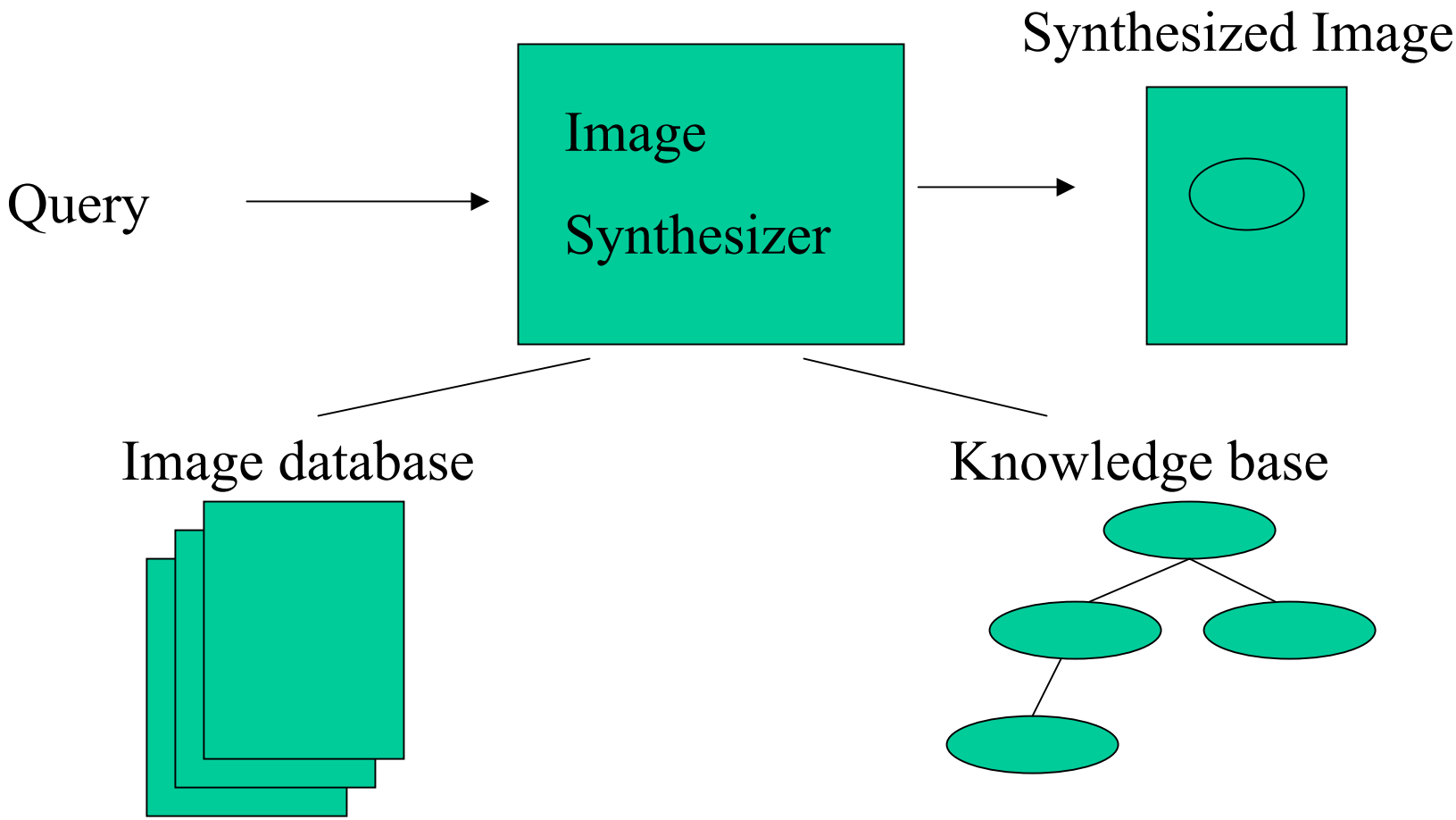
Problem


Synthesized Image
Query
Image
Synthesizer
Image database
Knowledge base

Hierarchical Content-Based Image Retrieval

Progress and Status

Dipti Vaidya

Bob Curry

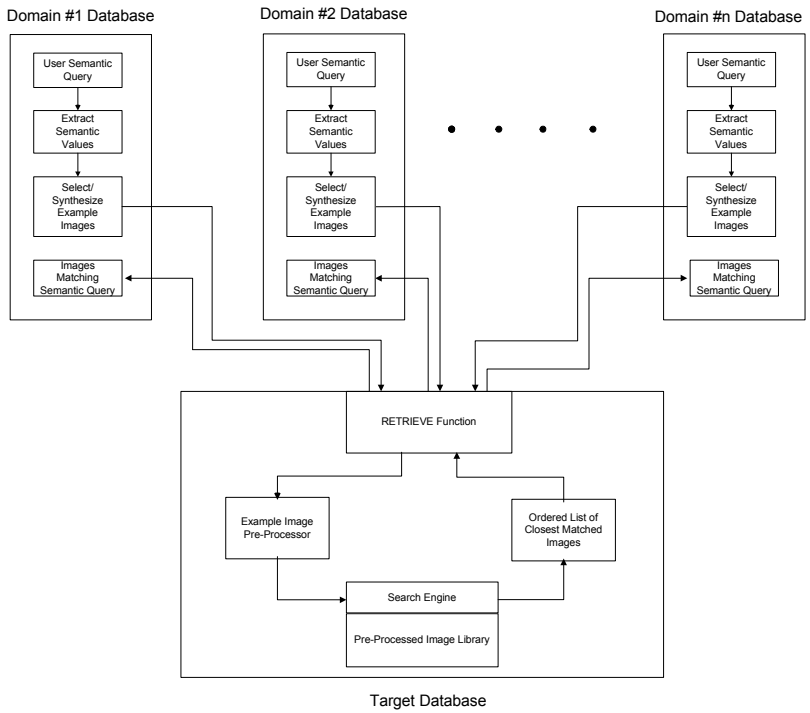
3/14/2003

Agenda

- Current System - Bob
  - Overview and Description of System
  - Changes in Approach since Last Review
  - Walk-Through Example and Demo
  - Results and Conclusions
- Improvements to Current System - Dipti

Overview and Description

- Hierarchical RSC and RIE methods are used
- User semantic query results in synthesis of example images
- Examples processed by Target DB to retrieve closest matches

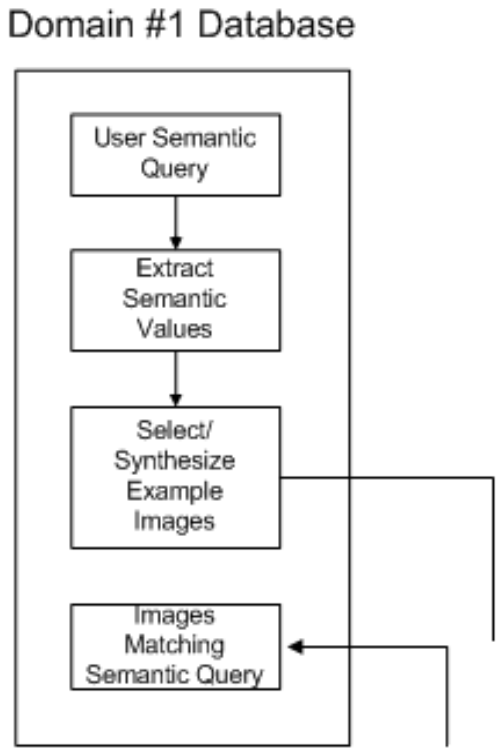


Problems to be Solved

- A method for capturing knowledge in the problem domain that allows a semantic-oriented query to be resolved to a set of semantic features.
- A method that allows semantic features in the knowledge system to be mapped to a set of images containing representations of the semantic feature.
- A query language which can be used to process the problem domain's semantic database.

Problems to be Solved

- Focus of this work is on the Domain Database
  - Given a semantic query relevant to a particular problem domain, synthesize a set of image exemplars that can be used by image matching CBIR system.



Structured Annotation

- SA is foundation of knowledge/query representation
  - Agent, Object, Action, Setting describe semantic feature
    - Show images of Giraffe (agent) Eating (action) Leaves (object) in Forest (setting)
  - Knowledge representation in these terms
    - Ontology using Protégé tool

Updates

- Expanded notion of constraints
  - Agent/Object Class Constraints
    - "Does performing this Action on this Agent and Object make sense semantically?"
      - Images of "Giraffe Eating Lion" are not semantically relevant and should not be synthesized
  - Image Synthesis Constraints
    - "Can the clip images be combined in a way that is semantically relevant?"
      - Images of "Person TalkingTo Person" should show the two people facing each other
      - Settings of clip images may be important
      - Perceive distance of clip image from observer may be important

