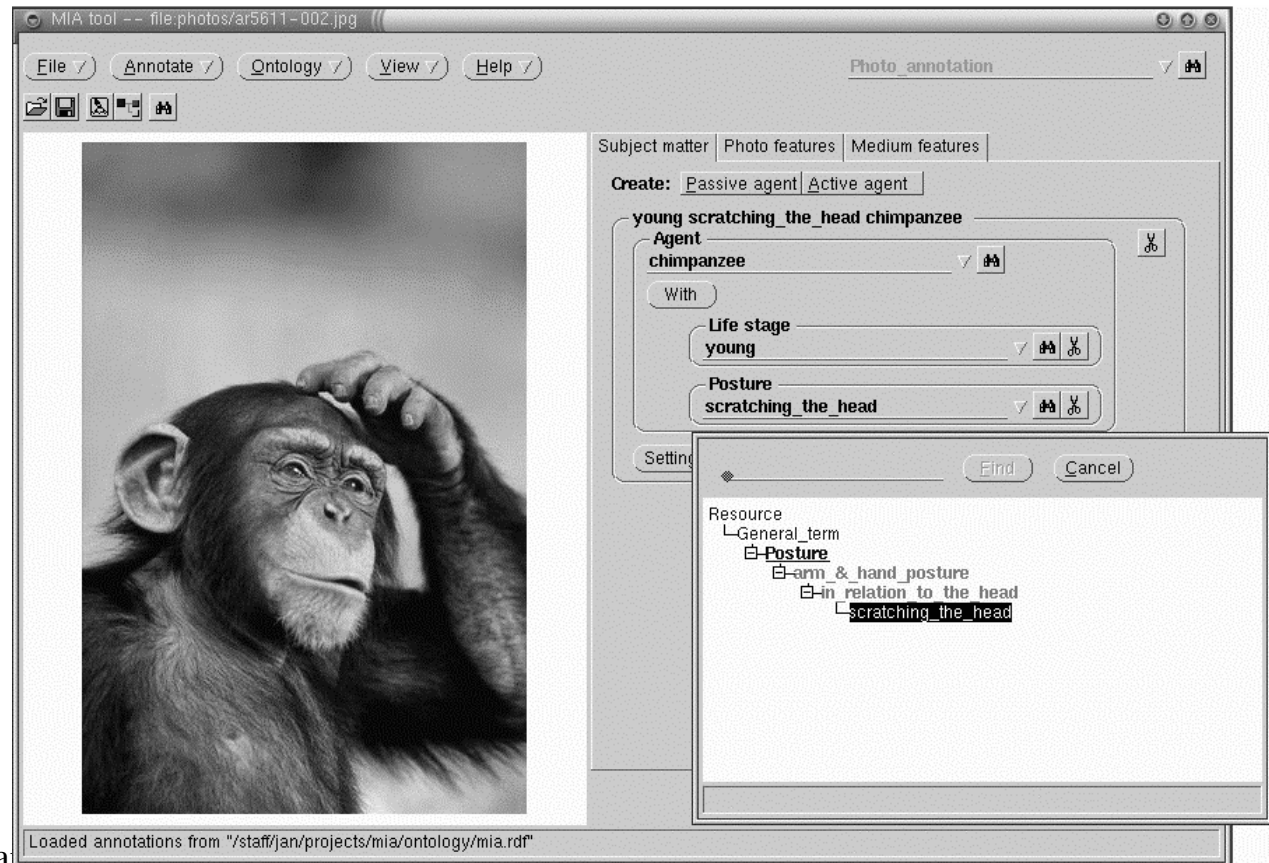# Use cases:
collection management

Guus Schreiber

with contributions from

Stephen Buswell, Jeremy Carrol,

Nick Gibbins and Mike Smith

Collection management area

- Large data/text/image/multimedia/website sets with a common theme/context/focus

Classes as instances of other classes

Aircraft-type
    no-of-engines: integer >0
    propulsion: {propeller, jet}

Fokker-50
    instance of Aircraft-type
    no-of-engines = 2
    propulsion = jet

Aircraft
    no-of-seats: positive integer
    owner: Airline

Fokker-50
    subclass of Aircraft
    no-of-seats: 40-50

PH-851
    instance of Fokker-50
    no-of-seats = 45
    owner = KLM

Some remarks about classes as instances

- Common modeling issue
  - Martin: all classes are instances, some instances are not classes (introduces notion of "power class")
  - Cf. Protégé-2000 modeling tool

- Representing Fokker-50 as a subclass of Aircraft-type does not solve the problem
  - Instances of Fokker-50 are not instances of Aircraft-type
  - Fixing class/instance border leads to constant revisions of ontology (cf. HPKB paper by Valente et al.)

- RDF and DAML+OIL allow it (?!), but semantics are problematic

Constraints (use cases 2, 3, 4)

- Examples:
  - `wing-spar.length < wing.length`
  - `furniture.style = Late-Georgian <=> furniture.culture = British AND furniture.date-created = 1760-1811`

- Can be expressed in DAML+OIL solution
  - Late Georgian things are an intersection of British things and 1760-1811 things

- But:
  - Syntax is awful (debatable whether this is an issue)
  - Language idioms required for usage in practice

Default knowledge (1, 2, 4)

- Examples:
  - `living things don't fly`
  - `Late Georgian chests of drawers are typically made of mahogany wood`
- Essential for semantic search in collection
- Solution: distinguish between definition and query semantics?!

Part-whole relation (use cases 2, 3, 4)

- Examples:
  - a wing spar is part of a wing assembly
  - chests of drawers have feet with their own style
- Most items in collections have some internal structure

Relation typing

- Example:
  - property P represents a time relation

- Possible in RDF
  - Define TimeProperty as subclass of Property
  - rdf:type P TimeProperty

- Is this desirable for OWL?
  - E.g. in a layered approach
  - Mechanism can also be used to define part-whole relation

Other requirements

- Synonyms / lexical term -> concept (1, 5)
- Provenance (1, 5)
  - Typical examples: distinguishing annotations by experts and non-experts, authorship of hyperlinks.
- Version management (1)
  - Ability to extend/revise ontology.
- Query support (1)
  - Ability to reason with falsehood, e.g. "whales as fish".
- Support for content standards (1, 3)
  - General thesauri (WordNet, TGN) as well as domain-specific thesauri (AAT, ICONCLASS) are often used to standardize annotations.

Using existing hierarchies from thesauri

```
<color>
 <chromatic color>
   pink
    vivid pink
    strong pink
    <intermediate pink>
       purplish pink
          brilliant purplish pink
       yellowish pink
   <neutral color>
```

[Content
interoperability issue?!]