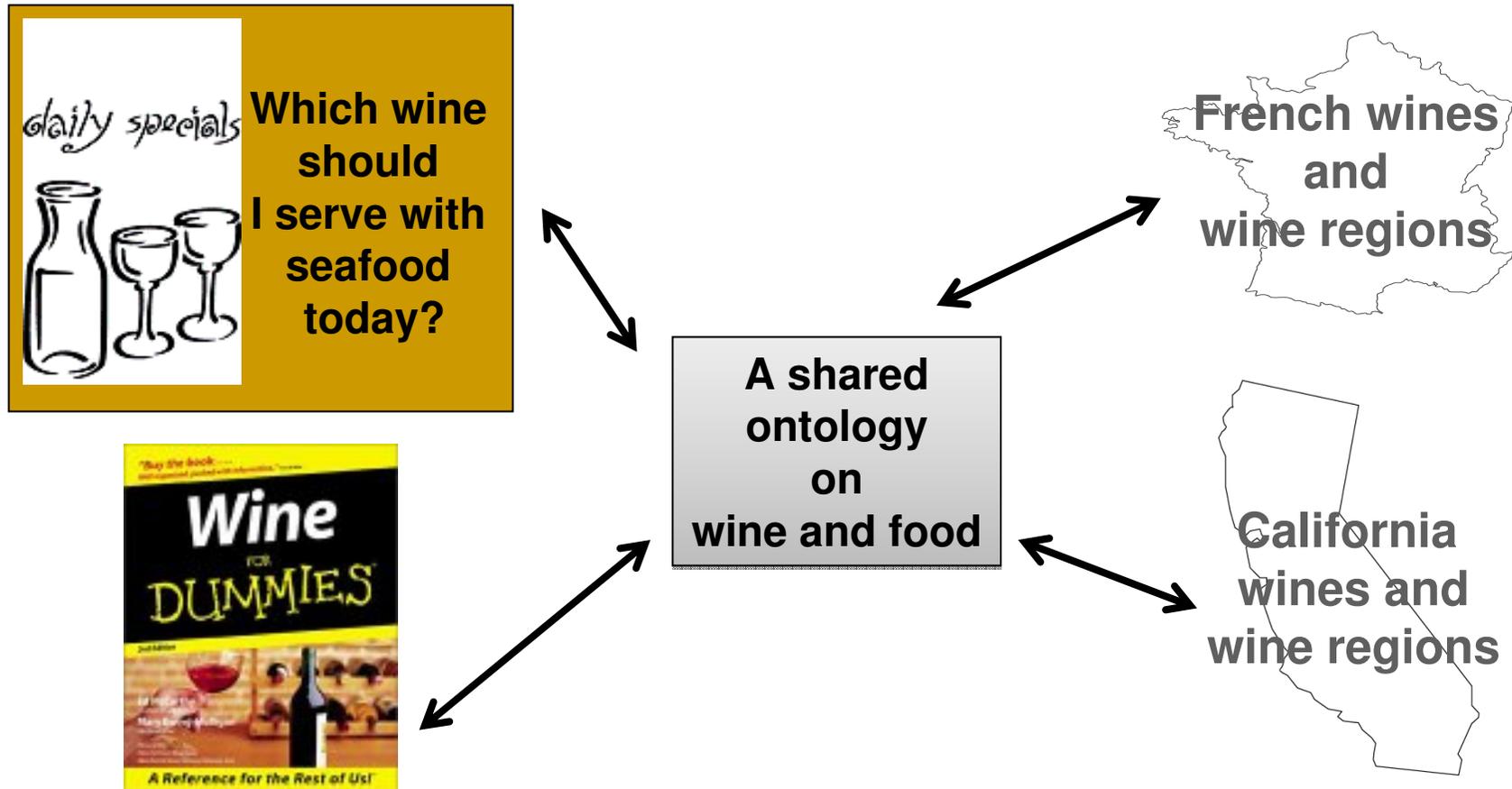
## Documentation

- Einrichten der eventuell unterschiedlichen Dokumentationsrichtlinien



# Ontology Development 101

**Beispiel:** Ontologie für Weinbau und Nahrungsmittel

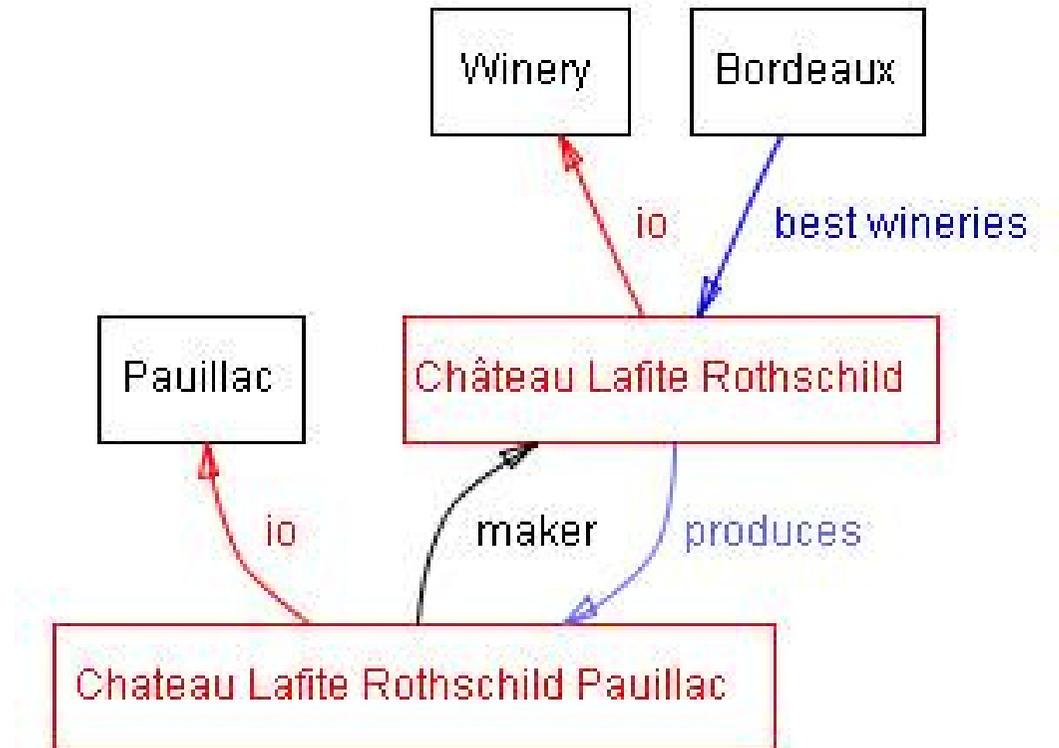


*N.F. Noy, D. McGuinness: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory, 2001.*



# Ontology Development 101

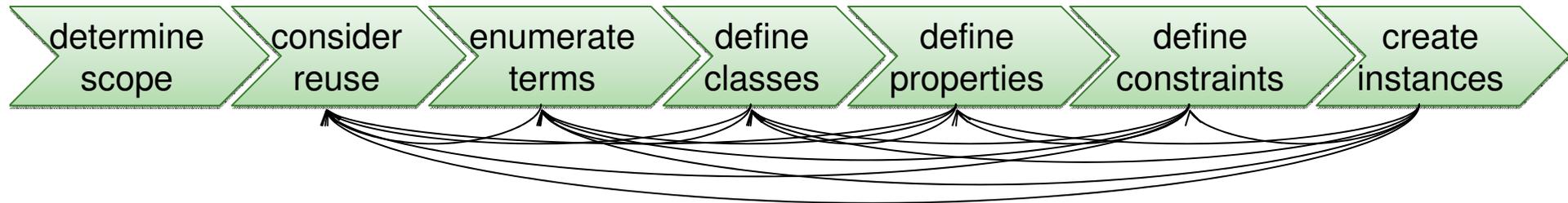
**Beispiel:** Ontologie für Weinbau und Nahrungsmittel



*N.F. Noy, D. McGuinness: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory, 2001.*



# Ontology Development Process



- in der Praxis **iterativer Prozess**, der **sich beständig wiederholt** und die Ontologie verbessert
- es gibt immer **unterschiedliche Vorgehensweisen** bei der Modellierung einer Ontologie
- in der Praxis entscheidet immer die angestrebte Anwendung über das Vorgehen bei der Modellierung

*„There is no one correct way to model a domain  
there are always viable alternatives.“*



