TRIPLE - eine regelbasierte Sprache fuer die Verarbeitung von Metadaten mit heterogener Semantik

Stefan Decker
Information Sciences Institute
University of Southern California

Gliederung

- Semantic Web Uebersicht
  - Vision, Herausforderungen, Gruende
- TRIPLE: Eine Regelsprache fuer RDF
  - Verarbeitung heterogener Semantiken

Semantic Web

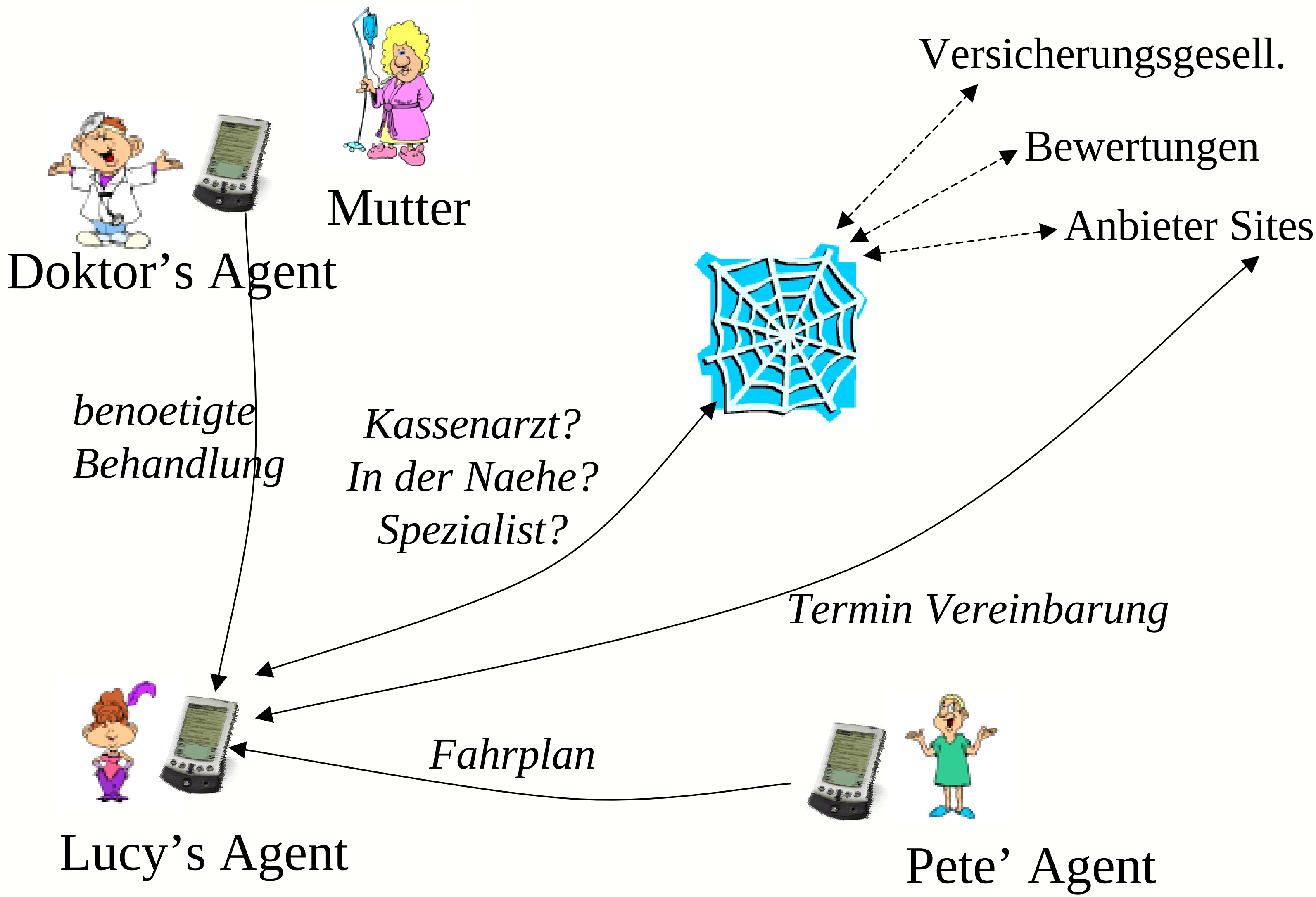
- Begriff gepraegt von Tim Berners-Lee (1997)

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

– *T. Berners-Lee, J. Hendler, O. Lassila,* *"The Semantic Web"*, Scientific American, May 2001

Doctor’s appointment

“The Semantic Web”, Scientific American, May 2001



Benoetigte Technologien

- Explizite Ontologien
  - Benoetigt um fremde Daten zu verstehen
- Web Services
  - Benoetigt um Dienste miteinander zu verbinden

Technologische Herausforderungen

- Interoperabilitaet
  - Unvollstaendige, heterogene Daten
  - Unvollstaendige, schlecht definierte, evolvierende Services
- Verarbeitung natuerlicher Sprache, Data Mining
  - implizite Informationen explizit machen
- Mensch-Maschine Interaktion
  - Anfrage Interface, Visualizierung
- Skalierbarkeit

Soziale Herausforderungen

- Standardisierung ist schwer
  - DublinCore
- Oekonomische Anreize
  - “Chicken and egg” Problem
- Metadata schlechter Qualitaet
  - Ratings, Konkurrenz
- Komplxitaet: Entwickler und Benutzer

Initiativen

- Machinenlesbare Daten:
  - dmoz open directory project.org (manual erstelltes Web Verzeichnis)
  - MusicBrainz.org (Musik Daten)
  - RSS (RDF Site Summary)
  - Adobe (embedded metadata)
  - CC/PP (Composite Capability/Preference Profiles)
  - P3P (Platform for Privacy Preferences)

Initiativen

- Industie/Verwaltungs Vokabular Projekte
  - PapiNet.org: Vokabular fuer die Papier Industrie
  - BPMI.org: Vokabular fuer den Austausch von Geschaeftsprozessmodellen
  - XML-HR: Vokabular for Human Resources (HR) Informationen
  - DMTF (Distributed Management Task Force) (Vokabular fuer das Management von Firmen
  - Digital Government Projekte
  - ...
- Forschungsprojekte
  - Gen Ontology Working Group (genontology.org)
  - Environmental Sciences (SCEC, NASA)
  - MathNet
  - …

Wie schaffen wir die noetigen Technologien?

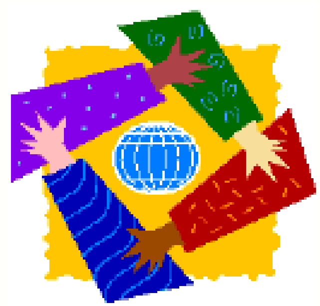
Forschungs-Communities

DL, AI, DB, …

Standardisierungs-organisationen

W3C, OMG, …

Nicht-Kommerzielle Organisationen



US, EC, Japan

Industrie

IBM, Nokia, HP, Microsoft(?),...

