# (Semi-)Automatische Ontologieerstellung

## Werkzeuge und Algorithmen

von Robert Engsterhold

Betreuer: Dr. Andreas Thor

# Übersicht

- Werkzeuge
  - Text2Onto (TextToOnto)
  - OntoLT
  - OntoLearn
- Methoden
  - Probabilistic Ontology Model
  - Structural Semantic Interconnections

# Text2Onto

- Text2Onto ist ein Programm zur automatischen Ontologieerstellung
- Es extrahiert Konzepte, Instanzen und Relationen aus Text, XML, HTML, PDF Dateien
- Es bietet zu jedem extrahierten Term einen Wahrscheinlichkeitswert, der Anzeigt wie sicher sich Text2Onto mit dem Term ist.
- Der Benutzer kann diese Confidence Werte manuell ändern.

# Text2Onto

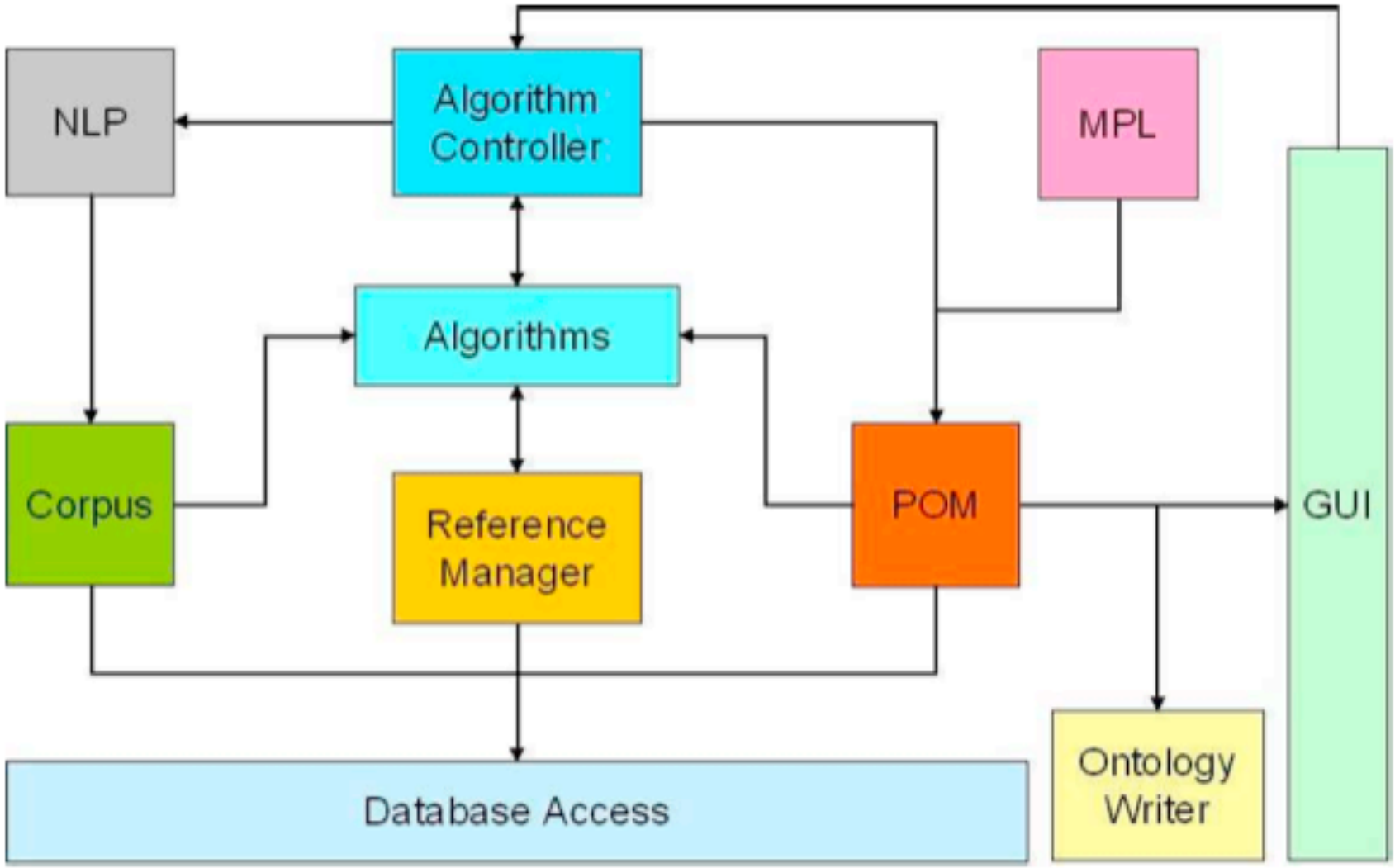



**Fig. 1.** Architecture of Text2Onto

# Text2Onto

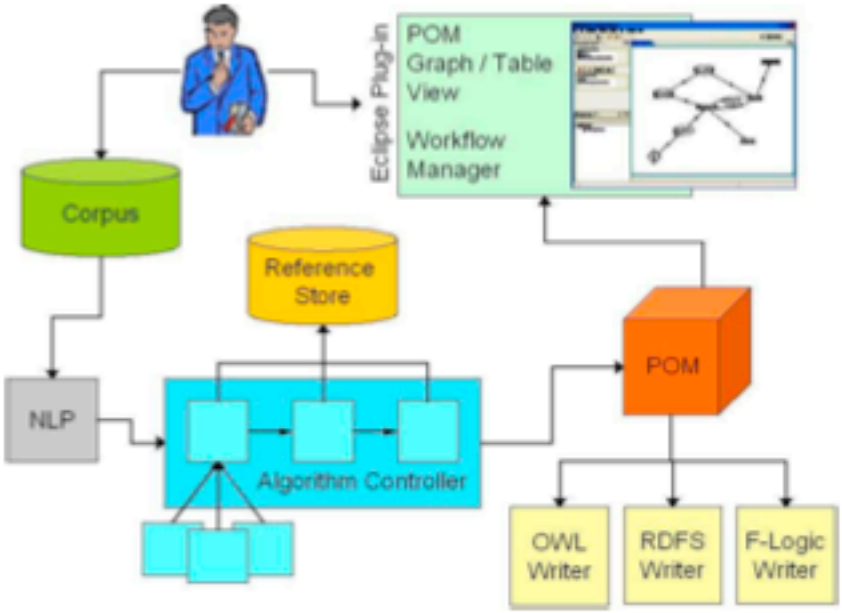
- Zentrale Komponente: POM (Probabilistic Ontology Model) wird später genauer erklärt.
- NPL: Natural Language Processing. Verwendet Gate (General Architecture for Text Engineering) und JAPE (Java Annotation Patterns Engine).

# Text2Onto

- Algorithmen:
  - Normalisierte Meßalgorithmen: RTF, TFIDF, Entropy C-value /NC-value
  - Subclass-of Relation Algorithmen auf WordNet Basis
  - Mereological Relations: Jape expression matching, indicating part-of relations
  - General Relations: transitiv, intransitiv + PP-complement, transitiv +PP-complement
  - Instance-of Relation:  Skewed divergence
  - Equivalence: Corpus-based similarity

# Text2Onto

- GUI:



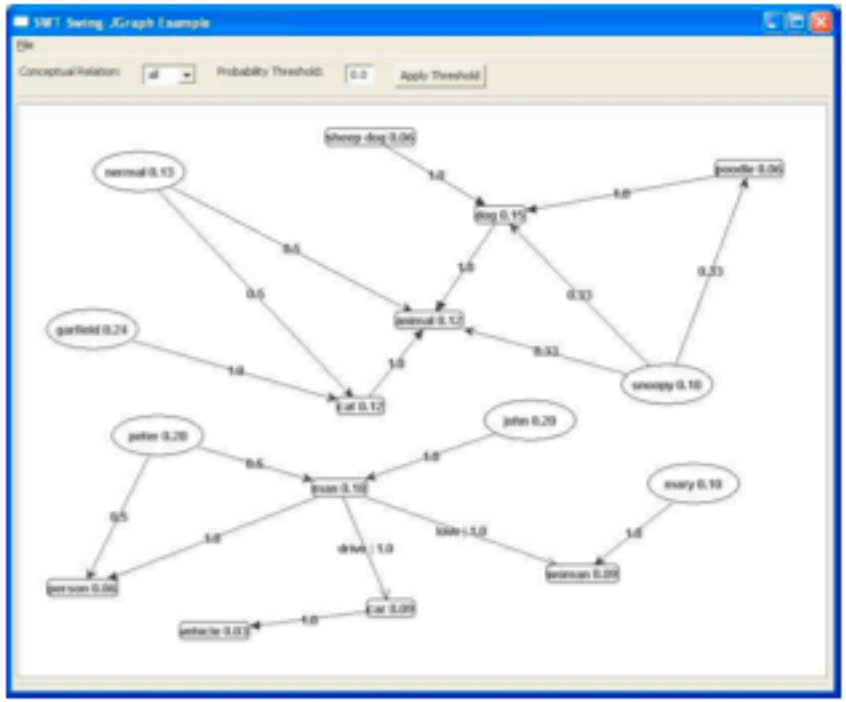
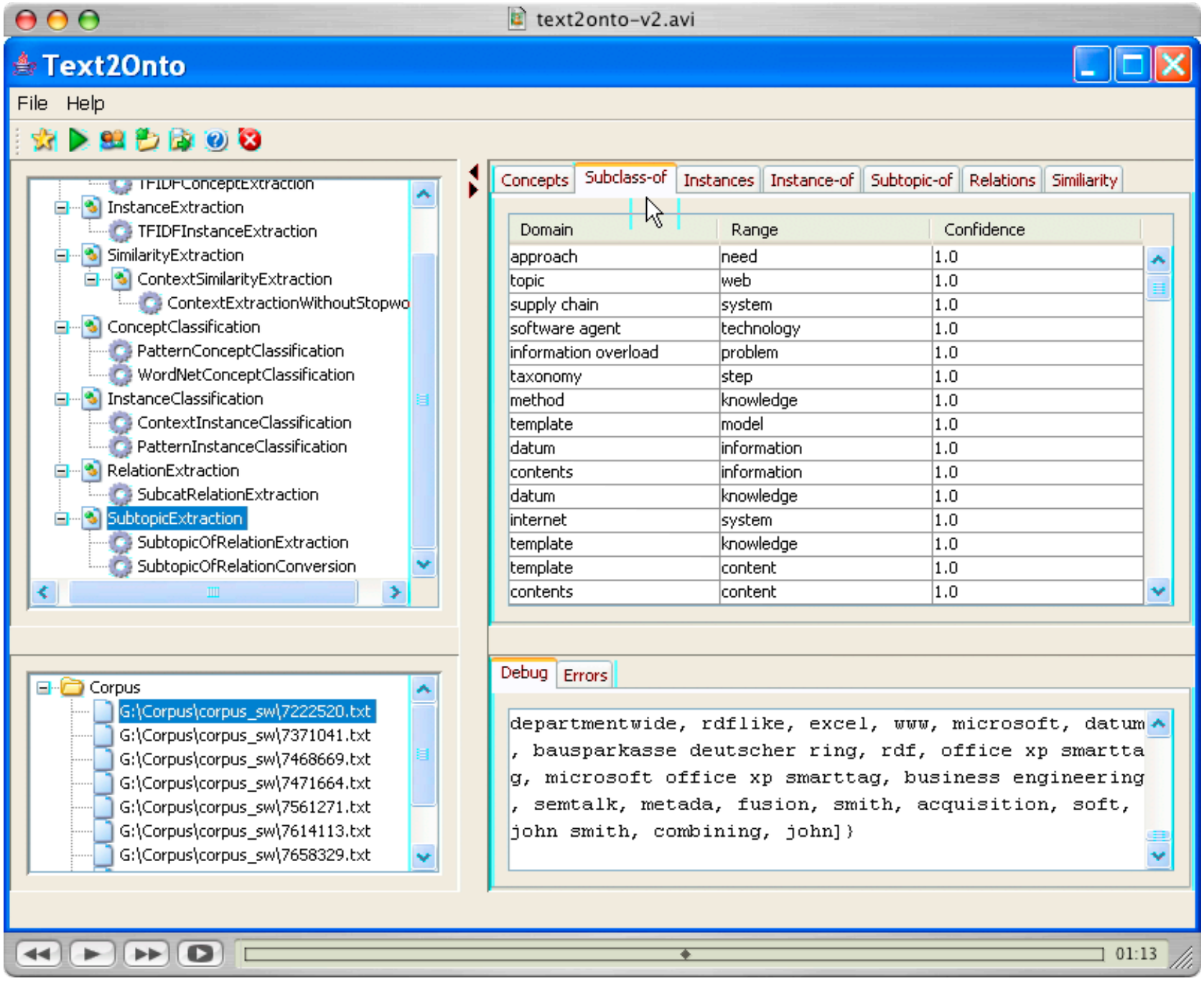



**Fig. 2.** Usage Scenario (left) and POM visualization (right)

# Text2Onto


text2onto-v2.avi
Text2Onto
File Help
TFIDFConceptExtraction
InstanceExtraction
TFIDFInstanceExtraction
SimilarityExtraction
ContextSimilarityExtraction
ContextExtractionWithoutStopwo
ConceptClassification
PatternConceptClassification
WordNetConceptClassification
InstanceClassification
ContextInstanceClassification
PatternInstanceClassification
RelationExtraction
SubcatRelationExtraction
SubtopicExtraction
SubtopicOfRelationExtraction
SubtopicOfRelationConversion
Corpus
G:\Corpus\corpus_sw\7222520.txt
G:\Corpus\corpus_sw\7371041.txt
G:\Corpus\corpus_sw\7468669.txt
G:\Corpus\corpus_sw\7471664.txt
G:\Corpus\corpus_sw\7561271.txt
G:\Corpus\corpus_sw\7614113.txt
G:\Corpus\corpus_sw\7658329.txt
Concepts Subclass-of Instances Instance-of Subtopic-of Relations Similiarity
Domain Range Confidence
approach need 1.0
topic web 1.0
supply chain system 1.0
software agent technology 1.0
information overload problem 1.0
taxonomy step 1.0
method knowledge 1.0
template model 1.0
datum information 1.0
contents information 1.0
datum knowledge 1.0
internet system 1.0
template knowledge 1.0
template content 1.0
contents content 1.0
Debug Errors
departmentwide, rdflike, excel, www, microsoft, datum
, bausparkasse deutscher ring, rdf, office xp smartta
g, microsoft office xp smarttag, business engineering
, semtalk, metada, fusion, smith, acquisition, soft,
john smith, combining, john]}
01:13

# Text2Onto

- http://ontoware.org/projects/text2onto/
- Der Nachfolger von TextToOnto
- Als Plug-In für das NeOn Toolkit verfügbar
- Das NeOn Toolkit ist eine Ontolgy Engineering Enviroment basierend auf Eclipse
- http://www.neon-toolkit.org/

# Text2Onto

- Benutzt GATE und WordNet
- GATE steht für General Architecture for Text Engineering. Ein Tool um Information aus natürlicher Sprache zu extrahieren
- WordNet ist eine Datenbank, die semantische und lexikalische Beziehungen zwischen Wörtern enthält.

# OntoLT

- Ein Protege Plugin zur semi-automatischen Ontologieerstellung.
- Benötigt einen Annotated Corpus im XML Format als Eingabe basierend auf SCHUG.
- Dient zur Extraktion von Konzepten und Relationen (Classes and Slots in Protege).
- Die Ergenisse werden in Protege weiterverwendet.