# Beurteile Fachgebiet und Fokus



- Welches **Fachgebiet** soll die Ontologie abdecken?
- **Wozu** soll die Ontologie genutzt werden?
- Welche Arten von **Fragen** sollen die in der Ontologie repräsentierten Informationen beantworten können?
- **Wer** wird die Ontologie pflegen und nutzen?
- Formulierung von **Kompetenzfragen**

*Im Laufe des Lebenszyklus einer Ontologie können sich diese Fragen auch ändern ! (Evolution)*



# Beurteile Fachgebiet und Fokus



## ***Kompetenzfragen zur Wine Ontology***

- *Welche Eigenschaften des Weins sollen bei der Modellierung berücksichtigt werden?*
- *Ist ein Bordeaux ein Weißwein oder ein Rotwein?*
- *Passt Cabernet Sauvignon gut zu Fischgerichten?*
- *Welcher Wein passt am besten zu gegrilltem Fleisch?*
- *Welche Eigenschaften eines Weins beeinflussen seine Eignung zu einem bestimmten Gericht?*
- *Ändert sich das Bouquet eines Weins mit unterschiedlichen Jahrgängen?*
- ...



# Berücksichtigung von Wiederverwendung



- Warum Wiederverwendung?
  - Einsparung von **Aufwand**
  - Reuse von **Werkzeugen** die mit anderen Ontologien arbeiten für die Eigene
  - Wiederverwendung von erfolgreich validierten Ontologien

*Wenn keine passende Ontologie existiert oder der Aufwand zur Anpassung zu groß wird, dann erschaffe eine Neue !*



# Terminologie entwickeln



- Von welchen **Begriffen** soll die Rede sein?
- Welche **Eigenschaften** haben diese Begriffe?
- **Was** wollen wir über diese Begriffe **aussagen**?

## ***Beispiel: Wine Ontology***

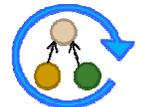
- *wine, grape, winery, location,...*
- *a wine's color, body, flavor, sugar content,...*
- *subtypes of wine: white wine, red wine, Bordeaux wine,...*
- *types of food: seafood, fish, meat, vegetables, cheese,...*
- ...



# Klassen und Klassenhierarchien definieren



- **Klassen** sind Konzepte in der betreffenden Domäne
  - *Klasse der Weine*
  - *Klasse der Weinbaubetriebe*
  - *Klasse der Rotweine*
- Klassen sind Sammlungen von Objekten mit gleichartigen Eigenschaften
- Wähle Top-Down / Bottom-Up / Middle-Out Ansatz zur Modellierung der Klassenhierarchien



# Definiere Properties



- **Properties** in einer Klassendefinition beschreiben Attribute von Instanzen
  - *Jeder Wein hat eine Farbe, Restzuckergehalt, Produzent, ...*

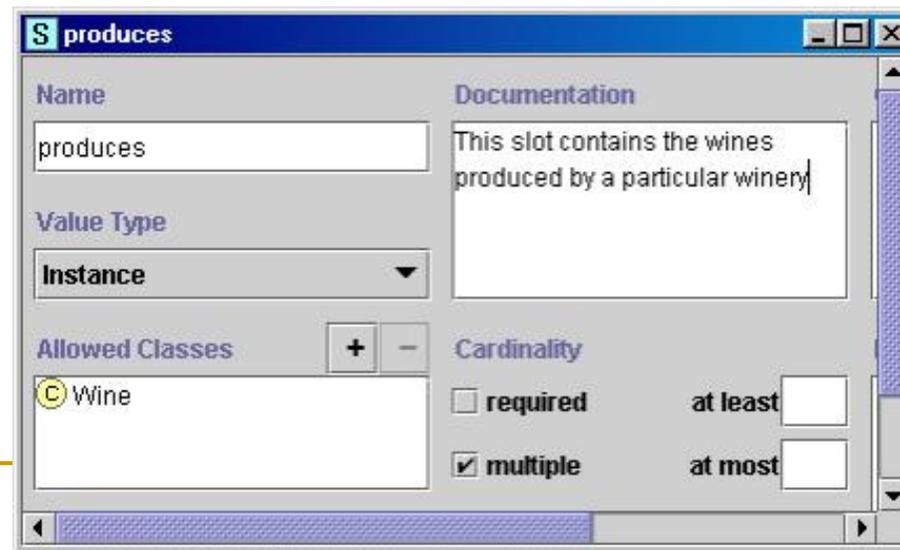
Template Slots				V	V	C	X	+	-
Name	Type	Cardinality	Other Facets						
<b>S</b> body	Symbol	single	allowed-values={FULL,MEDIUM,LIGHT}						
<b>S</b> color	Symbol	single	allowed-values={RED,ROSÉ,WHITE}						
<b>S</b> flavor	Symbol	single	allowed-values={DELICATE,MODERATE,STRONG}						
<b>S</b> grape	Instance	multiple	classes={Wine grape}						
<b>S</b> maker <sup>I</sup>	Instance	single	classes={Winery}						
<b>S</b> name	String	single							
<b>S</b> sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}						