Business.semanticweb.org

Nicht-Kommerziell

- DARPA
  - “DARPA Agent Markup Language”
  - seit August 2000

www.daml.org

- NSF
  - Sponsored events (e.g., SWWS)
  - Further support in the loop

www.semanticweb.org/SWWS

- Europaeische Kommission
  - “Semantic Web Technologies”, FrameWork 6

www.ontoweb.org

- Japan
  - Interoperability Technology Association for Information Processing, Japan (INTAP)

www.net.intap.or.jp/INTAP/

KI: “Logik fuer das Web”

- Rules
- Agenten
- Interoperabilitaet
  - Praedikatenlogik
  - Ontologien, Modellierungsprimitive, Beschreibungslogiken
  - Logikprogrammierung
  - Problemloesungsmethoden
  - …

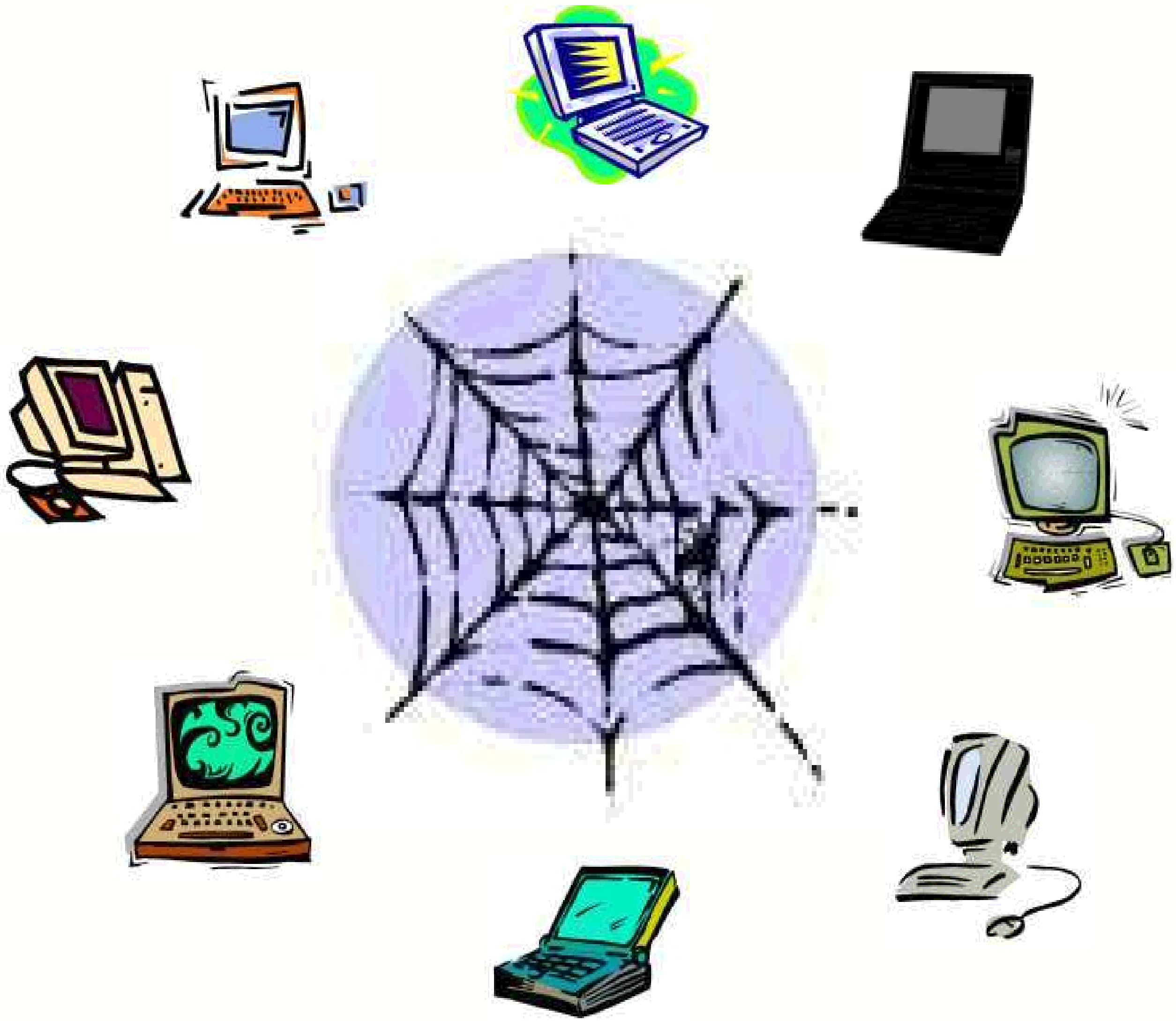
Verteilte Wissensbasen

DB: “Alles ist nur Syntax”

- Semi-strukturierte Daten
- Web Services
- Interoperabilitaet
  - Daten Integration (Local as View/ Global as View)
  - Mediation, Query rewriting
  - Model Management

Konglomerate von verteilten heterogenen (semi-strukturierten) Datenbanken

Unbekannte Kommunikationspartner



Heterogene Daten

- Viele Datenformate und Modellierungsprachen

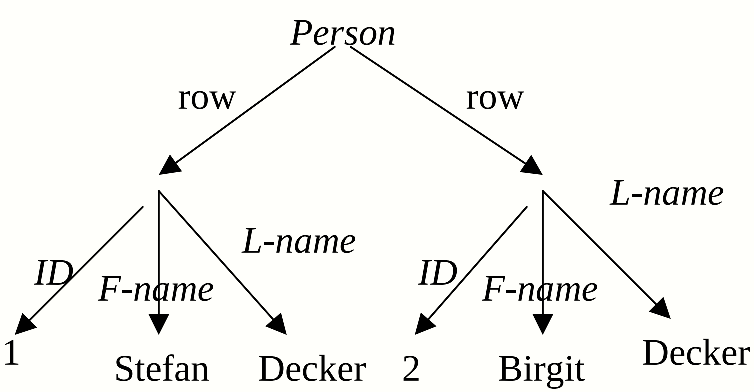
1. Schritt

- Verwende uniforme Basissyntax
  - Kleinster gemeinsame Basis: labeled Graphen (semi-strukturierte Daten) -> RDF

Relational Database

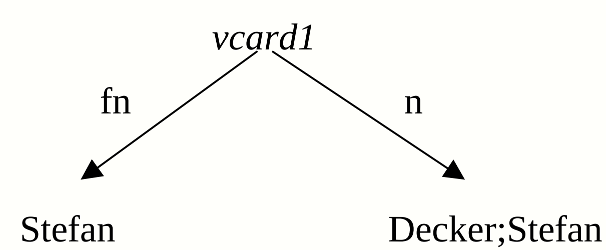
Person

ID	*F-name*	*L-name*
1	Stefan	Decker
2	Birgit	Decker



Structured Text (e.g., Vcard)

```
begin:          vcard
fn:             Stefan
n:              Decker;Stefan
end:            vcard
```