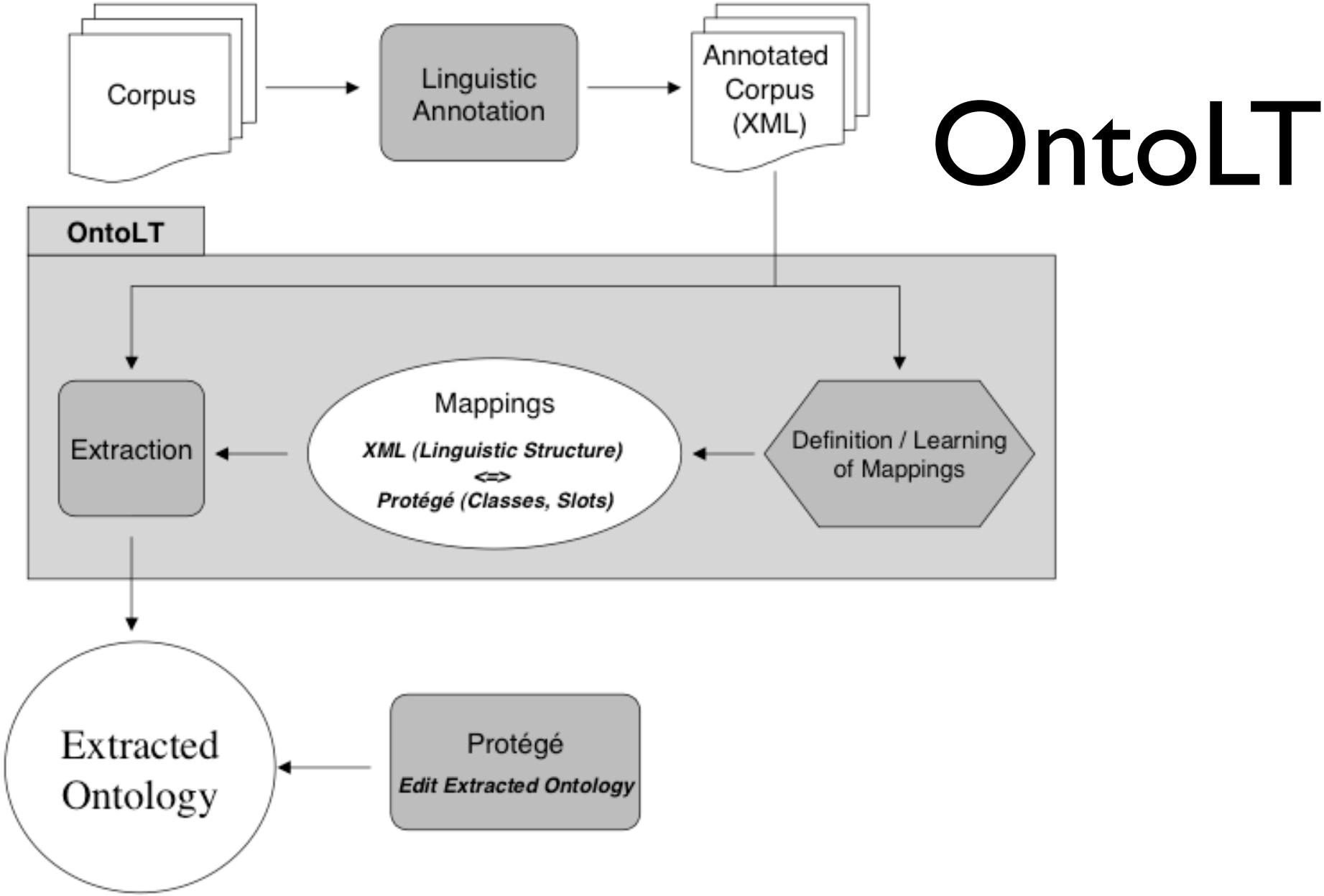



**Figure 1: Overview of the OntoLT Approach**

# OntoLT

- Über Vorbedingungen lassen sich Regeln fürs Mapping aufstellen. (Precondition Language)
- Diese Vorbedingungen sind XPATH Ausdrücke über den Annotated Corpus.

# OntoLT

- Vorgegebene Mapping Regeln sind:
  - HeadNounToClass_ModToSubClass
  - SubjToClass_PredToSlot_DObjToRange
- Eigene Regeln können manuell oder durch maschinelles Lernen hinzugefügt werden.

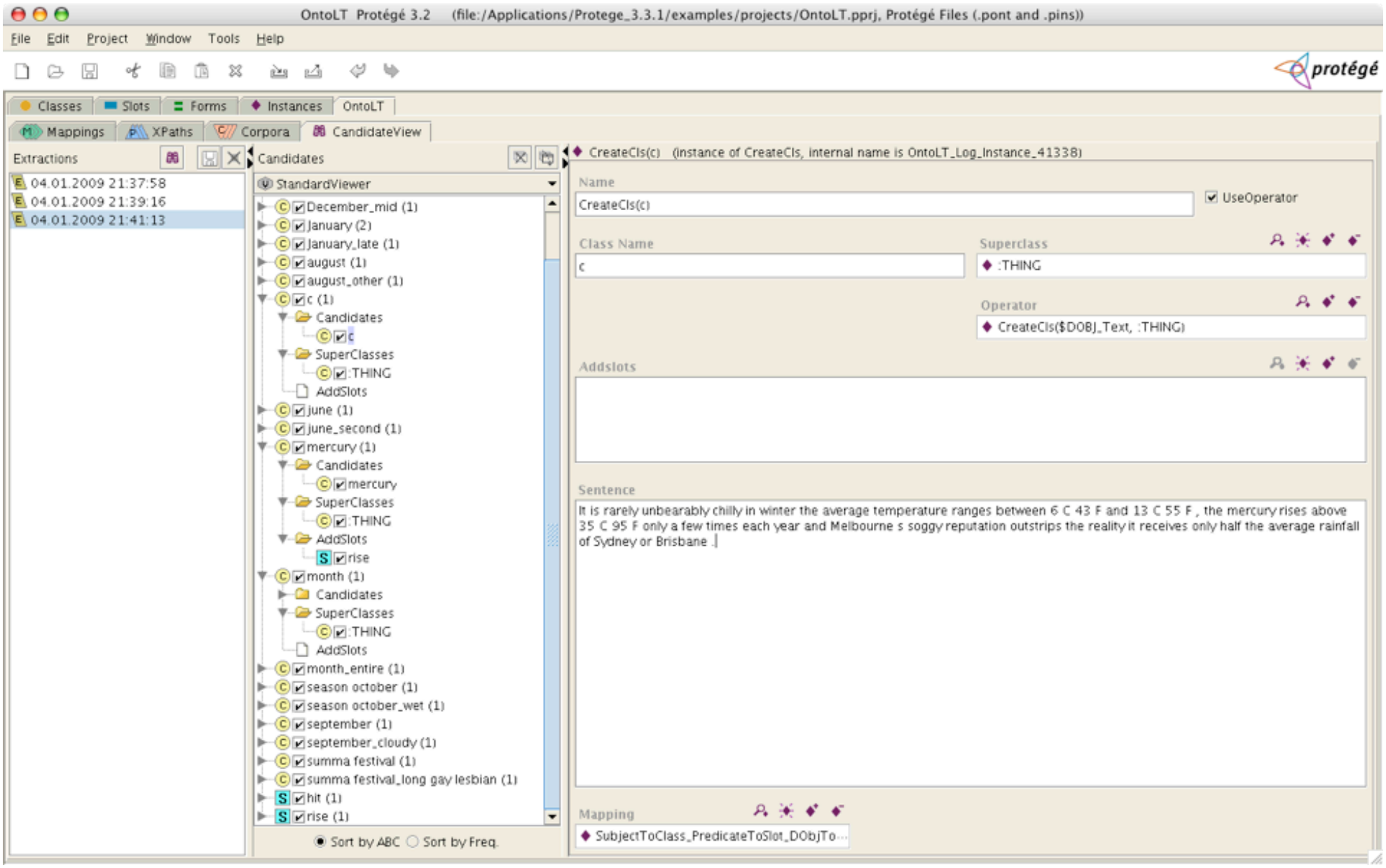
# OntoLT

- Die Precondition Language verfügt zudem über boolsche Abfragen zur Textanalyse:
- containsPath
- HasValue
- HasConcept
- AND, OR, NOT, EQUAL
- ID

# OntoLT

- Über Operationen können neue Klassen, Slots und Instanzen erzeugt werden.
- Um nur relevante Information zu extrahieren wird ein Statistisches Präprozessing auf Basis von “chi-square” verwendet.