# Definiere Beschränkungen auf Properties



- **Property constraints** (Restriktionen) beschreiben bzw. beschränken die Menge der möglichen Property-Werte
  - *Der Name eines Weines ist ein String*
  - *„produces“ in „Winery“ kann Instanzen der Klasse „Wine“ als Value aufweisen*



# Definition von Klasseninstanzen



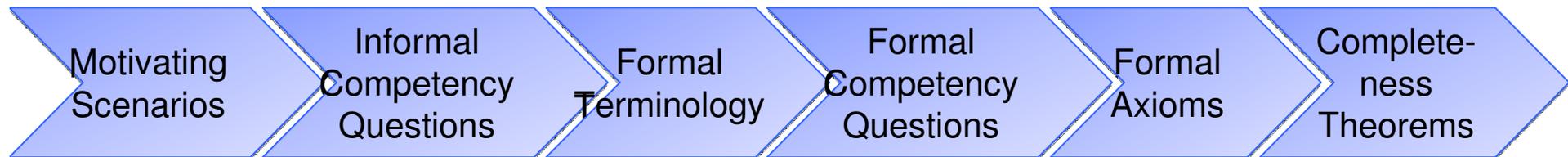
- Erzeuge die **Instanzen** der Klassen
- Jede Klasse wird zum direkten **Typen** für ihre Instanzen
- Jede Superklasse eines direkten Typs ist Typ der Instanz
- Zuweisung von Property-Werten für Instanzen entsprechend Constraints

*„das Glas spanischen Rotweins, das xy gestern Abend zum Abendessen getrunken hat“*



# Methode nach Grüninger und Fox

- Formaler Ansatz des Ontologie-Entwurfs
- Inspiriert durch den Entwurf wissensbasierter Systeme (Prädikatenlogik)