Updates

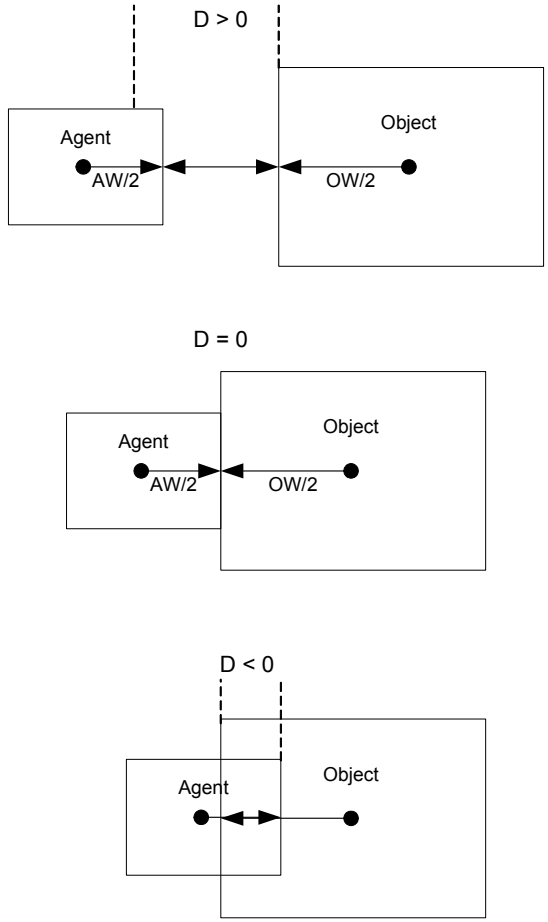
- Constraints (continued)
  - Constraints are defined by Actions
    - Each action contains slots for AgentClass and ObjectClass that limit image synthesis to Agent and Object instances of these classes only
    - Each action contains slots defining required orientation of Agent to the Object and orientation of Object to the Agent. Images are not synthesized unless these constraints can be satisfied.
    - Each action contains slots specifying Distance Match and Setting Match requirements

Updates

- Relative Positioning in image synthesis
  - Defines distance between midpoints of Agent/Object bounding rectangle
  - Defines placement of Agent wrt Object by Orientation constraints
  - Keeps position specification at the semantic level

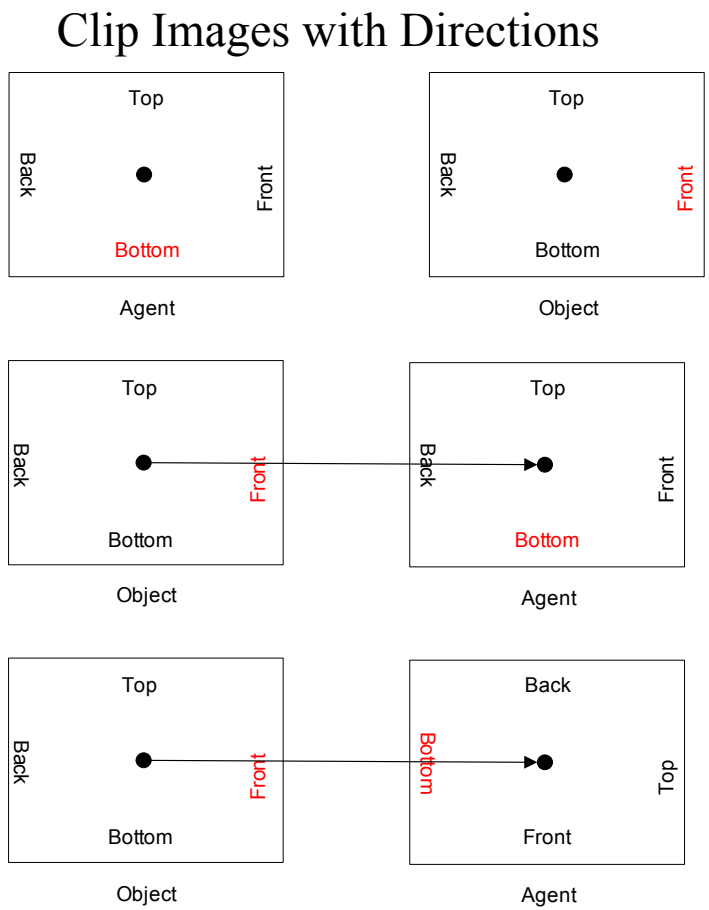
Updates

- RelPos defines distance of centroids
- AgToObjOrient defines placement of Agent Centroid
- ObToAgOrient defines Agent Rotation



Updates

- AgToObOrient = Front
- ObToAgOrient = Bottom
- Position Agent wrt Object
- Rotate Agent if possible