# OntoLT


OntoLT Protégé 3.2 (file:/Applications/Protege_3.3.1/examples/projects/OntoLT.pprj, Protégé Files (.pont and .pins))
File Edit Project Window Tools Help
protégé
Classes Slots Forms Instances OntoLT
Mappings XPaths Corpora CandidateView
Extractions
04.01.2009 21:37:58
04.01.2009 21:39:16
04.01.2009 21:41:13
Candidates
StandardViewer
December_mid (1)
January (2)
January_late (1)
august (1)
august_other (1)
c (1)
Candidates
c
SuperClasses
:THING
AddSlots
june (1)
june_second (1)
mercury (1)
Candidates
mercury
SuperClasses
:THING
AddSlots
rise
month (1)
Candidates
SuperClasses
:THING
AddSlots
month_entire (1)
season october (1)
season october_wet (1)
september (1)
september_cloudy (1)
summa festival (1)
summa festival_long gay lesbian (1)
hit (1)
rise (1)
Sort by ABC Sort by Freq.
CreateCls(c) (instance of CreateCls, internal name is OntoLT_Log_Instance_41338)
Name
CreateCls(c)
UseOperator
Class Name
c
Superclass
:THING
Operator
CreateCls($DOBJ_Text, :THING)
Addslots
Sentence
It is rarely unbearably chilly in winter the average temperature ranges between 6 C 43 F and 13 C 55 F , the mercury rises above 35 C 95 F only a few times each year and Melbourne s soggy reputation outstrips the reality it receives only half the average rainfall of Sydney or Brisbane .
Mapping
SubjectToClass_PredicateToSlot_DObjTo

# OntoLT

- http://olp.dfki.de/OntoLT/OntoLT.htm
- Da OntoLT einen Annotated Corpus benötigt, ist es nicht immer verwendbar.
- Bietet wegen der Mapping Regeln auf XPATH sehr viele manuelle Bearbeitungsmöglichkeiten
- Fällt daher in die Kategorie der semiautomatischen Ontologieerstellung.

# OntoLearn

- OntoLearn ist eine Sammlung von Tools zur Ontologieerstellung.
- Zur Zeit sind diese Tool online zum Testen unter http://lcl2.uniroma1.it/home.jsp verfügbar.
- Von besonderem Interesse sind dabei:
  - TermExtractor
  - Structural Semantic Interconnections (SSI)

# TermExtractor

- Extrahiert Terme aus Dokumenten.
- Als Corpus kann fast jedes Eingabeformat dienen (txt, word, pdf, etc)
- Verwendet zwei Entropy-basierte Algorithmen:
  - Domain Relevance
  - Domain Consensus

# Structural Semantic Interconnections

- Structural Semantic Interconnections ist eine Art Struktur Erkennungs Algorithmus der auf Graphen basiert.
- Benutzt WordNet Datenbank.
- wird noch genauer erläutert.

# OntoLearn

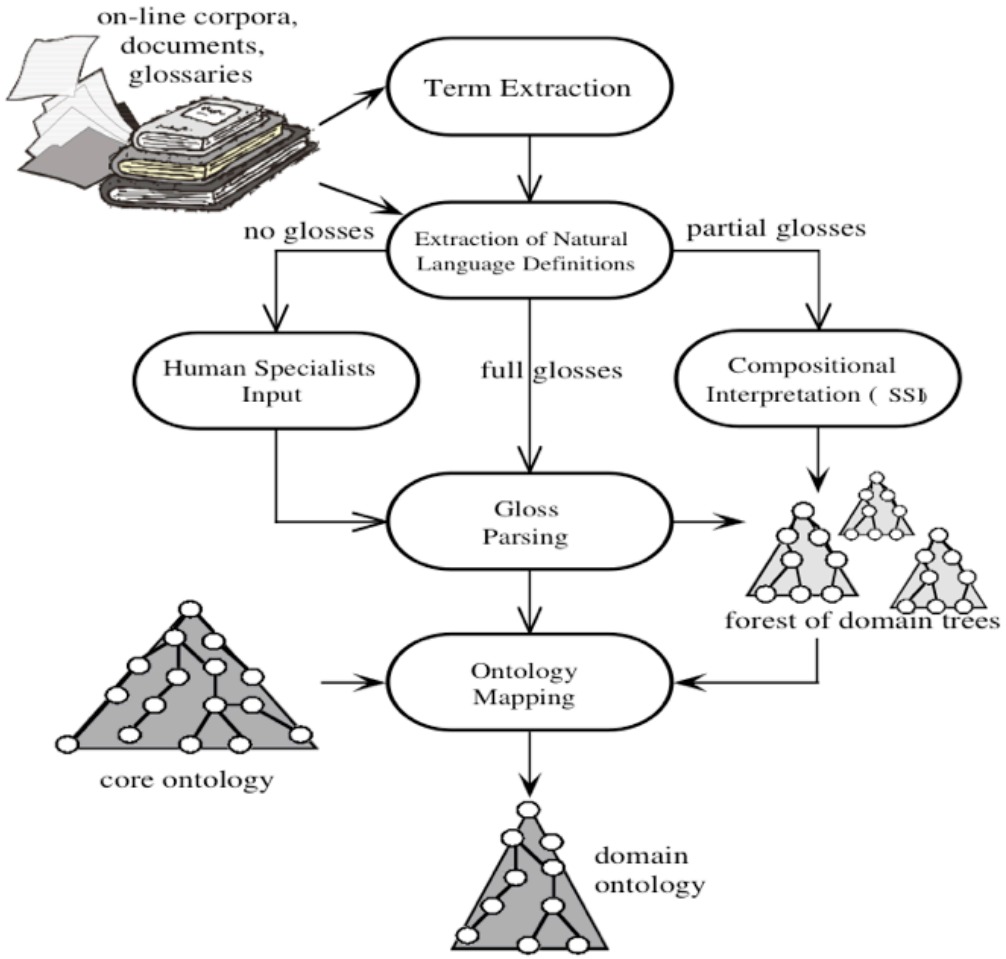



**Figure 1.** An outline of the ontology learning phases in the OntoLearn system.

| | Text2Onto | OntoLT | OntoLearn |
|---|---|---|---|
| Art | PlugIn für NeOn /Application | PlugIn für Protege | Webtool |
| Eingabeformat | Text, XML, HTML, PDF<br>Text wird bevorzugt | Annotated XML | Text, XML, HTML, PDF usw<br>und direkte Texteinfabe |
| Verwendete Methoden | POM, Data-driven Change Discovery, Natural Language Processing | (Semi-Automatic Generation of )Mapping Rules, Precondition Language, Operators, Statistical Preprocessing (chi-square) | SSI, Domain Relevance, Domain Consensus |
| Verwendete Datenbanken / dritt Programme | Gate (Jape), WordNet | - | WordNet |
| Ausgabeformat | OWL, RDF, F-Logic | Ausgabe über Protege, also OWL und RDF | XML, direkt online |

# Methoden

- POM (Probabilistic Ontology Model)
- SSI (Structural Semantic Interconnections)

# POM

- Ist das Datenmodel für Text2Onto
- Enthält eine Reihe von “Primitves”, die es unabhängig von einer bestimmten Ontologie machen.
- Diese “ Primitves” werden vom MPL deklarativ definiert.
- Dadurch kann das Model leicht erweitert werden und es kann leicht in andere Sprachen transformiert werden.