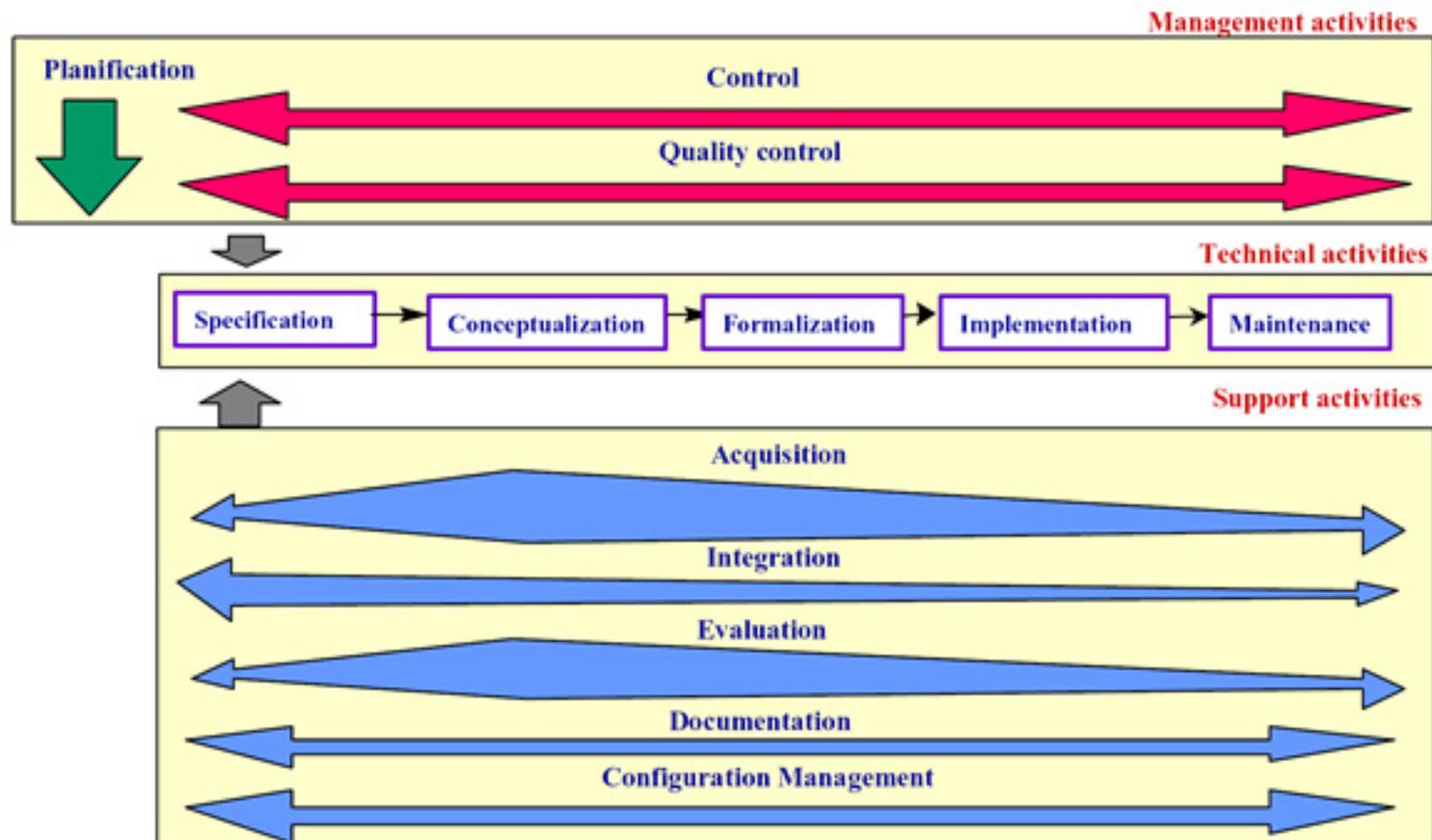
1. Identifikation des Hauptanwendungsszenarios
2. Kompetenzfragen, um Spielraum/Fachgebiet der Ontologie festzulegen
3. Extraktion der Terminologie/Konzepte/Relationen, ...
4. Formale Kompetenzfragen (mit eigens entwickelter Terminologie)
5. Bestimmung der wichtigsten Axiome
6. Überprüfung der Vollständigkeit

*M. Grüninger, M. Fox: Methodology for the Design and Evaluation of Ontologies, 1995.*