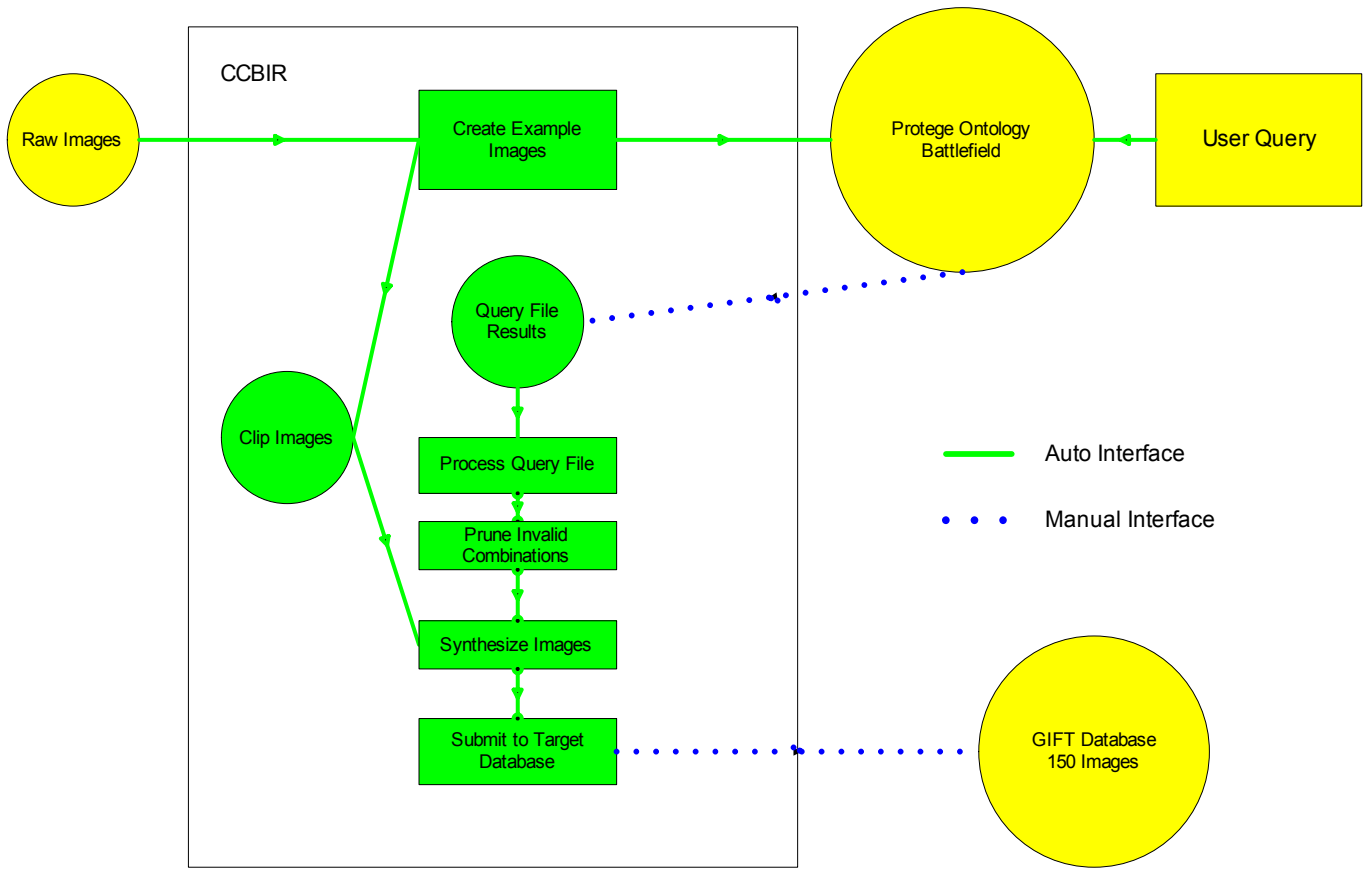
The Action Class

Template Slots					
Slot name	**Documentation**	**Type**	**Allowed Values/Classes**	**Cardinality**	**Default**
AbsolutePositioning	If true, DeltaX, DeltaY, DeltaZ, and Theta are used to postition the Agent relative to the Object.	Boolean		1:1	false
AgToObOrientation		Symbol	Front, Any, Top, LSide, RSide, Bottom, Back	0:1	
RelDist	RelDist is used when AbsolutePositioning is false to position the Agent image with respect the the Object image in the synthesized image.	Float		0:1	
AgentClass	required class for the Agent	ClassList		0:1	
Theta		Float		0:1	0.0
ObToAgOrientation		Symbol	Front, Any, Top, LSide, RSide, Back, Bottom	0:1	
SettingMatchRequired		Boolean		0:1	false
ObjectClass	required class for the object	ClassList		0:1	
DeltaZ		Integer		0:1	0
DeltaX		Integer		0:1	0
DistMatchRequired		Boolean		0:1	true
DeltaY		Integer		0:1	0

Example – The Battlefield

- Domain concepts include Weapons, People, Structures, Terrain Features as well as actions such as Attacking, Retreating, etc. represented in the Protégé tool
- Image Synthesis uses CCBIR
- Target Database implemented in GIFT
- Provides reasonable complexity and a set of semantic concepts that are visually distinct