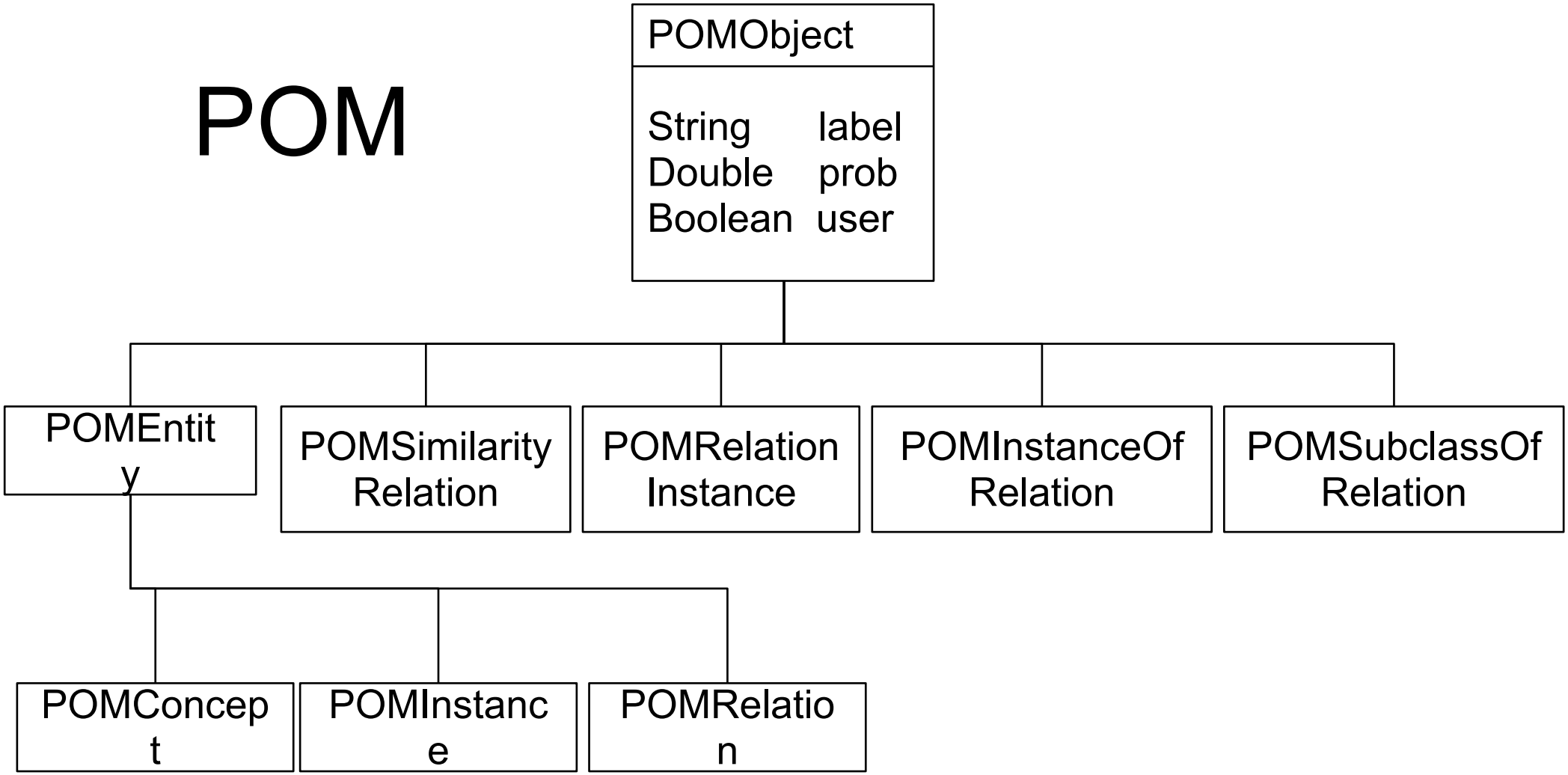
# POM

- Die vorhandenen “primitives” sind:
- concepts (CLASS)
- concept inheritance (SUBCLASS-OF)
- concept instantiation (INSTANCE-OF)
- properties / relations (RELATIONS)
- domain and range restrictions (DOMAIN / RANGE)
- mereological relations
- equivalence

