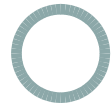


# Description Logics as Ontology Languages for the Semantic Web



**FGI 3 Seminar**

**Jan-Christoph Wolf**

# Inhalt

2

- Motivation
- Semantic Web
  - Knowledge Representation
  - Ontologien
- Beschreibungslogik als Ontologiesprache
- Beschreibungslogik SHIQ
  - Syntax und Semantik
  - Reasoning
  - Beschreibungsontologien in SHIQ
  - Beispiel
- DAML+OIL
- OWL

# Motivation

3

- WWW zum Suchen und Bereitstellen von Informationen
- Heutige Suchmaschinen liefern viele Antworten
- Davon sind viele unpassend
- Wichtige Dinge werden nicht gefunden
- Webinhalte sind entworfen für Menschen, nicht für Maschinen
- Mark-Up (HTML) liefert nur Renderinformationen für die textuelle und grafische Präsentation für den Menschen

# Eine Möglichkeit

4

- Parsing
- Seite kann ihr Layout ändern
- Kein zuverlässiger Weg die Semantik zu bestimmen

# Semantic Web

5

- **Ziele:**
  - Maschinenverständliche Webressourcen
  - Strukturierung der bedeutsamen Inhalte
  - Austausch von Informationen zwischen Software-Agenten untereinander und mit menschlichen Agenten
  - Kein neues Web, sondern Erweiterung des vorhandenen

# Semantic Web

6

- **Eigenschaften des Web:**
  - „anything can link to anything“
  - Verschiedene Informationsarten
  - Dezentral
  - keine komplette Konsistenz
  - exponentielles Wachstum
- **Semantisches Mark-Up notwendig**

# Knowledge Representation

7

- Zugang zu strukturierten Informationssammlungen und Schlussregeln
- Automatisches Folgern
- Traditionell: zentralisierte Systeme
- Größe steigt schnell an -> nicht mehr zu kontrollieren
- Dieselben Definitionen allgemeiner Konzepte müssen geteilt werden

# Ontologien

8

- In der Philosophie: Theorie über die Natur der Existenz
- Im SW: Dokument, das formal die Relation zwischen Termen definiert
- Besteht normalerweise Taxonomie und Schlussregeln
- Vergleich und Kombination von Informationen zwischen zwei Datenbasen
- Möglichkeit allgemeine Bedeutungen zu finden

# Ontologien

9

- Bedeutung von Termen kann auf der Webseite durch einen Pointer auf eine Ontologie definiert werden
- Problem: verschiedene Ontologien nutzen verschiedene Terme für die selbe Bedeutung
- Lösung: Äquivalenzrelationen zwischen den Ontologien
- Der Agent „versteht“ die Information nicht
- Aber: bessere Manipulation der Terme
  - nutzbar und bedeutungsvoll für menschliche Agenten

# Aufgabe

10

- Eine Sprache zur Verfügung stellen, die:
  - Daten und Regeln für das Folgern aus den Daten ausdrücken kann
- Exportieren von Daten und Regeln von existierenden KR-Systemen in das Web
- Logik zum Web hinzufügen, die:
  - Stark genug ist, komplexe Eigenschaften zu beschreiben
  - Berechnbar ist

# Beschreibungslogik als Ontologiesprache

11

- DL sind sehr ausdrucksstark
- Kompromiss zwischen Ausdrucksstärke und Komplexität musste gefunden werden
- Reasoning in der DL SHIQ ist entscheidbar
- Aber hohe Worst-Case-Komplexität (ExpTime)
- Hoch optimierter SHIQ-Reasoner (FaCT)

# Beschreibungslogik SHIQ

12

- Basiert auf Erweiterung der DL ALC mit transitiv geschlossenen primitiven Rollen: S
- Dann Erweiterung mit:
  - Rollenhierarchien(H),
  - Inversen Rollen (I),
  - und Qualifying Number Restrictions (Q).

