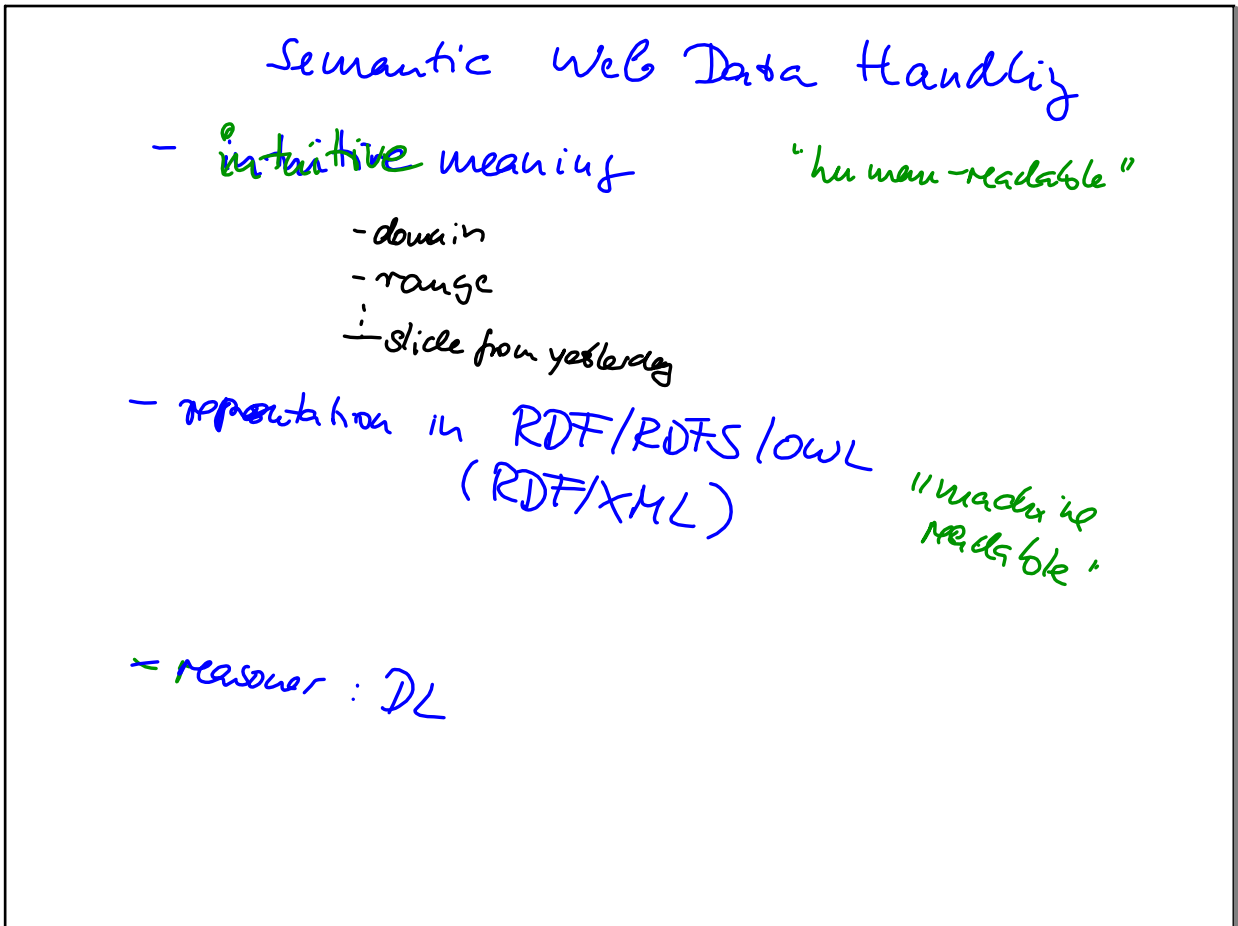


Jan 9-10:09



Jan 9-10:24

Example: $hasDaughter(john, alice)$
 (range)

intn. lie: range of $hasDaughter$ is Female

RDFS: $:hasDaughter \text{ rdfs:range } :Female$

DL: $T \sqsubseteq \forall hasDaughter. Female$

\Rightarrow Reasoner:

$hasDaughter(john, alice)$

$T \sqsubseteq \forall hasDaughter. Female$

$T(john)$

$(\forall hasDaughter. Female)(john)$

Female(alice)

Jan 9-10:28

Example: $hasDaughter(john, alice)$
 $hasDaughter \sqsubseteq hasChild$
 $T \sqsubseteq \forall hasDaughter. Female$
 $(\exists hasChild. Male)(john)$
 $Male \sqsubseteq \neg Female$
 $Female \sqsubseteq \neg Male$

]} : Male
 owl:disjointWith
 :Female

- how many children does John have?
 . open world: arbitrary many.
 \rightarrow how many children at least?