Implementation Block Diagram



Candidate Test Queries

- Developed queries to test significant aspects of system
  1. Direct match to a single example image. "Show images of Bombers"
  2. Direct match to a set of example images. "Show images of Land Weapons"
  3. Synthesized single example image. "Show images of JSOW being dropped from Bombers"
  4. Synthesized set of example images, no pruning. "Show images of Attack aircraft approaching Land Weapons – setting match not important"
  5. Synthesized set of example images with pruning. "Show images of Attack aircraft approaching Land Weapons – setting match is important"
  6. Synthesized set of example images with Action Constraints. "Show a weapon dropping another weapon where the dropping weapon is an Airplane and the dropped weapon is an ATGMissile"
  7. Synthesized example image with rotation. "Show a JSOW undershooting Bomber".
  8. Synthesized example image with inflation and orientation alignment. "Show Attack aircraft retreating from a JSOW missile"

Query Number	Agent PAL Query	Object PAL Query	Action	Resulting Agent Instances	Resulting Object Instances
1	TestCase1		None	B2Bomber	
2	TestCase2		None	M1A1-1 M1A1-2 Tiger AA1 155Howitzer MLRS	
3	TestCase3	TestCase1	DroppedFromAP	JSOW	B2Bomber
4	TestCase4	TestCase2	Approaching	A10Warthog	M1A1-1 M1A1-2 Tiger AA1 155Howitzer MLRS
5	TestCase4	TestCase2	Approaching	A10Warthog	M1A1-1 M1A1-2 Tiger AA1 155Howitzer MLRS
6	TestCase5	TestCase6	DroppedFromAP	JSOW Maverick	A10Warthog C17Airplane B2Bomber FighterF117 FighterF15Flying FighterF15Ground
7	TestCase3	TestCase1	Undershooting	JSOW	B2Bomber
8	TestCase4	TestCase3	RetreatingFrom	A10Warthog	JSOW

Gift Image Database Contents

- Baseline GIFT image “collection” contains
  - 100 images of various outdoor scenes
    - Animals, landscapes, etc
  - 40 images of various military scenes
    - Aircraft, buildings, vehicles, etc
- For each query, the resultant synthesized images were combined with the baseline collection to obtain a new collection.