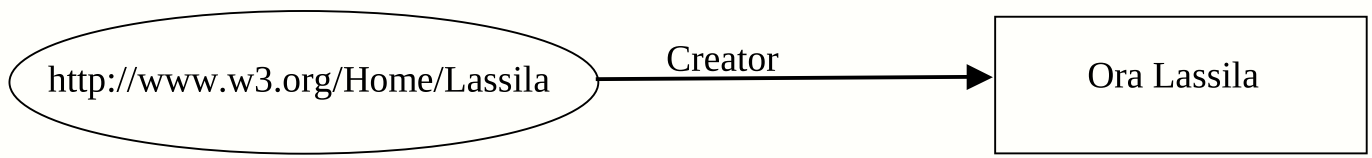
XML

- Verschachtelung, Hierarchy
- Ordnung (A gefolgt von B)
- Attribute (atomare Werte)
- Opaque Referenzen (IDREF)

Gut fuer die Serialisierung, nicht geeignet fuer die Modelierung von Semantik

Kodierung von Informationen in XML

“The Creator of the Resource “http://www.w3.org/Home/Lassila” is Ora Lassila



Endless encoding possibilities in XML:

```
<Creator>
    <uri>http://www.w3.org/Home/Lassila</uri>
    <name>Ora Lassila</name>
</Creator>
```

```
<Document uri=“http://www.w3.org/Home/Lassila”
    <Creator>Ora Lassila</Creator>
</Document>
```

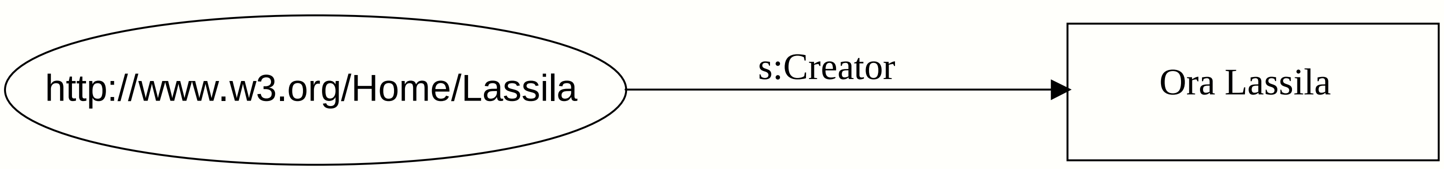
```
<Document uri=“http://www.w3.org/Home/Lassila” Creator=“Ora Lassila”/>
```

RDF (Resource Description Framework)

- RDF vereint verschiedene Interessengruppen:
  - Digital librarians, content-raters, privacy advocates, B2B industries, KI...
  - Signifikantes (aber geringeres als XML) industrielles Momentum, gefuehrt vom W3C
- RDF besteht aus zwei Teilen
  - RDF Model (eine Menge vom Tripeln)
  - RDF Syntax (verschiedene XML Serialisierungsvarianten)
- RDF Schema fuer die Definition von Vokabularen (einfache Ontologies) fuer RDF (und in RDF)

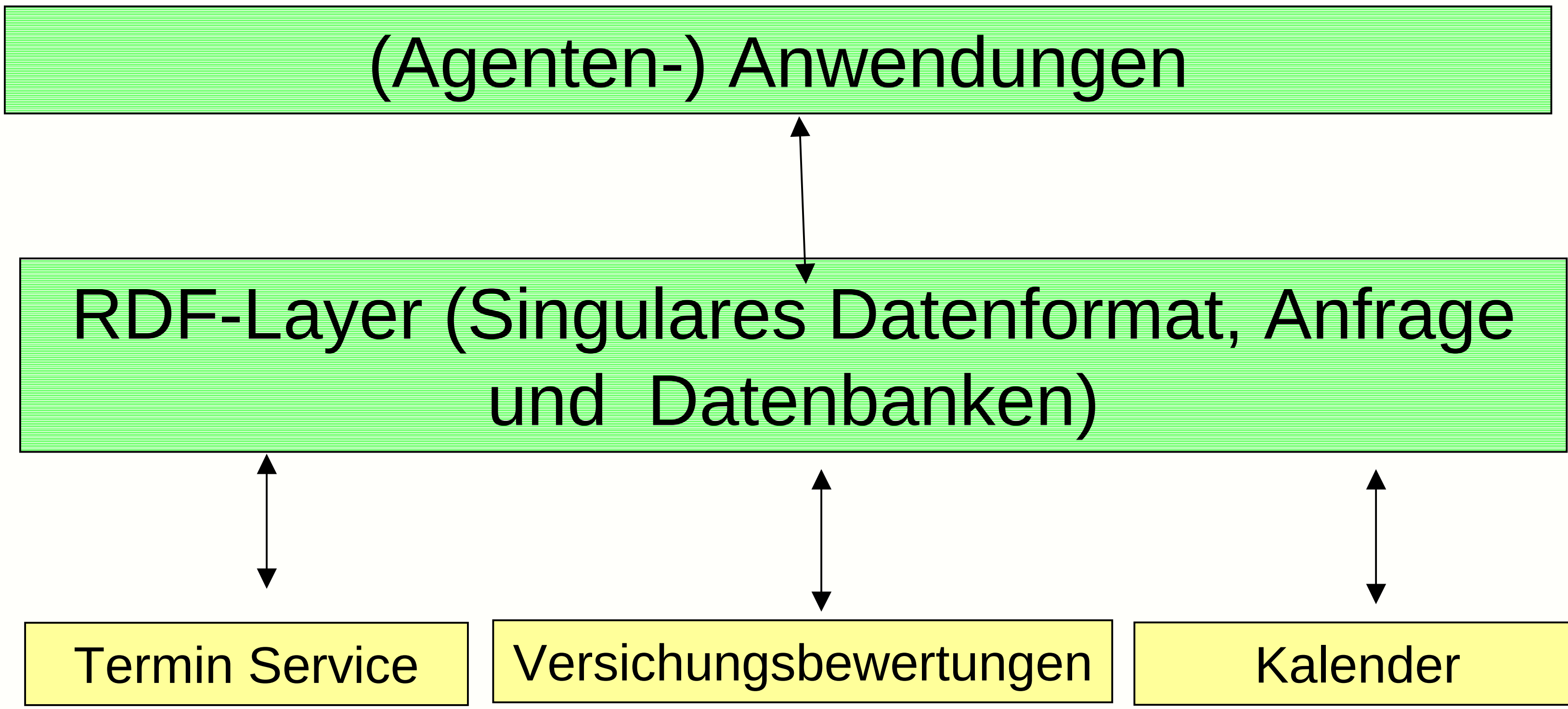
Ein Beispiel

- Beschreibung von Resourcen
  - URIs: globale OIDs, Literale
  - Binaere Relationen zwischen Objekten
  - Links (Relationen) sind first-class Objekte
  - Blank (anonyme) Knoten
- "Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila"
- Struktur
  - Resource (subject) http://www.w3.org/Home/Lassila
  - Property (predicate) http://www.schema.org/#Creator
  - Value (object) "Ora Lassila"



RDF

- Graph-based universal syntax



Semantik in einer globalen offenen Umgebung?

Schritt 2: Ontologien

- Was ist eine Ontologie?