# Beschreibungslogik SHIQ

13

**Definition 1.** Let  $\mathbf{C}$  be a set of concept names and  $\mathbf{R}$  a set of role names with a subset  $\mathbf{R}_+ \subseteq \mathbf{R}$  of transitive role names. The set of roles is  $\mathbf{R} \cup \{R^- \mid R \in \mathbf{R}\}$ . To avoid considering roles such as  $R^{--}$ , we define a function  $\text{Inv}$  on roles such that  $\text{Inv}(R) = R^-$  if  $R$  is a role name, and  $\text{Inv}(R) = S$  if  $R = S^-$ . We also define a function  $\text{Trans}$  which returns true iff  $R$  is a transitive role. More precisely,  $\text{Trans}(R) = \text{true}$  iff  $R \in \mathbf{R}_+$  or  $\text{Inv}(R) \in \mathbf{R}_+$ .

A role inclusion axiom is an expression of the form  $R \sqsubseteq S$ , where  $R$  and  $S$  are roles, each of which can be inverse. A role hierarchy is a set of role inclusion axioms. For a role hierarchy  $\mathcal{R}$ , we define the relation  $\sqsubseteq^*$  to be the transitive-reflexive closure of  $\sqsubseteq$  over  $\mathcal{R} \cup \{\text{Inv}(R) \sqsubseteq \text{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\}$ . A role  $R$  is called a sub-role (resp. super-role) of a role  $S$  if  $R \sqsubseteq^* S$  (resp.  $S \sqsubseteq^* R$ ). A role is simple if it is neither transitive nor has any transitive sub-roles.

The set of SHIQ-concepts is the smallest set such that

- every concept name is a concept, and,
- if  $C, D$  are concepts,  $R$  is a role,  $S$  is a simple role, and  $n$  is a nonnegative integer, then  $C \sqcap D, C \sqcup D, \neg C, \forall R.C, \exists R.C, \geq n.S.C$ , and  $\leq n.S.C$  are also concepts.

A general concept inclusion axiom (GCI) is an expression of the form  $C \sqsubseteq D$  for two SHIQ-concepts  $C$  and  $D$ . A terminology is a set of GCIs.

Let  $\mathbf{I} = \{a, b, c, \dots\}$  be a set of individual names. An assertion is of the form  $a : C$ ,  $(a, b) : R$ , or  $a \neq b$  for  $a, b \in \mathbf{I}$ ,  $a$  (possibly inverse) role  $R$ , and a SHIQ-concept  $C$ . An Abox is a finite set of assertions.

# Beispiele

14

- **Inverse Rollen**
  - `hasChild = inv(hasParent)`
- **transitive Rollen**
  - z.B. `hasAncestor`
- **Subrollen**
  - `hasParent` ist Subrolle von `hasAncestor`

# Syntax und Semantik

15

**Table 1.** Syntax and semantics of concept descriptions.

Construct name	Syntax	Semantics
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y : (x, y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
at-least restriction	$(\geq nr.C)$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq n\}$
at-most restriction	$(\leq nr.C)$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$

# Reasoning in SHIQ

16

- ExpTime-complete
  - Tableau-Verfahren
- Aber: Worst-case: nondeterministic double exponential time
- Trotzdem „gut“ in der Praxis
- Erfüllbarkeit und Subsumption von SHIQ-Konzepten

# Reasoning in SHIQ

17

- Subsumption kann (in lineare Zeit) auf Erfüllbarkeit reduziert werden
- Einführung einer transitiven Superrolle für alle Rollen der TBox
- Testen eines Konzeptes auf Erfüllbarkeit
- Erweiterung des Konzeptes

$$\hat{C}_0 := C_0 \sqcap \prod_{C \sqsubseteq D \in \mathcal{T}} (\neg C \sqcup D) \sqcap \forall u. (\neg C \sqcup D)$$

# Reasoning in SHIQ

18

## The $\rightarrow_{\sqcap}$ -rule

*Condition:*  $\mathcal{A}$  contains  $(C_1 \sqcap C_2)(x)$ , but it does not contain both  $C_1(x)$  and  $C_2(x)$ .

*Action:*  $\mathcal{A}' := \mathcal{A} \cup \{C_1(x), C_2(x)\}$ .