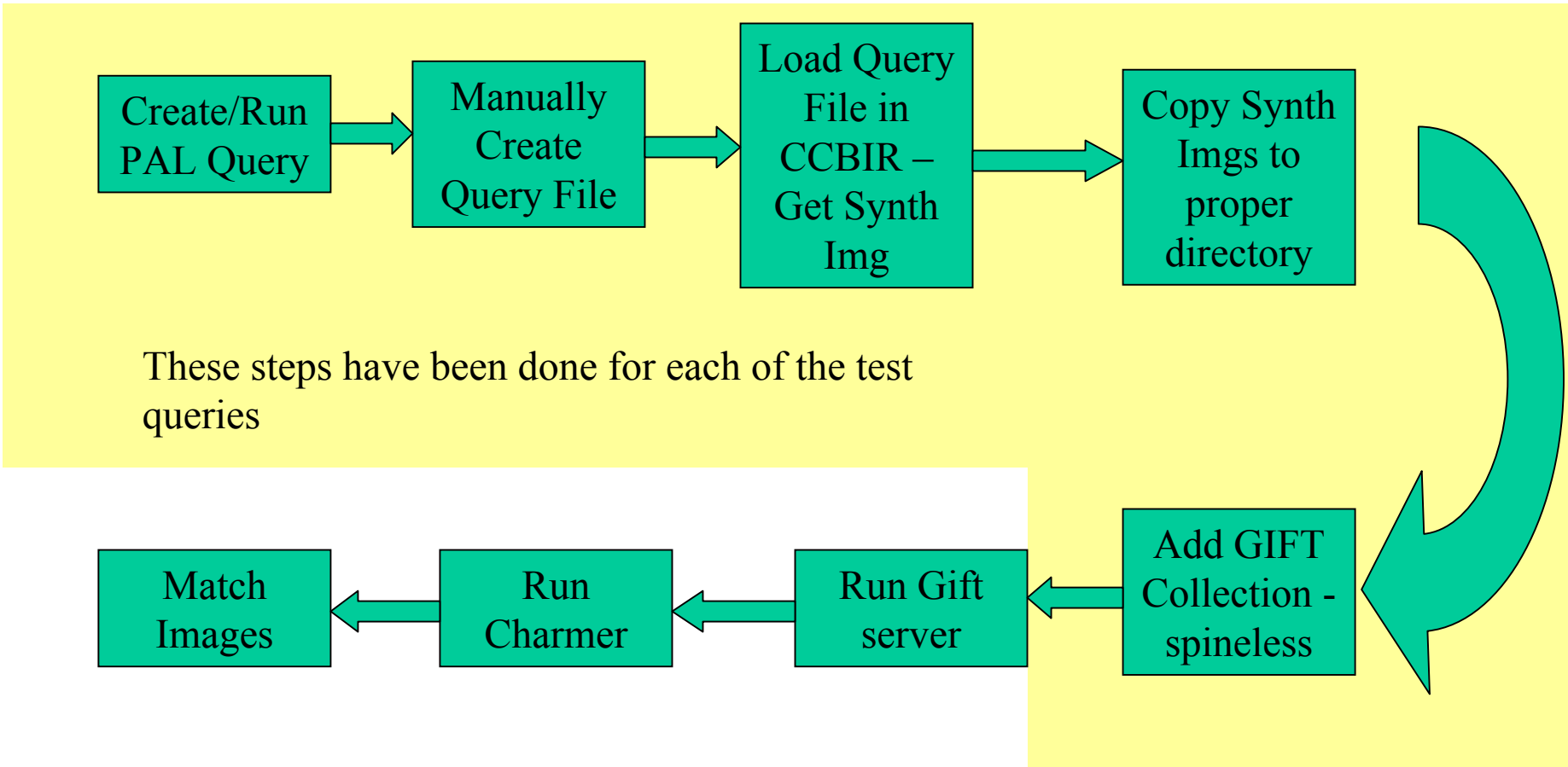
Limitations of Test Environment

- GIFT, Protégé, and CCBIR are three separate programs running on two separate OSs
  - All three have programmatic interfaces that should facilitate automating the interfaces
- Manual/Interactive interfaces to these programs require human intervention to simulate end-to-end query processing

Steps in Test Environment

- Create Clip Images in CCBIR (do this once)
- For each Query
  1. Create appropriate PAL query(ies) and run them against Protégé ontology
  2. Manually create Query file to record the results of Pal query(ies)
  3. Process Query file in CCBIR to get synthesized images
  4. Copy synthesized images to directory containing baseline GIFT image files
  5. Run gift-add-collection on spineless
  6. Run gift server
  7. Open Charmer in browser and select proper collection
  8. Identify images that were synthesized and use them as match query
  9. Obtain results from Charmer query

Steps in Test Environment



Demo of Battlefield

Query Results and Discussion

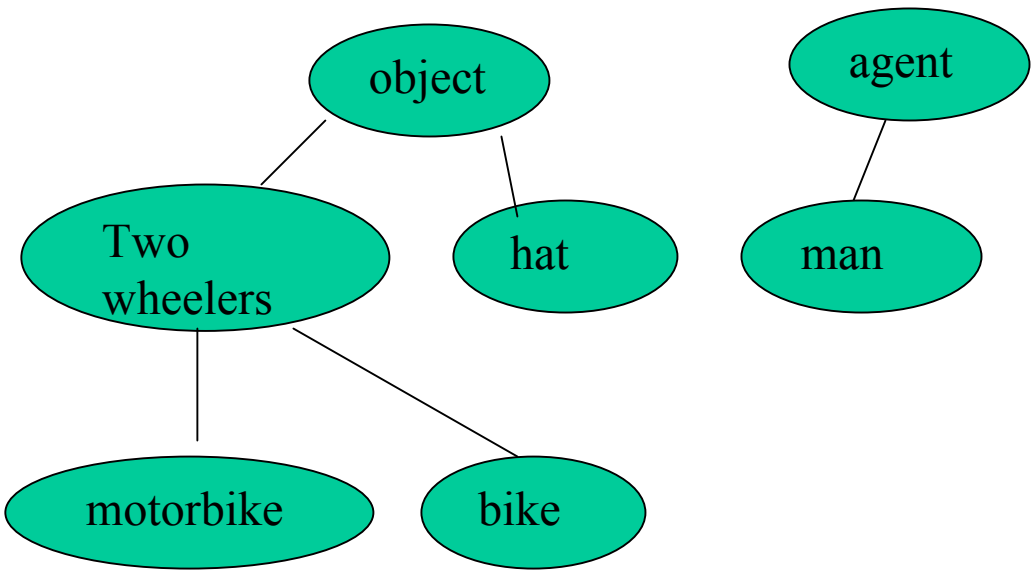
- For GIFT, match strength appears to have a sharp cutoff point for relevance/non-relevance
- Different matching algorithms have some impact on validity of matches
- Semantic differences that are not distinctly observable are not matched well
  - A JSOW missile looks like several other different missiles. This is one form of ambiguity
- Image setting has a significant effect on matching, at least in GIFT
- In general, a semantic query resulted in extracting images from the Target Database that correlated to the meaning of the query