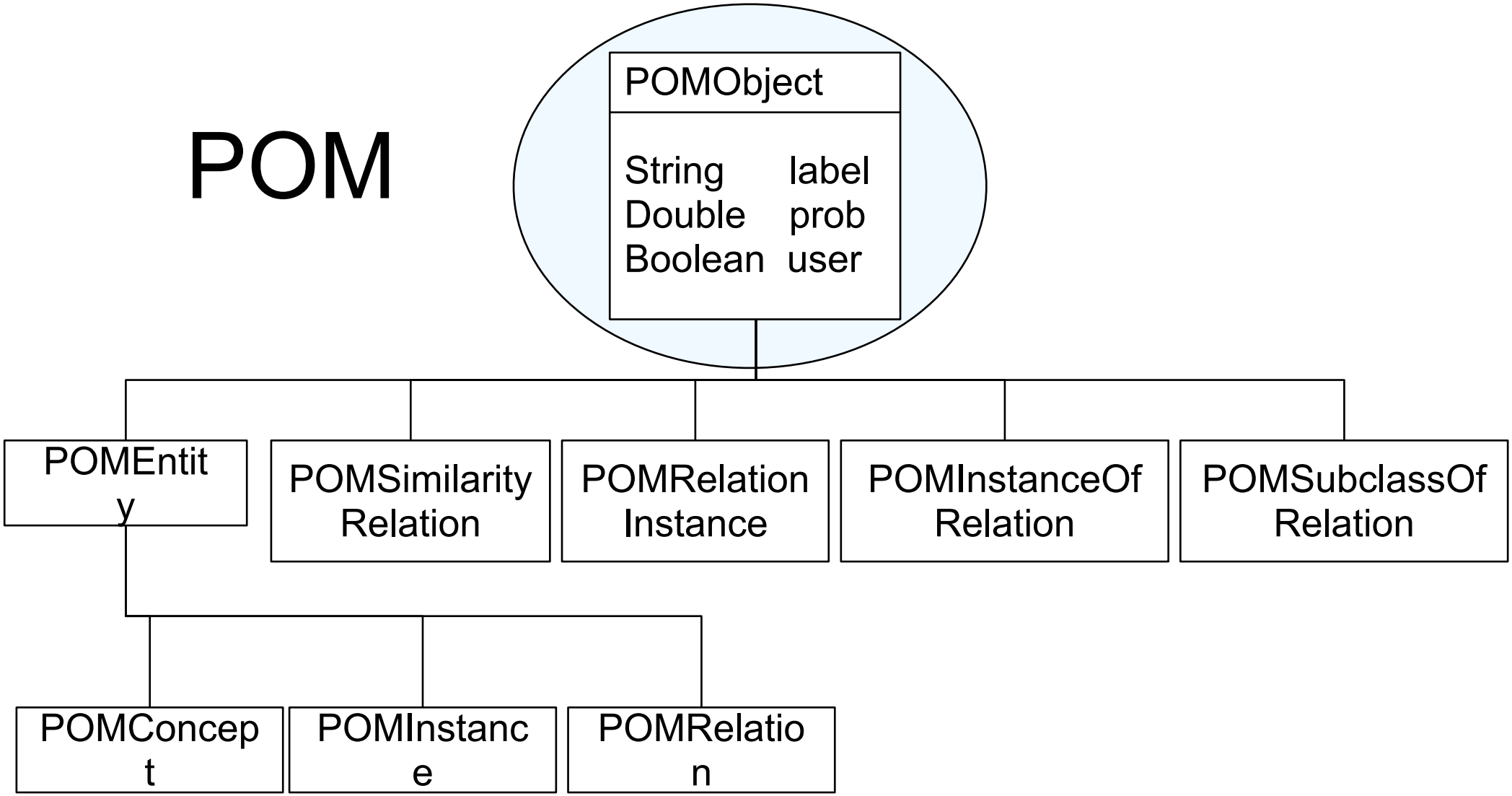
# POM

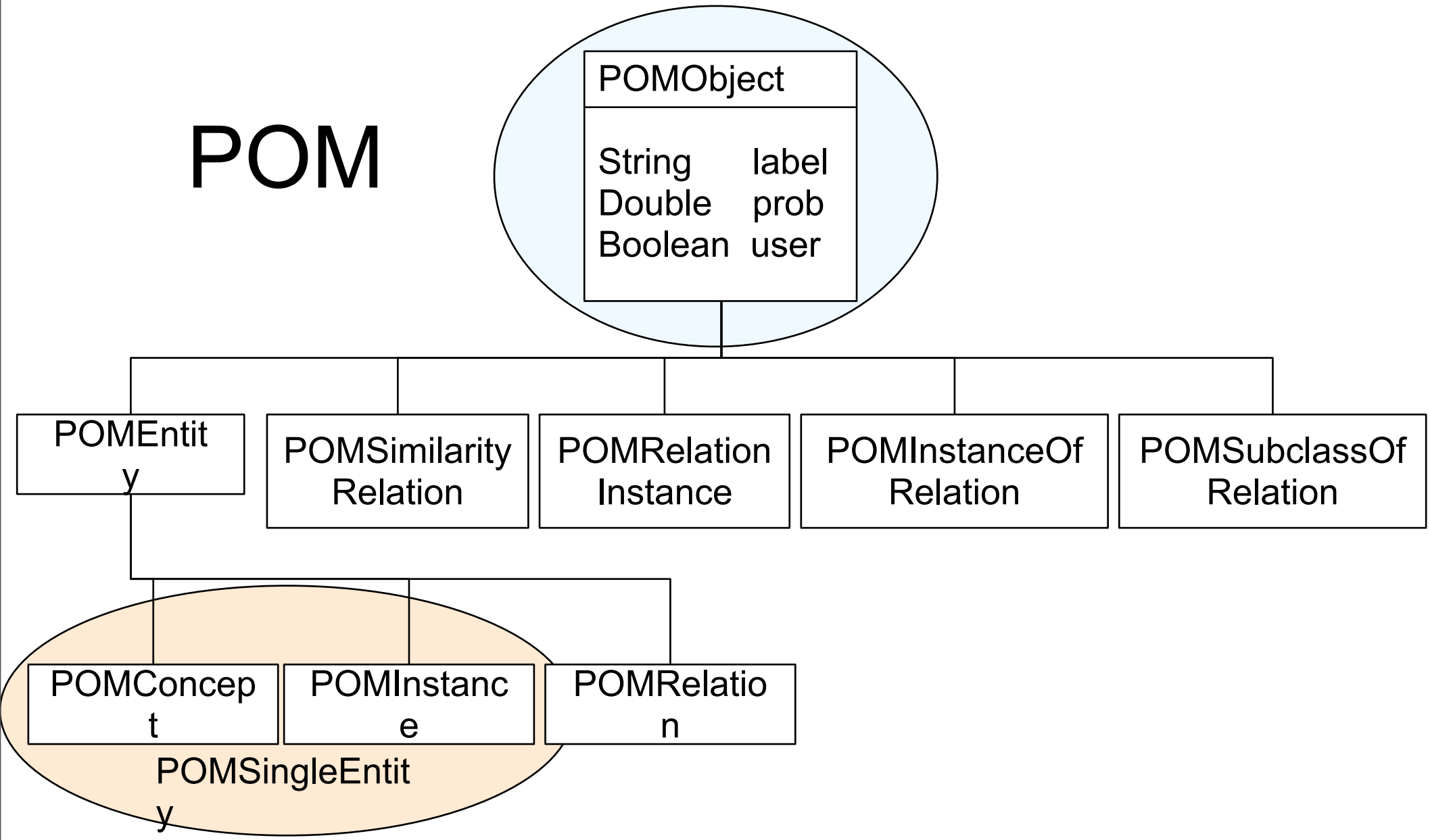
- Zu diesen “primitives” sind die entsprechenden Algorithmen zugeordnet.
- Jeder Algorithmus speichert einen Wahrscheinlichkeitswert, wie sicher er sich mit seinem Ergebnis ist.
- Zudem speichert das POM Änderungen ab, wenn sich die Ontologie oder der zugrundeliegende Corpus ändern.
- Über ontology writers kann das POM in gängige Ontologie Sprachen, wie RDFS, OWL oder F-Logic überführt werden.