# METHONTOLOGY

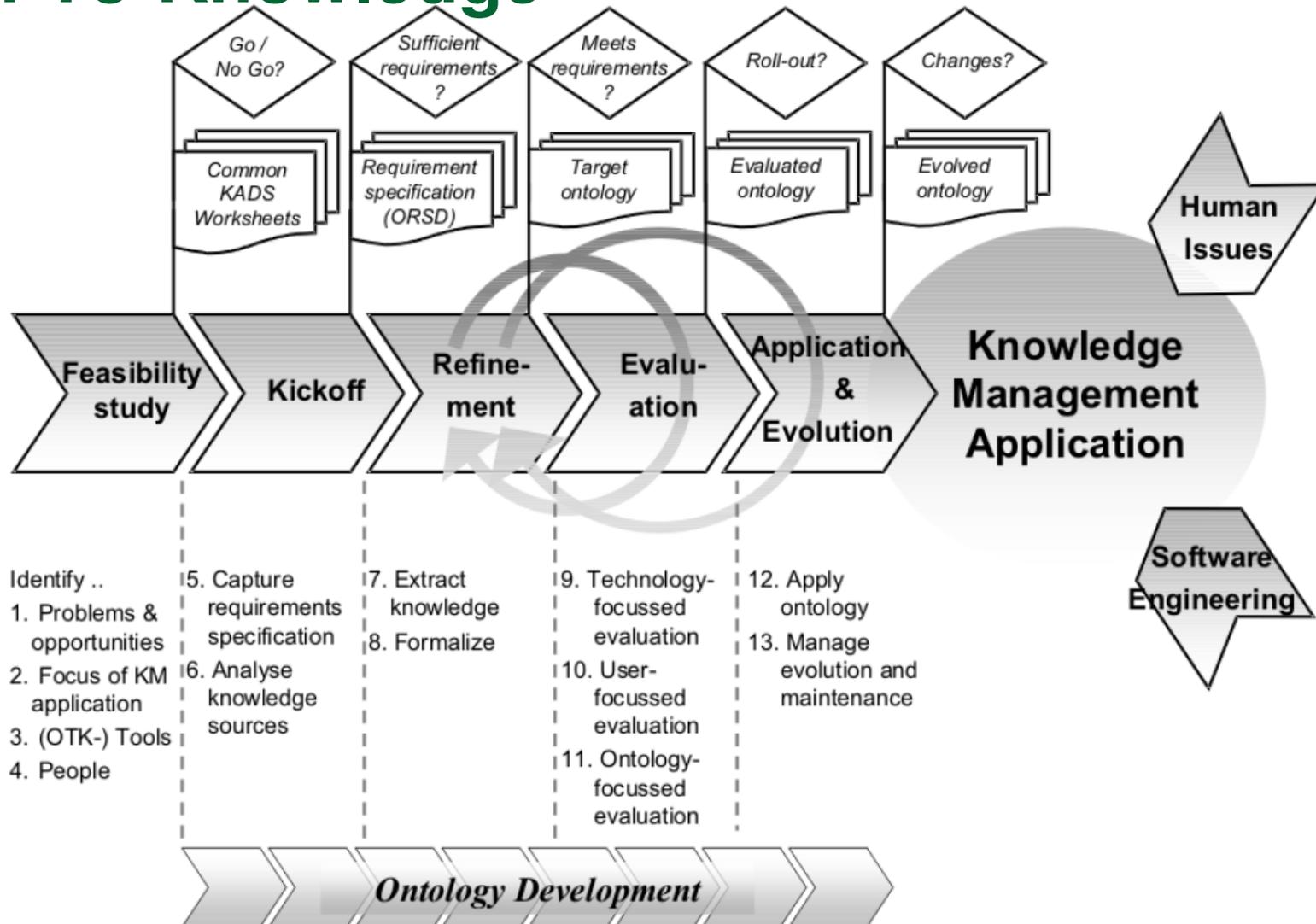
- Lebenszyklus-Modell basierend auf entwickelten Prototypen



*Fernandez-Lopez/Gomez-Perez, 1999*



# On-To-Knowledge



Y. Sure, S. Staab, R. Studer: Methodology for Development and Employment of Ontology based Knowledge Management Applications. *Sigmod Record* 31(4), 2002.



# Ontology Learning

- Ontologie-Entwurf ist sehr aufwändig bzgl. Zeit und Ressourcen
  - kann das Verfahren (teil-)automatisiert werden?
- Ontologien können (automatisch) „gelernt“ werden

## Ontology Learning definiert Methoden und Techniken

- zum **grundlegenden Aufbau** einer neuen Ontologie
  - zur **Erweiterung oder Anpassung** einer bereits existierenden Ontologie
  - in einer (teil-)automatisierten Weise aus unterschiedlichen Ressourcen
- Automatisierung basiert u.a. auf
    - Natural Language Processing (NLP)
    - Data Mining, Text Mining
    - Machine Learning Techniken (ML)



---

# Datenquellen für Ontology Learning

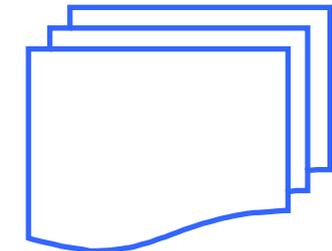
## Welche Datenquellen können für Ontology Learning verwendet werden?

- Strukturierte Daten (Datenbanken, Wissensbasen, ...)
  - Data Mining, Machine Learning
- Semi-strukturierte Daten (XML Dokumente, ...)
  - Data Mining, Natural Language Processing, Machine Learning
- Unstrukturierte Daten (Texte in natürlicher Sprache)
  - Text Mining, Natural Language Processing, Machine Learning

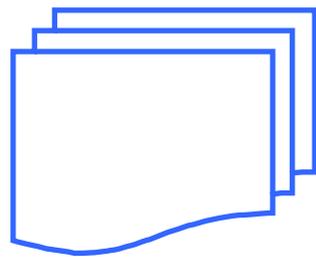


# Ontology Learning – Prinzipielles Vorgehen

Dokumentenkorpus



...