Remaining Items

- Complete data analysis and quantify metrics

Example (Bob's system)

Ontology



Sample database

man

hat

bike

Role

- Wears(agent, object) = object above agent
- Rides(agent,two-wheeler) = agent above two wheeler

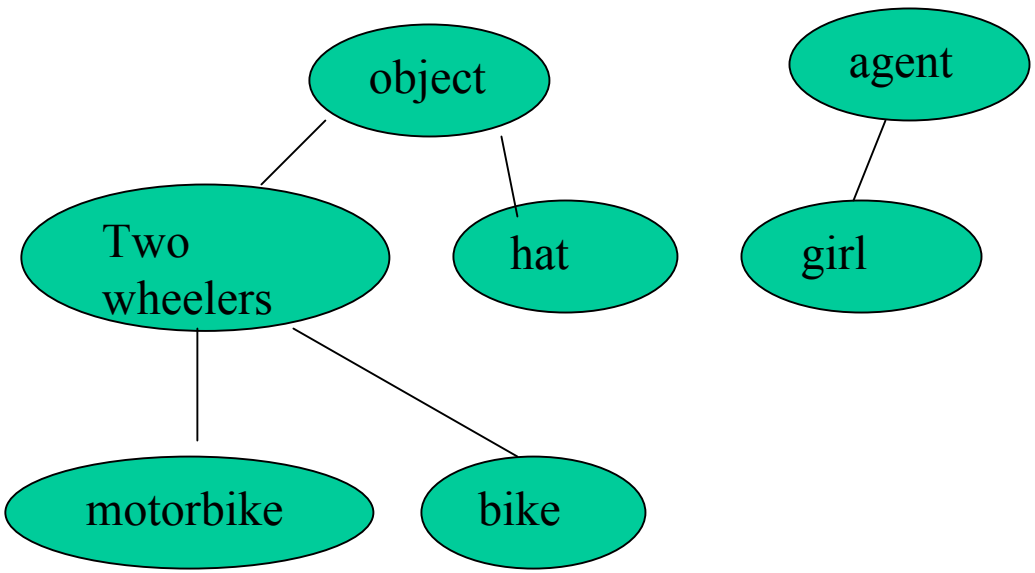
Example (simple queries)

Query: man rides a bike

Query: man wears hat

Example (Dipti's system)

Ontology



Sample database

man

hat

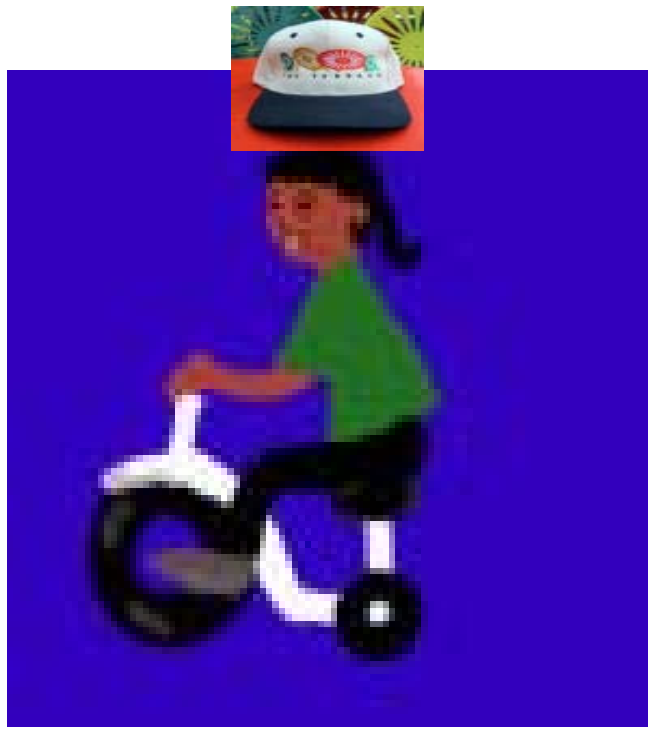
Rides(Girl, two-wheeler)

Role

- Wears(agent, object) = object above agent
- Rides(agent,two-wheeler) = agent above two wheeler

Example (composite queries)

Query: girl rides a bike and wears a hat



Challenges

- Representing the semantic content of an image
- Dynamic classification and image reuse
- Identifying objects in an image
- Resolving queries
- Synthesizing images

Dynamic Classification & Reuse

Flexible extensible descriptors are required as the same image can be reused from many different perspectives and can be classified by different interpretations

Eg: Jane riding Biking may be classified as an image of Jane, girl, a person, a vehicle, a bike.

Should we ask for an image of a " bike ", it should be able to retrieve this image, though it's primary focus is on "Jane "

Images carry annotations. So INCREMENTAL SUPPORT is needed.