  „An ontology is a specification of a conceptualization.“

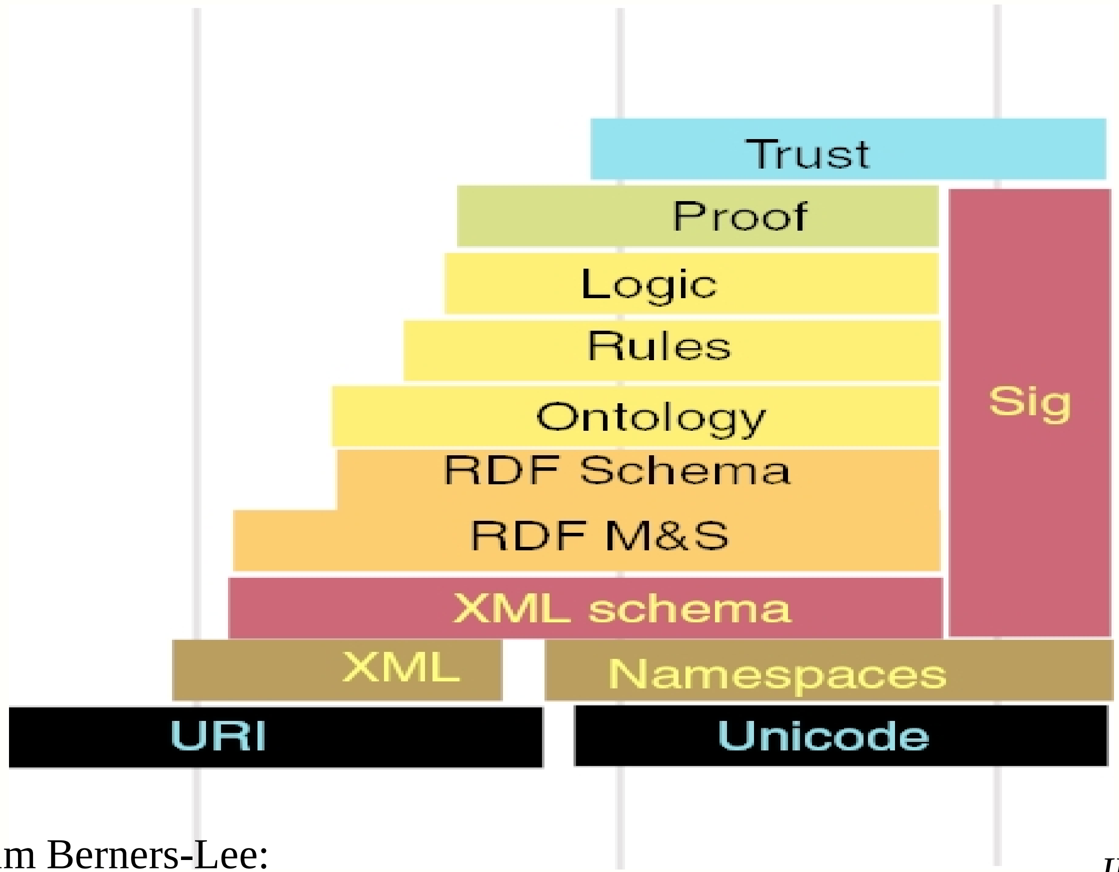
  Tom Gruber, 1993

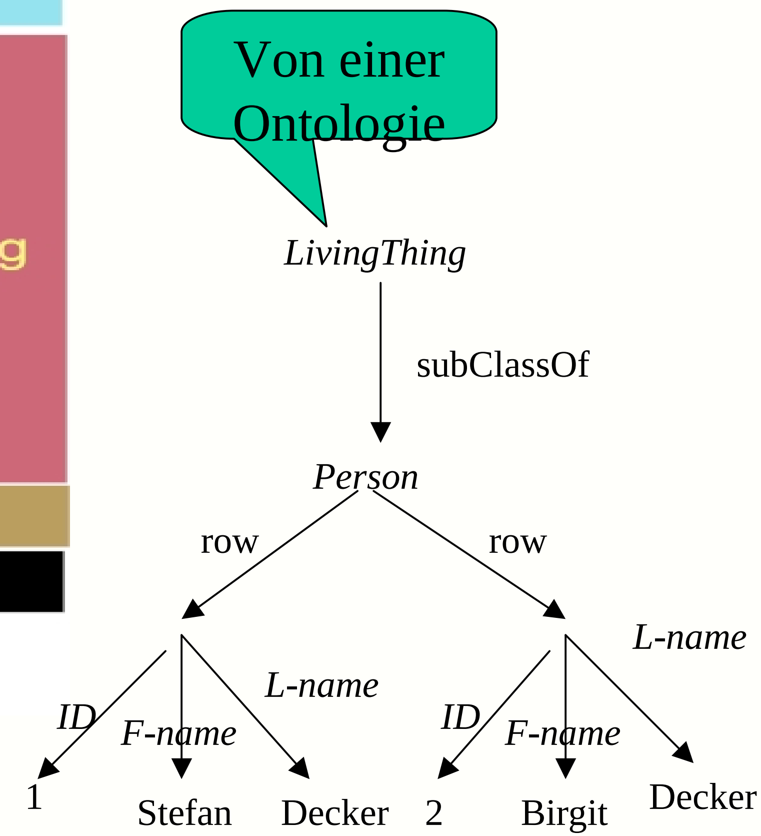
- Ontologien sind soziale Vertraege
  – Akzeptierte explizite Semantik
  – Nachvollziehbare Semantik
  – (Normalerweise) entwickelt in einem “community process”
- Ontologien benoetigen Wissenrepresentationtechniken
  – Is_a Hierarchy, teil-von, Attribute, Axiome, Defaults

RDF and Ontologies

- Idea: Definiere eine Ontologiesprache durch vordefinierte Knoten und Kanten
- The Ontologiesprache selber ist nur eine Ontoloie
- Ontologien werden benutzt um Daten zu annotieren

Schritt 3: Lagen ueber RDF





Tim Berners-Lee:
“Axioms, Architecture and Aspirations”
W3C all-working group plenary Meeting
28 February 2001

W3C Semantic Web Aktivitaet

Arbeitsgruppen

- RDF Core
- Web Ontology

Entwicklungsarbeiten

- Annotation (Annotea)
- Access control
- Calendaring
- Collaboration
- Logic
- Rules
- Workflows

RDF Core Working Group

- Resource Description Framework (RDF)
- Ziele
  - Verbessern des RDF Models und der XML Syntax basierend auf Feedback der Entwickler
  - Praezise Semantik fuer RDF und RDF Schema
  - Beziehungen mit der XML Familie (XML Schema)