## The $\rightarrow_{\sqcup}$ -rule

*Condition:*  $\mathcal{A}$  contains  $(C_1 \sqcup C_2)(x)$ , but neither  $C_1(x)$  nor  $C_2(x)$ .

*Action:*  $\mathcal{A}' := \mathcal{A} \cup \{C_1(x)\}$ ,  $\mathcal{A}'' := \mathcal{A} \cup \{C_2(x)\}$ .

## The $\rightarrow_{\exists}$ -rule

*Condition:*  $\mathcal{A}$  contains  $(\exists r.C)(x)$ , but there is no individual name  $z$  such that  $C(z)$  and  $r(x, z)$  are in  $\mathcal{A}$ .

*Action:*  $\mathcal{A}' := \mathcal{A} \cup \{C(y), r(x, y)\}$  where  $y$  is an individual name not occurring in  $\mathcal{A}$ .

## The $\rightarrow_{\forall}$ -rule

*Condition:*  $\mathcal{A}$  contains  $(\forall r.C)(x)$  and  $r(x, y)$ , but it does not contain  $C(y)$ .

*Action:*  $\mathcal{A}' := \mathcal{A} \cup \{C(y)\}$ .

**Fig. 1.** Transformation rules of the satisfiability algorithm for  $\mathcal{ALC}$ .

# Reasoning in SHIQ

19

## The $\rightarrow_{\geq}$ -rule

**Condition:**  $\mathcal{A}$  contains  $(\geq nr)(x)$ , and there are no individual names  $z_1, \dots, z_n$  such that  $r(x, z_i)$  ( $1 \leq i \leq n$ ) and  $z_i \neq z_j$  ( $1 \leq i < j \leq n$ ) are contained in  $\mathcal{A}$ .

**Action:**  $\mathcal{A}' := \mathcal{A} \cup \{r(x, y_i) \mid 1 \leq i \leq n\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\}$ , where  $y_1, \dots, y_n$  are distinct individual names not occurring in  $\mathcal{A}$ .

## The $\rightarrow_{\leq}$ -rule

**Condition:**  $\mathcal{A}$  contains distinct individual names  $y_1, \dots, y_{n+1}$  such that  $(\leq nr)(x)$  and  $r(x, y_1), \dots, r(x, y_{n+1})$  are in  $\mathcal{A}$ , and  $y_i \neq y_j$  is not in  $\mathcal{A}$  for some  $i \neq j$ .

**Action:** For each pair  $y_i, y_j$  such that  $i < j$  and  $y_i \neq y_j$  is not in  $\mathcal{A}$ , the ABox  $\mathcal{A}_{i,j} := [y_i/y_j]\mathcal{A}$  is obtained from  $\mathcal{A}$  by replacing each occurrence of  $y_i$  by  $y_j$ .

**Fig. 2.** The transformation rules handling unqualified number restrictions.

# Beschreibungsontologien in SHIQ

20

- **Formalisierung der Ontologie in einer Tbox:**
  - Restriktion der möglichen Welt durch Restriktion der erlaubten Interpretationen (mit GCIs)
  - Definition der relevanten Begriffe durch Konzeptdefinitionen
  - Eindeutige Definition der Konzeptnamen
    - ✦ Keine multiplen Definitionen
    - ✦ Keine zyklischen Definitionen
    - ✦ Definierte Namen dürfen nicht in den GCIs auftreten

# Beispiel

21

- Menschen sind entweder Zauberer, oder nicht

$\text{Human} \sqsubseteq \text{Muggle} \sqcup \text{Sorcerer}$  and  $\text{Muggle} \sqsubseteq \neg\text{Sorcerer}$

- Menschen haben genau zwei Eltern und alle Eltern und Kinder von Menschen sind Menschen

$\text{Human} \sqsubseteq \forall \text{hasParent}.\text{Human} \sqcap (\leq 2 \text{ hasParent}.\top) \sqcap (\geq 2 \text{ hasParent}.\top) \sqcap \forall \text{hasParent}^-. \text{Human}$