Conclusions:
 $Female(Alice)$
 $hasChild(john, alice)$
 $\exists hasChild. Male:$
 $hasChild(john, x)$
 $Male(x)$

possible question: does John have exactly one child?
 $(\exists = 1 hasChild. T)(john)$

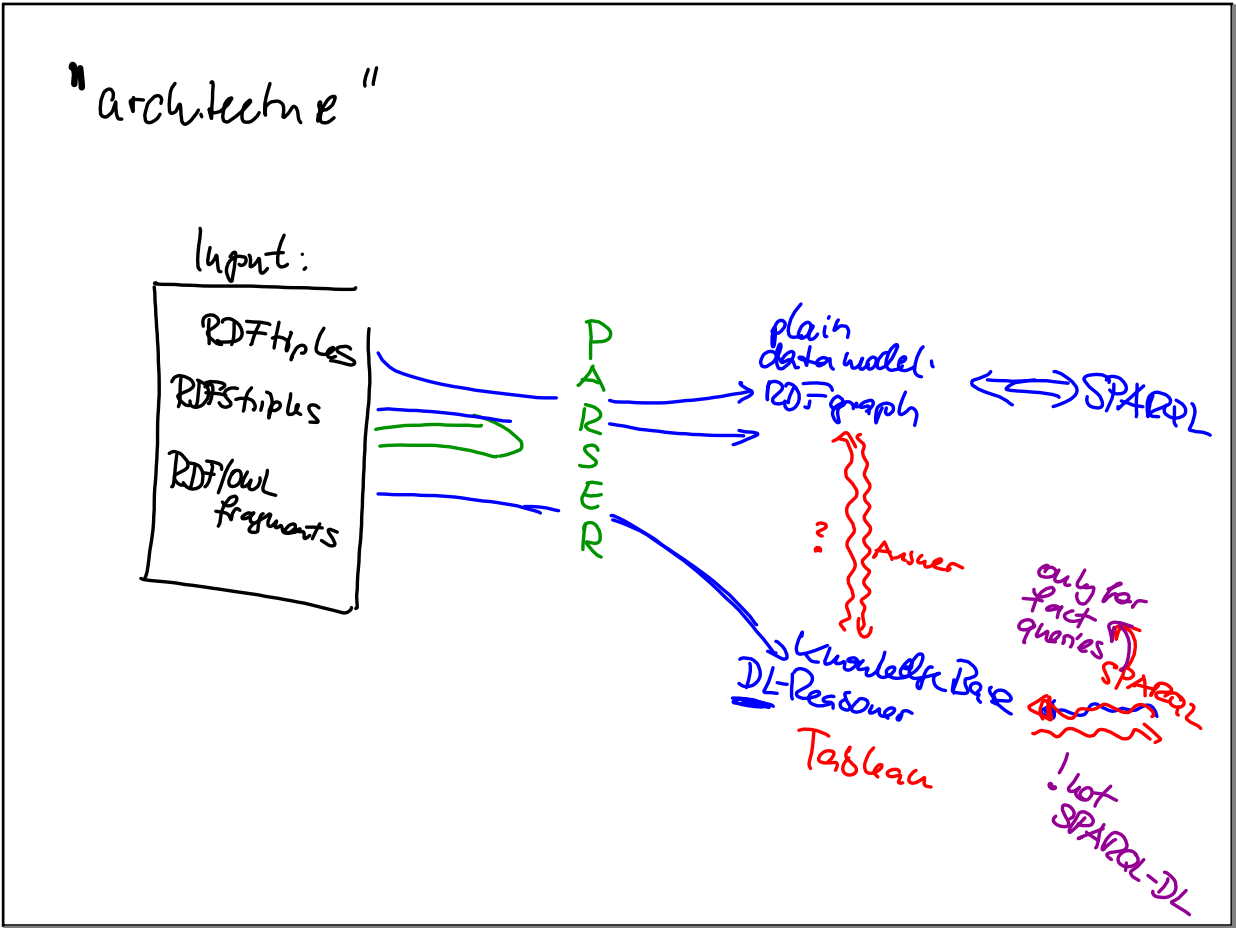
therefore:
 $hasChild(john, y)$ $\exists some\ child$

$\exists exactly\ 1: \exists z_1, z_2 : hasChild(john, z_1) \wedge hasChild(john, z_2)$
 $\rightarrow z_1 = z_2$

$\neg (hasChild(john, z_1) \wedge hasChild(john, z_2))$
 $\neg hasChild(john, z_1) \vee \neg hasChild(john, z_2)$
 $\square z_1 \neq alice \quad \square z_2 \neq x$

$alice = x$
 $Female(x)$
 $(\neg Male)(x)$
 $\neg Male(x)$
 \square

Jan 9-10:34



SE 313

Person \equiv Male \cup Female

RDF: `:Person owl:equivalentClass [owl:unionOf (:Male :Female)]`

Equivalent: `:Person owl:equivalentClass [owl:unionOf [a rdf:List; rdf:first :Male; rdf:rest [a rdf:List; rdf:first :Female; rdf:rest rdf:nil]]]]`

Jan 9-10:56