(1) *Term-  
extraktion*



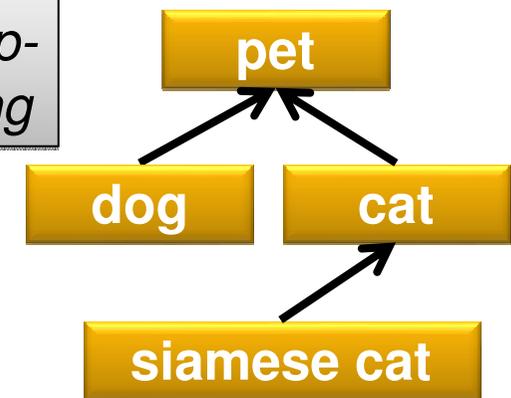
Terme

<dog> <dogs>  
<cat> <felis catus>  
<siamese cat>

(2) *Konzept-  
tualisierung*



Ontologie



(3) *Evaluation  
und Nachbesserung*



**Semiautomatischer Prozess**



---

# Ontology Learning – NLP

## Natural Language Processing

1. Tokensizer / Sentence Splitter
2. Morphologische Analyse
  - Stemming (Grundform) / Lemmatizer
3. POS-Tagger
  - Syntaktische Kategorien  
(Verb, Hauptwort, Präposition, ...)
4. Regular Expression Matching
5. Chunks
  - Erfassung größerer zusammenhängender Gebilde in Sätzen
6. Syntactic Parser

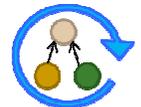


---

# Ontology Learning Tasks

**Welche Aufgaben im Rahmen des Ontologieentwurfs können (teil-)automatisiert werden?**

1. Ontology creation
2. Ontology schema extraction
3. Extraction of ontology instances
4. Ontology integration and navigation
5. Ontology update
6. Ontology enrichment



---

# Ontology Learning Tasks (2)

## ■ **Ontology creation**

- Entwurf von Grund auf (from the scratch) durch einen Experten
- *Maschinelles Lernen* (ML) unterstützt den Experten beim
  - Entwurf durch Vorschlag von geeigneten Relationen zwischen den Konzepten
  - Überprüfung der Integrität/Konsistenz der entworfenen Ontologie

## ■ **Ontology schema extraction**

- Extraktion von Schemata aus Web-Dokumenten / Texten
- *ML* benutzt Eingabedaten und Meta-Ontologien, um fertige Domain-Ontologien (ggf. mit Hilfe des Experten) zu erzeugen



---

# Ontology Learning Tasks (3)

- **Extraction of ontology instances**
  - Extraktion von Objekten aus semi-strukturierten und unstrukturierten Daten, um vorgegebene Ontologie-Schemata mit Instanzen zu füllen
  - nutzt Techniken aus den Bereichen *Information Retrieval* und *Data Mining*
- **Ontology integration and navigation**
  - Umbau (Rekonstruktion) von existierenden Wissensbasen und Navigation in vorhandenen Wissensbasen
  - z.B. Übersetzung einer Wissensbasis aus FOL nach OWL
  - (Teil-)automatisiertes Mapping/Merging zum Zusammenführen mehrerer Ontologien



---

# Ontology Learning Tasks (4)

## ■ **Ontology update**

- Erweiterung, Umbau und Veränderung von bereits bestehenden Ontologien, z.B. zur Anpassung an veränderte Domain
- betrifft Teilbereiche von Ontologien, die speziell so angelegt wurden, dass diese verändert werden können

## ■ **Ontology enrichment**

- (auch *Ontology tuning*) betrifft automatische Aktualisierung kleinerer Teilbereiche bereits existierender Ontologien
- verändert nicht wichtige (bedeutende) Konzepte und Relationen, sondern präzisiert diese



---

# Zusammenfassung

## ■ Methodologie des Ontologieentwurfs

- Alle Aktivitäten die zur Konstruktion einer Ontologie notwendig sind
- Warum: Konsistente und komplexe Ontologien, verteilte Entwicklung
- Unterschiedliche Methoden
  - Uschold und King
  - Ontology Development 101
  - ...

## ■ Ontology Learning

- (teil-)automatisierte Unterstützung des Ontologieentwurfs
- Nutzung vers. Techniken
  - Data Mining, Natural Language Processing, Machine Learning