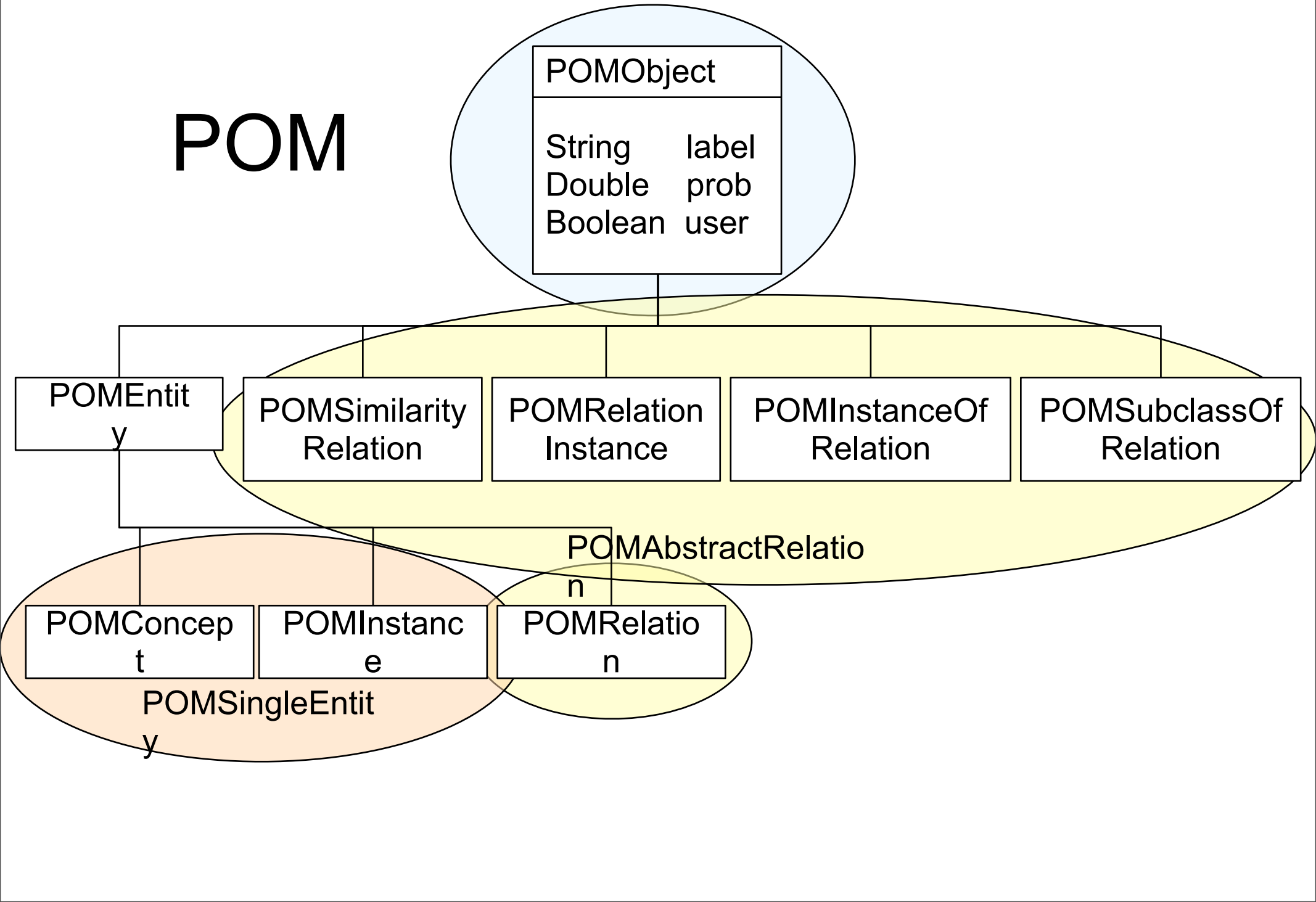
# POM

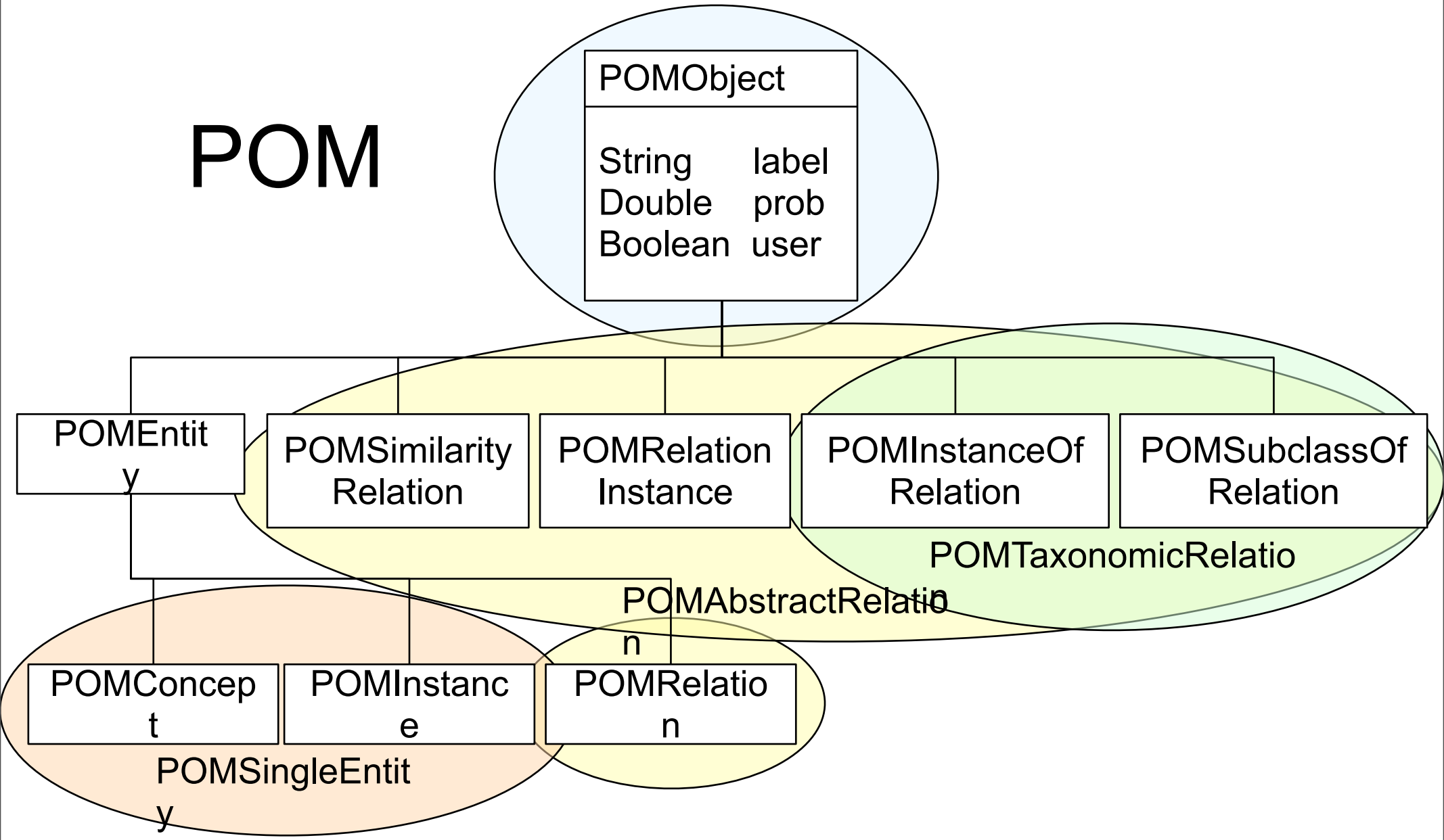

POMObject
String label
Double prob
Boolean user
POMEntity
POMSimilarityRelation
POMRelationInstance
POMInstanceOfRelation
POMSubclassOfRelation
POMConcept
POMInstance
POMRelation

# POM


POMObject
String label
Double prob
Boolean user
POMEntity
POMSimilarityRelation
POMRelationInstance
POMInstanceOfRelation
POMSubclassOfRelation
POMConcept
POMInstance
POMRelation


POM
POMObject
String label
Double prob
Boolean user
POMEntity
POMSimilarityRelation
POMRelationInstance
POMInstanceOfRelation
POMSubclassOfRelation
POMConcept
POMInstance
POMRelation
POMSingleEntity


POM
POMObject
String label
Double prob
Boolean user
POMEntity
POMSimilarityRelation
POMRelationInstance
POMInstanceOfRelation
POMSubclassOfRelation
POMAbstractRelation
POMConcept
POMInstance
POMRelation
POMSingleEntity


POM
POMObject
String label
Double prob
Boolean user
POMEntity
POMSimilarityRelation
POMRelationInstance
POMInstanceOfRelation
POMSubclassOfRelation
POMTaxonomicRelation
POMAbstractRelation
POMConcept
POMInstance
POMRelation
POMSingleEntity

# POM

- Ein POMConcept stellt ein Konzept dar.
- Eine POMInstance definiert eine Instanz.
- Eine POMSimilarityRelation definiert eine Ähnlichkeitsrelation zwischen zwei Instanzen vom Typ POMObject
- Eine POMSubclassOfRelation definiert eine taxonomische Relation zwischen zwei Instanzen der Klasse POMObject
- Eine POMInstanceOfRelation ordnet einem Konzept eine Instanz zu.
- Eine POMConceptualRelation definiert eine beliebige Relation zwischen zwei Instanzen vom typ POMObject, z. B. part-of(motor, car) oder love(man, woman).
- Eine POMConceptualRelationInstance stellt eine konkrete Ausprägung einer POMConceptualRelation dar, z. B. love(Peter, Mary).

# SSI Algorithmus

- Wird im OntoLearn Konzept verwendet
- Ist eine Art structural pattern recognition Algorithmus.
- Er versucht als Beziehungen zwischen Worten herzustellen und diese in Graphform aufzuzeigen.
- Ist Wissensbasiert, als Grundlage dient die WordNet Datenbank.

# SSI

Schritt 2 ist die Anwendung des SSI Algorithmus.