Web Ontology Working Group

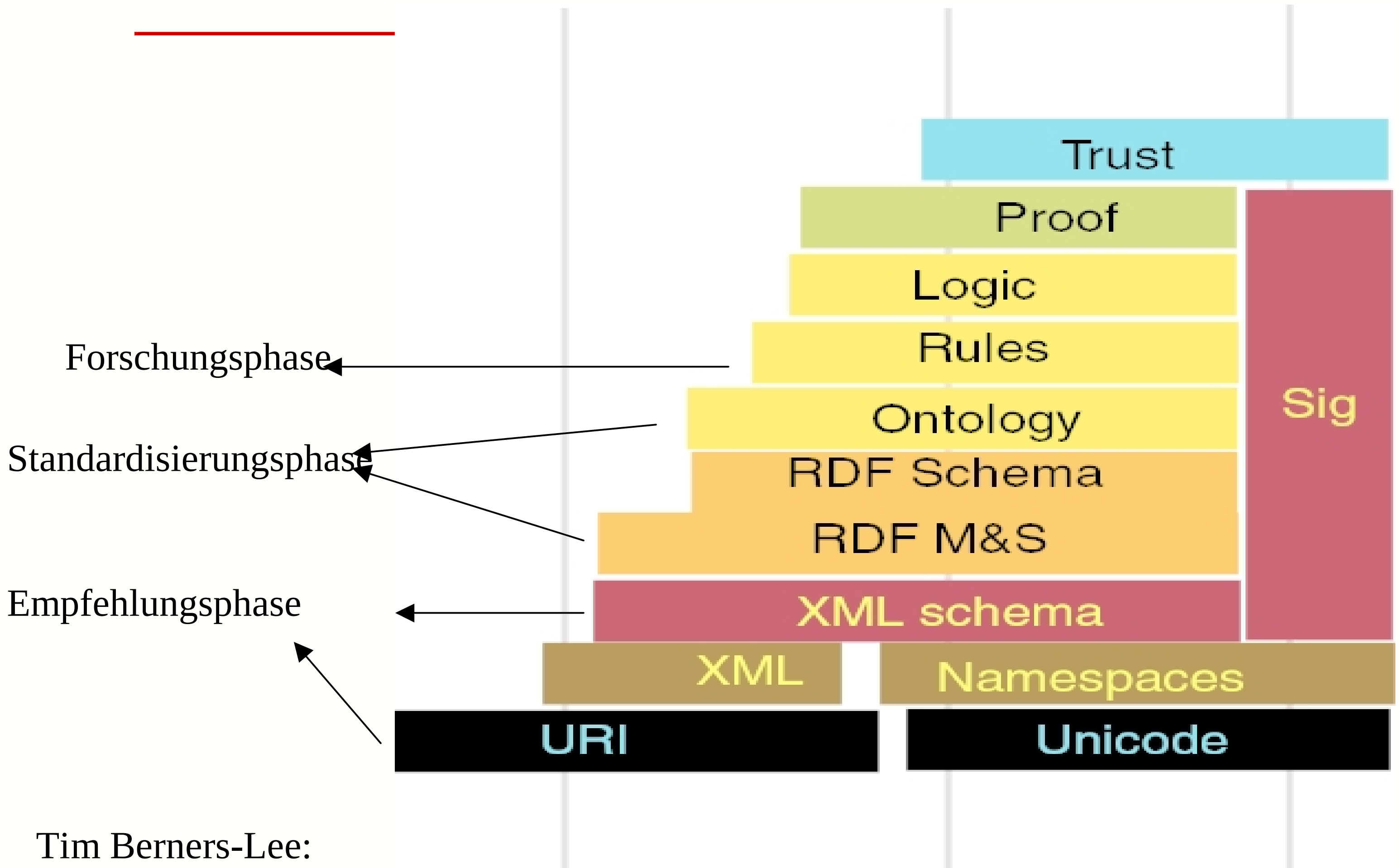
- Standardsprache fuer die Definition Ontologies
- Erweiterung von RDF Schema und DAML+OIL
  - Class Expressions (Intersection, Union, Complement)
  - XML Schema Datatypes
  - Aufzaehlungstypen
  - Property Restrictions
    - Kardinalitaet-constraints
    - Werteinschraenkungen

Events und Informationen

- 2st Intl. Semantic Web Conf., Sanibel Island, Florida, September 2003 (ISWC.semanticweb.org)
- Tracks and workshops
  - WWW Conf. 2003
  - SWDB auf der VLDB 2003
  - Knowledge Representation Conf. (KR) 2002
  - ECDL, ICEC, ECML, PKDD, …

www.semanticweb.org

Nochmal – die Torte



Tim Berners-Lee:
“Axioms, Architecture and Aspirations”
W3C all-working group plenary Meeting
28 February 2001

TRIPLE –
eine RDF Anfrage, Inferenz, und Transformations Sprache

Joint Work with Michael Sintek

Motivation: Why Rule Languages for the Web

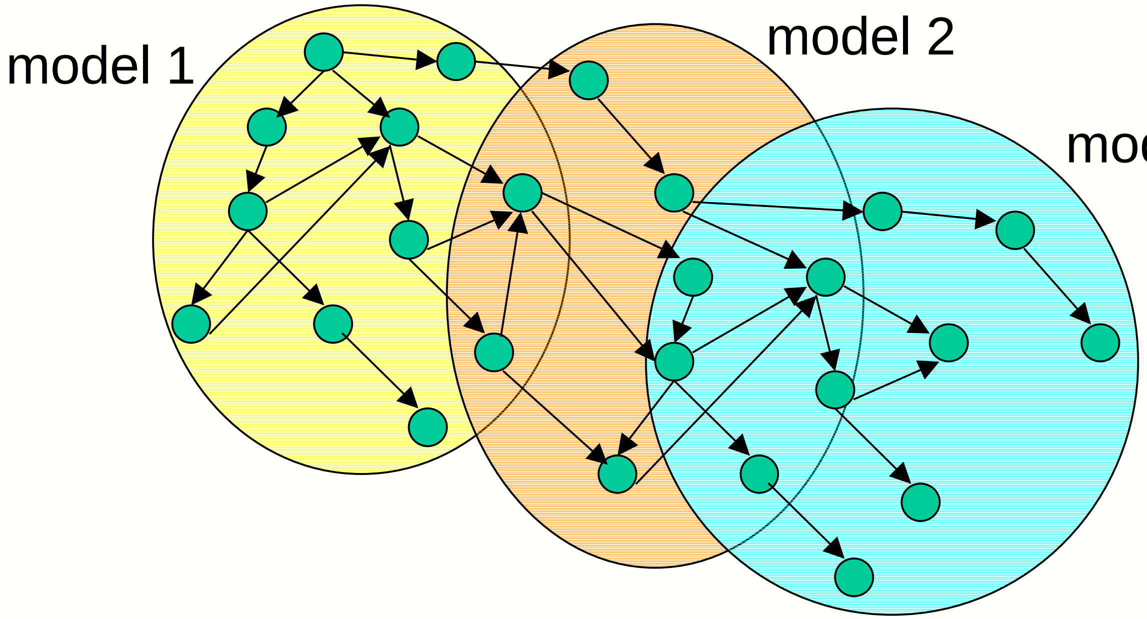
- Plethora of data available
  - Data needs to be adapted and combined
  - "*Time to Market*": Faster to write rules than code
  - Data Transformation and Integration
- Logic specification, not programming
  - Tabled evaluation/bottom-up evaluation
  - Semi-structured data
  - Multiple semantics (Relational Data, UML, ER, TopicMaps, DAML+OIL, XML-Schema, special purpose data models)
  - Distributed, heterogeneous sources

What's Wrong With Existing Approaches?

- Built-in *semantics* (e.g. SiLRI, RQL, DQL)
  - but: many RDF-based languages with different semantics (DAML+OIL, RDF Schema, UML/RDF, TopicMaps/RDF,…)
  - For each language a specialized query language ????