- Transitive Rolle hasAncestor und die Rolleninclusion

$\text{hasParent} \sqsubseteq \text{hasAncestor}$

- Menschen die einen Zauberer als Vorfahren haben, sind selbst einer

$\text{Human} \sqcap \exists \text{hasAncestor}.\text{Sorcerer} \sqsubseteq \text{Sorcerer}$

# Beispiel

22

- Konzepte Parent und Grandparent

$$\text{Parent} \equiv \text{Human} \sqcap \exists \text{hasParent}^- . \top$$
$$\text{Grandparent} \equiv \exists \text{hasParent}^- . \text{Parent}$$

- Aus der Tbox zusammen mit dem Fakt, dass hasAncestor eine transitive Superrolle von hasParent ist, wird folgende Beziehung impliziert

$$\text{Grandparent} \sqcap \text{Sorcerer} \sqsubseteq \exists \text{hasParent}^- . \exists \text{hasParent}^- . \text{Sorcerer}$$

# DAML+OIL

23

- DAML+OIL ist eine Semantic-Web-Ontology Sprache
- Basiert auf RDF (Resource Description Framework)
- Semantik kann in DL übersetzt werden
- Formale Ergebnisse der DL können ausgenutzt werden
- Implementierte DL-Reasoner können genutzt werden (FaCT)
  - Reasoning-Service für DAML+OIL-Anwendungen

# RDF

24

- W3C Basis für das Semantic Web
- RDF: beschrifteter gerichteter Graph
- Knoten sind Ressourcen oder Literale
- Graph definiert über eine Meng von Trippeln
- <Subject, Property, Object>
- Alles Beschreibbare ist eine Ressource
- Serialisierung der Daten in XML-Format

# Beispiel

25

```
<daml:Class>
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Human"/>
    <daml:Class rdf:about="#Male"/>
  </daml:intersectionOf>
</daml:Class>

<daml:Restriction daml:minCardinalityQ="2">
  <daml:onProperty rdf:resource="#hasChild"/>
  <daml:hasClassQ rdf:resource="#Lawyer"/>
</daml:Restriction>
```

# DAML+OIL Konstruktoren

26

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human $\sqcap$ Male
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor $\sqcup$ Lawyer
complementOf	$\neg C$	$\neg$ Male
oneOf	$\{x_1 \dots x_n\}$	{john, mary}
toClass	$\forall P.C$	$\forall$ hasChild.Doctor
hasClass	$\exists r.C$	$\exists$ hasChild.Lawyer
hasValue	$\exists r.\{x\}$	$\exists$ citizenOf.{USA}
minCardinalityQ	$(\geq n r.C)$	$(\geq 2$ hasChild.Lawyer)
maxCardinalityQ	$(\leq n r.C)$	$(\leq 1$ hasChild.Male)
inverseOf	$r^-$	hasChild $^-$

Fig. 1. DAML+OIL constructors

# DAML+OIL-Axiome

27

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
sameClassAs	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \equiv P_2$	cost $\equiv$ price
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
transitiveProperty	$P \in \mathbf{R}_+$	hasAncestor <sup>+</sup> $\in \mathbf{R}_+$
uniqueProperty	$\top \sqsubseteq (\leq 1 P.\top)$	$\top \sqsubseteq (\leq 1 \text{hasMother}.\top)$
unambiguousProperty	$\top \sqsubseteq (\leq 1 P^-\top)$	$\top \sqsubseteq (\leq 1 \text{isMotherOf}^-\top)$

Fig. 2. DAML+OIL axioms

# Beispiel: XML-Schema-Datatypes

28

```
<xsd:simpleType name="over17">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:minInclusive value="18"/>
  </xsd:restriction>
</xsd:simpleType>

<daml:Class rdf:ID="Adult">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Person"/>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#age"/>
      <daml:hasClass rdf:resource="#over17"/>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>
```

- Web-Ontology-Sprache
- Weiterentwicklung von DAML+OIL
- Existiert in verschiedenen Ausführungen
  - OWL Full
  - OWL Lite