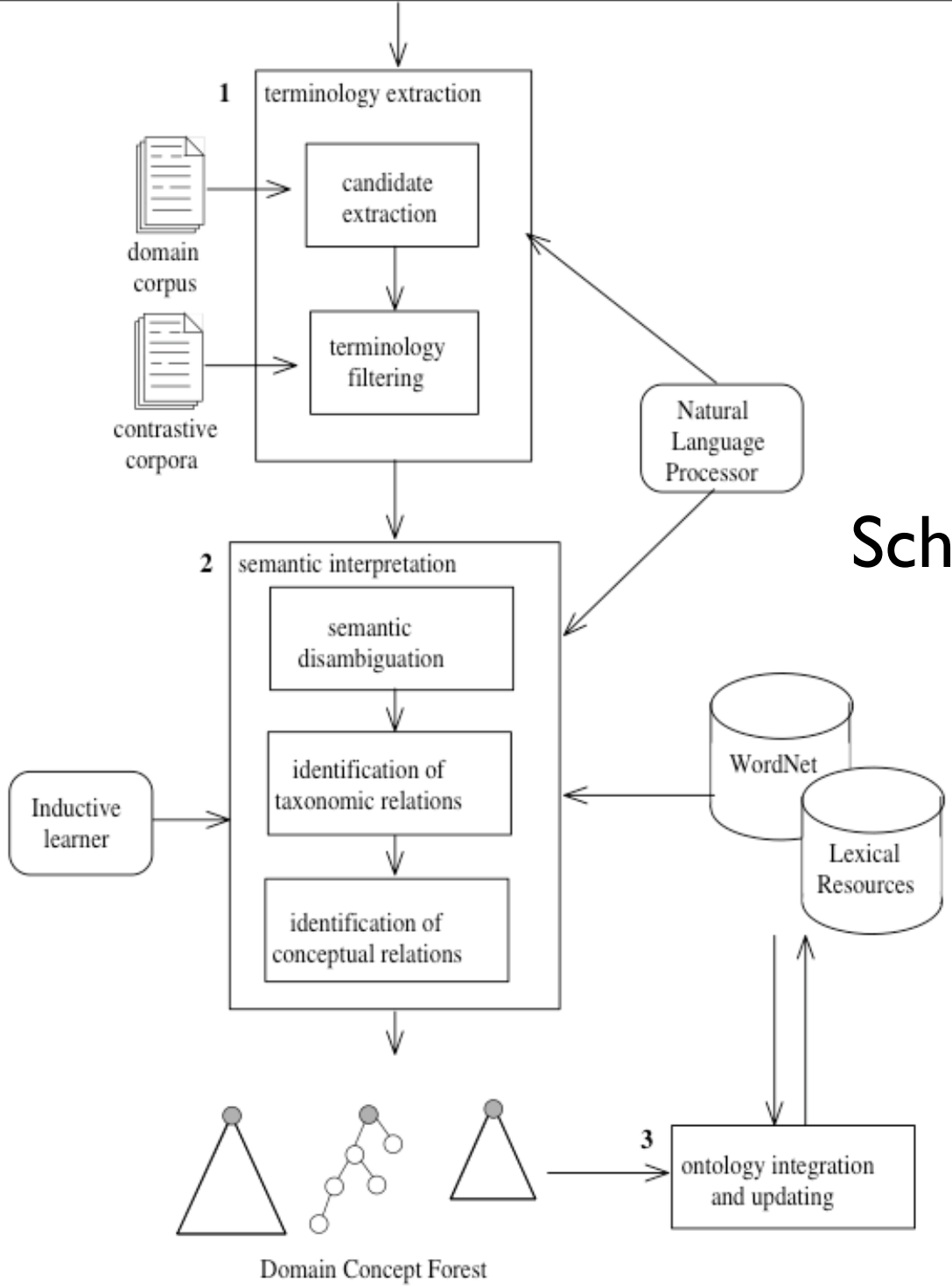



**Figure 3**
The architecture of OntoLearn.

# SSI Beispiel

Als Termmenge T wird [service, train, ferry, car, boat, car-ferry, bus, coach, transport, public transport, taxi, express, customer] verwendet.

1. Aufteilung zwischen eindeutigen und mehrdeutigen Worten:

I = [taxi#1, car ferry#1, public transport#1, customer#1 ]
P ={service, train, ferry, car, boat, bus, coach, transport, express}

# SSI Beispiel

## 2. Anwenden der Regeln

- {taxi} kind-of→ {car, auto}(hyper)
- {taxi} kind-of→ {car, auto} kind-of→ {motor vehicle,automotive vehicle} kind-of→ {vehicle} ←gloss {bus, autobus, coach}(hyper + gloss)
- {taxi} kind-of→ {car, auto} kind-of→ {motor vehicle,automotive vehicle} kind-of→ {vehicle} ←gloss {ferry, ferryboat}(hyper + gloss)
- {bus, autobus, coach} kind-of→ {public transport}(hyper) {car ferry} kind-of→ {ferry, ferryboat}(hyper)
- {customer, client} topic→ {service}(topic)
- {service} ←gloss {person, someone} has-kind→{consumer} has-kind→ {customer, client}(gloss + hypo)

# SSI Beispiel

2. Anwendung der Regeln (Fortsetzung)

- {train, railroad train} kind-of→ {public transport}(hyper)
- {express, expressbus} kind-of→ {bus, autobus, coach} kind-of→ {public transport}(hyper)
- {conveyance, transport} has-kind→ {public transport} (hypo)

Daraus folgt:

- I = [taxi#1, car ferry#1, public transport#1, customer#1, car#1, ferry#1, bus#1, coach#5, train#1, express#2, transport#1, service#1]
- P = {boat}

# SSI Beispiel

3. Schritt

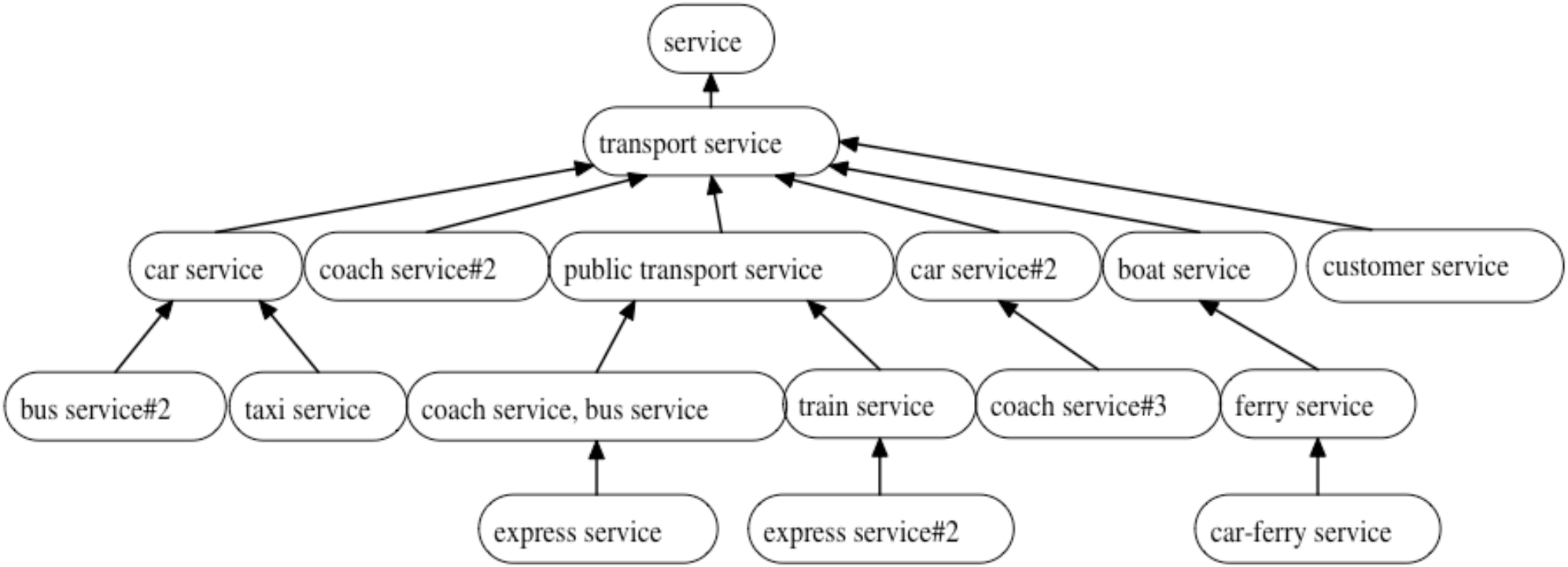
- {boat} has-kind→ {ferry, ferryboat}(hypo)
- I = [taxi#1, car ferry#1, public transport#1, customer#1, car#1, ferry#1, bus#1, coach#5, train#1, express#2, boat#1, transport#1, service#1 ]

- P = leer.

- P ist leer, der Algorithmus stoppt.

# SSI Beispiel

- Daraus lässt sich dann ein Konzept Baum erzeugen



# Zusammenfassung

- Es gibt verschiedene Tools, mit unterschiedlichen Ansätzen und Ausprägungen.
- Die Tools sind noch in der Entwicklung.
- Das Interesse an (semi-)automatischer Ontologieerzeugung hat erst kürzlich wieder zugenommen.
- Text2Onto ist am Umfangreichtsten, OntoLearn am zugänglichsten.

# Zusammenfassung

- Die Algorithmen kommen aus dem Bereich der Texterkennung und natürliche Sprachverarbeitung.
- Es werden bereits bestehende Frameworks und Datenbanken aus diesem Bereich verwendet.
- Eine vollständig automatische Ontologieerstellung ist noch nicht möglich.