RDF Models

- No support for RDF *models*
  - one large heap of RDF data



TRIPLE:Language Overview

- Native support
  - for Resources & namespaces,
  - Abbreviations
  - Models (sets of RDF statements)
  - Reification
- Rules with expressive bodies (full FOL syntax)
- Inspired by F-Logic:
  - *subject*[*predicate*→*object*]   ("molecule")

Language Description I

- Namespace and resource abbreviations:
  - rdf := “http://www.w3.org/1999/02/22-rdf-syntax-ns#”.
  - isa := rdf:subClassOf.
- Statements, triples, molecules:
  - *subject*[*predicate*→*object*]
  - $subject[p_1 \rightarrow o_1; p_2 \rightarrow o_2; \ldots]$
  - $s_1[p_1 \rightarrow s_2[p_2 \rightarrow \mathrm{o}]\ ]$
- Models, model expressions, parameterized models:
  - $s[p \rightarrow o]@m$ — “triple <*s*,*p*,*o*> in model *m*”
  - $s[p \rightarrow o]@(m_1 \cap m_2)$ — model intersection, union, diff.
  - $s[p \rightarrow o]$@sf(m1, X, Y) — Skolem function

Language Description II

- Reification:
  - stefan[believes → <Ora[isAuthorOf→homepage]>]
- Logical formulae:
  - usual logical connectives and quantifiers: $\wedge \vee \neg \forall \exists \leftrightarrow \leftarrow \rightarrow$
  - all variables introduced via ∀ (or ∃)
- Clauses:
  - facts: $s[p_1 \rightarrow o_1;\ p_2 \rightarrow o_2;\ ...]$.
  - rules: $\forall X \;\; s_1[p_1 \rightarrow X] \;\leftarrow\; s_2[p_2 \rightarrow X] \wedge \ldots$.
- Model blocks:
  - *@model* { *clauses* }
  - ∀*Mdl* *@model*(*Mdl*) { *clauses* }

Example: Dublin Core

dc := “http://purl.org/dc/elements/1.0/”.
db := “http://www-db.stanford.edu/”.
....

namespace abbreviations

@db:documents {

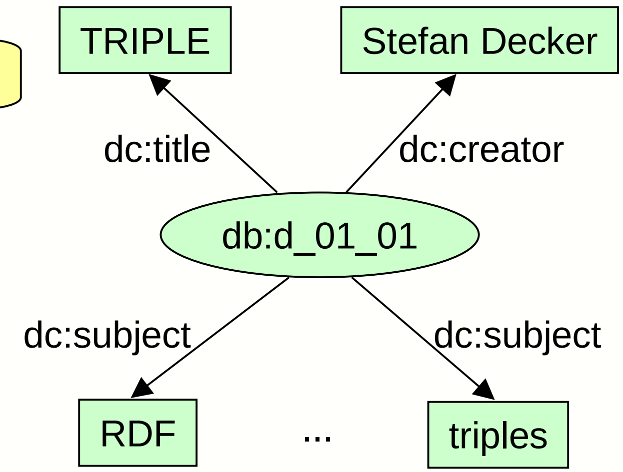
model block

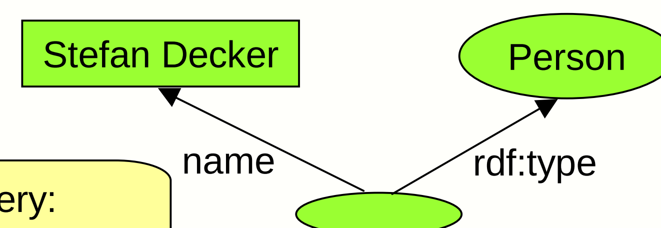
db:d_01_01 [
dc:title $\rightarrow$ TRIPLE;
dc:creator $\rightarrow$ “Stefan Decker”;
dc:subject $\rightarrow$ RDF;
dc:subject $\rightarrow$ triples; ...].

fact

$\forall$N p(N)[rdf:type $\rightarrow$ xyz:Person;
xyz:name $\rightarrow$ N] $\leftarrow$
$\exists$D D[dc:creator $\rightarrow$ N].

rule





query: “find all names”

$\forall$N $\leftarrow$ $\exists$P P[rdf:type $\rightarrow$ xyz:Person;
xyz:name $\rightarrow$ N]@db:documents.

$\Rightarrow$ N = “Michael Sintek”

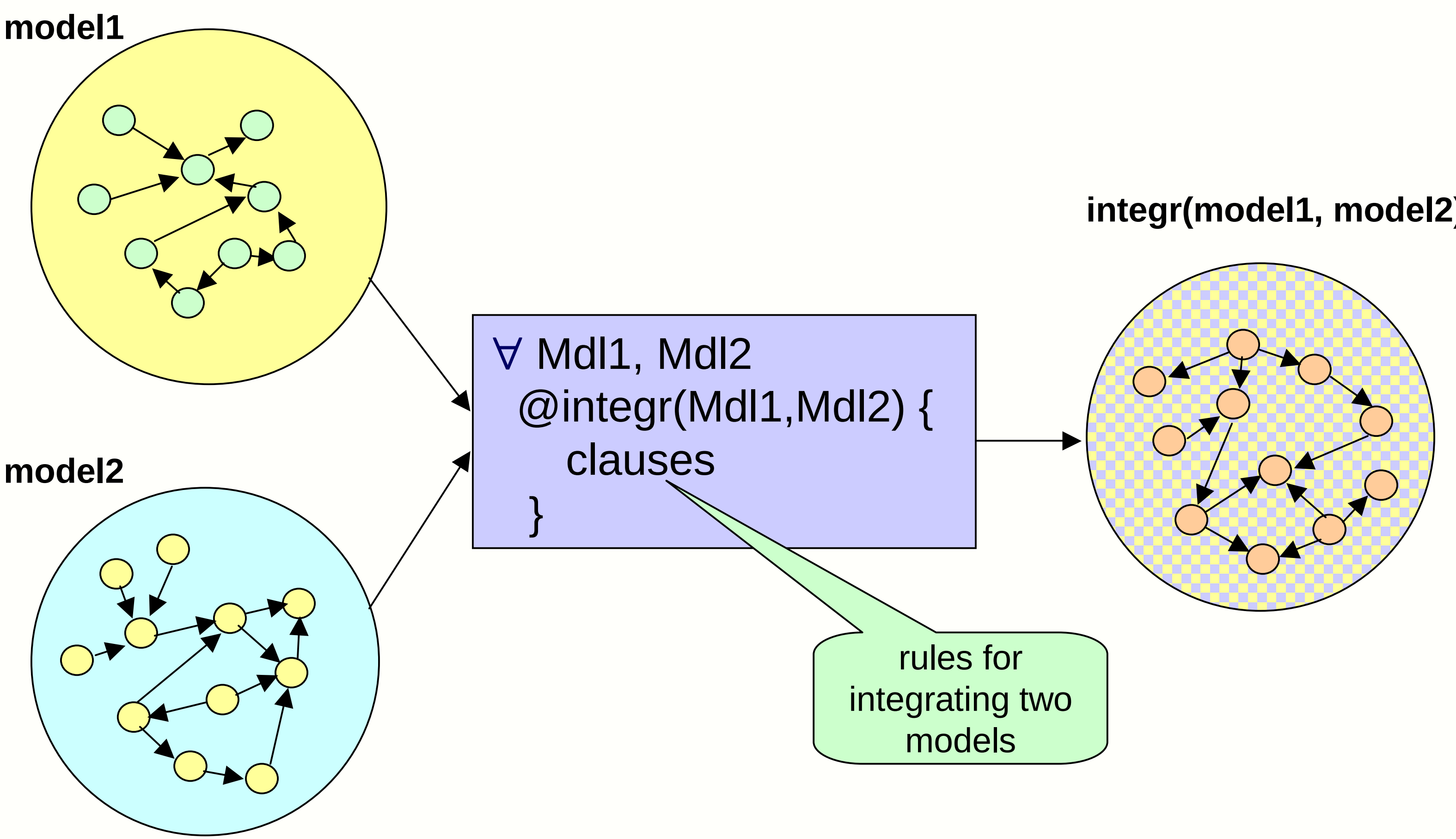
Parameterized Models

- General format:

$\forall P_1, \ldots, P_n$ $@model(P_1, \ldots, P_n)$ {
  $clauses[P_1, \ldots, P_n]$
}

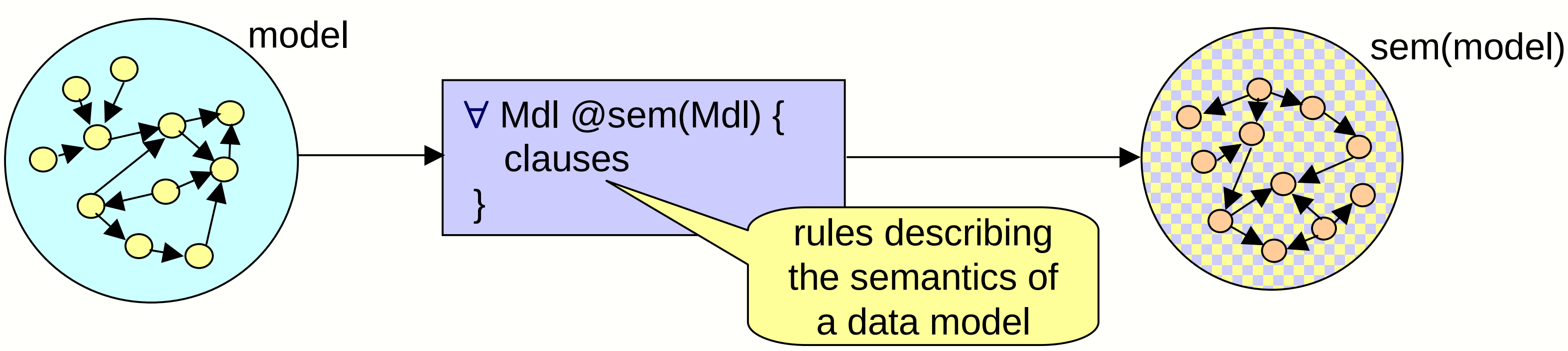
- Used for:
  - Data integration
  - Model transformation
  - Defining the *semantics* of languages layered on top of RDF ("semantic spaces")
  - Module system

Parameterized Models for Data Integration



Semantic Spaces: Specifying Semantics via Parameterized Models

- RDF Schema, UML (and other frame/OO systems): semantics can be *directly* defined in TRIPLE as a parameterized model
- OIL, DAML+OIL, OWL (i.e., expressive ontology languages, DL):
  requires interaction with *foreign* reasoning components (e.g., DL classifier)



Specification of RDF Schema Semantics

```
rdf := 'http://www.w3.org/...rdf-syntax-ns#'.
rdfs := 'http://www.w3.org/.../PR-rdf-schema-...#'.
type := rdf:type.
subPropertyOf := rdfs:subPropertyOf.
subClassOf := rdfs:subClassOf.

FORALL Mdl @rdfschema(Mdl) {

  FORALL O,P,V   O[P->V] <-
     O[P->V]@Mdl.

  FORALL O,V   O[subClassOf->V] <-
   EXISTS W   (O[subClassOf->W]
                  AND W[subClassOf->V]).

  …
}
```

namespace abbreviations

resource abbreviations

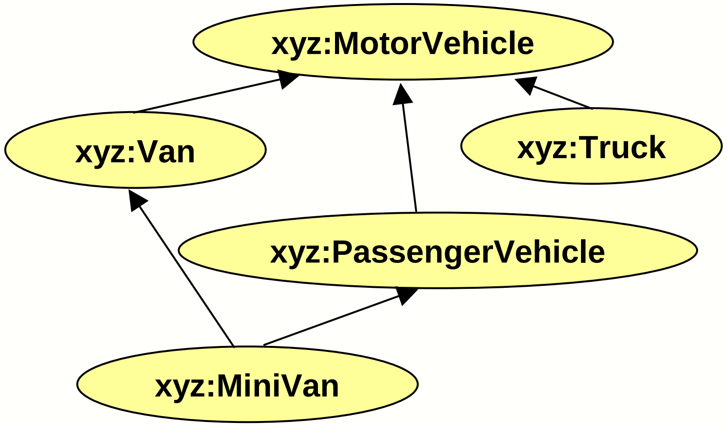
model block

“copy” triples from *Mdl*

Transitivity of subClassOf

Example: Cars Ontology

```
@cars {
  xyz:MotorVehicle[rdfs:subClassOf -> rdfs:Resource].
  xyz:PassengerVehicle[rdfs:subClassOf -> xyz:MotorVehicle].
  xyz:Truck[rdfs:subClassOf -> xyz:MotorVehicle].
  xyz:Van[rdfs:subClassOf -> xyz:MotorVehicle].
  xyz:MiniVan[
    rdfs:subClassOf -> xyz:Van;
    rdfs:subClassOf -> xyz:PassengerVehicle].
}
```




```
FORALL X <-
    X[rdfs:subClassOf -> xyz:MotorVehicle]@cars.
```

⇨

X = xyz:Van
X = xyz:Truck
X = xyz:PassengerVehicle

```
FORALL X <-
    X[rdfs:subClassOf -> xyz:MotorVehicle]@rdfschema(cars).
```

⇨

X = xyz:Van
X = xyz:Truck
X = xyz:PassengerVehicle
X = xyz:MiniVan

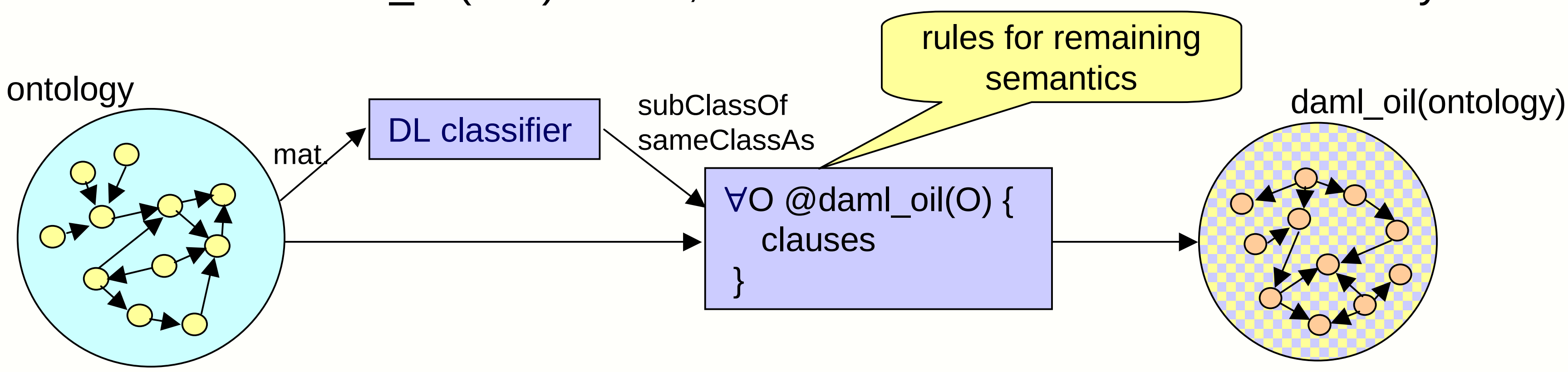
Specification of UML Semantics

```
rdf := 'http://www.w3.org/...rdf-syntax-ns#'.
uml := 'http://www.omg.org/uml/1.3/'.

FORALL Mdl @uml(Mdl) {

  FORALL O,P,V   O[P->V] <-
      O[P->V]@Mdl.

  FORALL X,Z   g(X,Z)[rdf:type->uml:Generalization;
                          uml:'Generalization.child'->X;
                          uml:'Generalization.parent'->Z]<-
         EXISTS Y,G1,G2
              G1[uml:'Generalization.child'->X; uml:'Generalization.parent'->Y]
                 AND
              G2[uml:'Generalization.child'->Y; uml:'Generalization.parent'->Z] .
…
}
```

Transitivity of Generalization

DAML+OIL Semantics

- daml_oil(Mdl) model realized by accessing a DL classifier (e.g., Racer or FaCT)
- access only allowed in rule bodies
- realization: Mdl is materialized and transformed into input for DL classifier; classifier is invoked; (direct) subClassOf and sameClassAs added to daml_oil(Mdl) model; rest handled via TRIPLE rules directly:

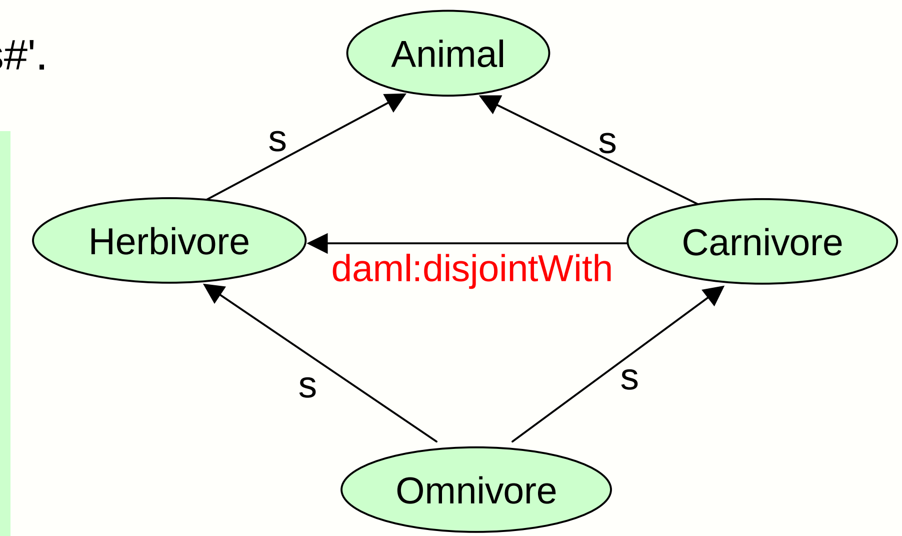


- results in hybrid rule language similar to Carin, but more pragmatic approach: powerful but incomplete

DAML+OIL Example

```
daml := 'http://www.daml.org/.../daml+oil#'.
animals := 'http://www.example.org/animals#'.
@animals:ontology {
  animals:Animal[rdf:type -> daml:Class].
  animals:Herbivore[rdf:type -> daml:Class;
    daml:subClassOf -> animals:Animal].
  animals:Carnivore[rdf:type -> daml:Class;
    daml:subClassOf -> animals:Animal;
    daml:disjointWith -> animals:Herbivore].
  animals:Omnivore[rdf:type -> daml:Class;
    daml:subClassOf -> animals:Herbivore;
    daml:subClassOf -> animals:Carnivore].
}
FORALL C
  <- C[daml:sameClassAs -> daml:Nothing]@daml_oil(animals:ontology).
```

find all unsatisfiable classes (will detect Omnivore)



s = daml:subClassOf

Realization: Mapping to Horn Logic

- First implementation by mapping
  to Horn Logic / XSB system (Prolog with tabled resolution)
- model theory for full logic completed
- Lloyd-Topor transformation for quantifiers etc.
- RDF-specific transformations given as rewrite rules:

$$
\begin{aligned}
A : N &\longrightarrow \mathrm{resource}(A, N) & (1)\\
O[P \rightarrow V] &\longrightarrow \mathrm{triple}(O, P, V) & (2)\\
S@M &\longrightarrow \mathrm{true}(S, M) \quad \text{for statements/atoms } S & (3)\\
<S> &\longrightarrow S \quad \text{for statements } S & (4)\\
O[P_1 \rightarrow V_1; P_2 \rightarrow V_2; \ldots]@M &\longrightarrow O[P_1 \rightarrow V_1]@M \ \wedge & (5)\\
& O[P_2 \rightarrow V_2]@M \ \wedge \ \ldots & \\
\mathrm{true}(S, M_1 \cap M_2) &\longrightarrow \mathrm{true}(S, M_1) \ \wedge \ \mathrm{true}(S, M_2) & (6)\\
\mathrm{true}(S, M_1 \cup M_2) &\longrightarrow \mathrm{true}(S, M_1) \ \vee \ \mathrm{true}(S, M_2) & (7)\\
\mathrm{true}(S, M_1 \backslash M_2) &\longrightarrow \mathrm{true}(S, M_1) \ \wedge \ \neg \ \mathrm{true}(S, M_2) & (8)\\
X := Y.\ S(X) &\longrightarrow \forall X \ \ (X = Y \ \wedge \ S(X)) & (9)\\
& \text{for clause sequences } S(X) &
\end{aligned}
$$

Conclusions

- Semantic Web is inevitable (maybe XML++ instead of RDF)
- Logic Programming provides techniques for data processing on the Web

- TRIPLE is an RDF-specific query, inference, and transformation language
- Allows specification of/access to multiple semantics
- Implementation (XSB-based) available at: http://TRIPLE.SemanticWeb.org
- Part of RuleML initiative: http://www.dfki.uni-kl.de/ruleml/
- Optimizations to come