Problem Statement

We need a method to represent the image contents such that it supports:

- Automatic clustering of images which are similar in content
- Images can be reused
- Reclassification of these images is possible if needed
- Imprecise retrieval of conceptually similar queries

Ex: if asked for a image of vase and we have pot, it should be able to retrieve

- What might it mean to include Description logic based ontology in Protégé ?
- Is it worth it ?
- Reuse: annotating images from scratch to meta authoring
- Expressiveness: can describe composite images

But Protégé already has ontologies

- Hand written and manually maintained.
- Easy to understand but difficult to provide multiple views
- Enumerated rather than compositional.
- Not reusable

Things that are easy in DL

- Reorganization and views
- Composition
- Inference and constraints

  reasoner / FaCT
- Complex queries and query subsumption

Representing the semantic content of an image

- Use Description Logic to represent the content of the image in a way that it is hierarchical and compositional
- What is DL ?

  It's a language that allows reasoning about information, in particular supporting the classification of descriptors

  Reference: CLASSIC, Meghini(image description using DL),GALEN.

What is in a DL Ontology

- Hierarchy of Primitive or Elementary Concepts
- Definitions of Composite Concepts

  - to name new concepts

- Descriptions (axioms) about Concepts

  - necessary conditions for a concept to be satisfiable

- Rules / Inferences : Definition + Description

  - IF something fits this definition THEN it conforms to that description.

What is in a DL Ontology

Primitive Concepts are placed by the designer whereas ***defined concepts*** are placed automatically by subsumption

Examples:

(def PrimConcept Person Thing)
(def PrimConcept Girl Person)
(def PrimConcept Vehicle Thing)
(def PrimConcept Bike Vehicle)

(def Concept BikeRider (and person(some riding Bike)))
(def Concept HatWearer (and person (some wearing hat)))

Reasoning with DL

- Subsumption : $\subseteq$
  - Basic inferencing tool
  - Checks whether a concept is more general than other
  - Given definitions of concepts A and B and all relevant axioms and other definitions in the model:
    - Are each of A and B satisfiable
      - Is either a kind of nothing (‘Bottom’ 👌)
    - Is A a kind of B?
      - Is an A which is not a B a contradiction in terms?
        - » If so then A is a kind of B
    - Is B a kind of A?
      - Is a B which is not an A a contradiction in terms?
        - » If so then B is a kind of A
    - If both, then A is equivalent to B

Reasoning with DL

Classification

- Collection of descriptions can be classified using subsumption, providing a hierarchy of descriptions ranging from general to specific.

Example

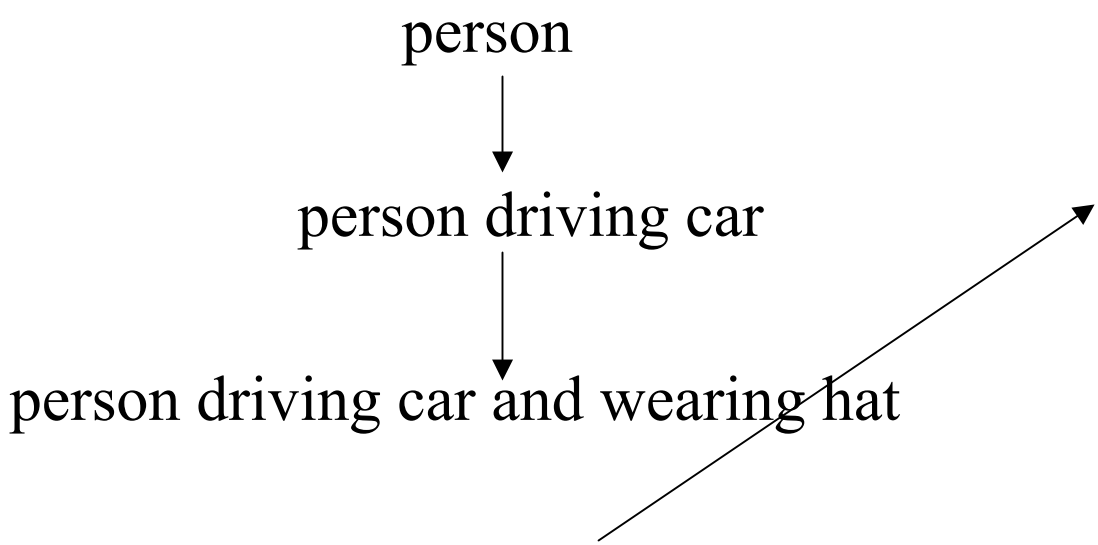
person driving car and wearing hat $\subseteq$ person driving car $\subseteq$ person

New Concept: person wearing hat ?

person driving car and wearing hat $\subseteq$ person driving car $\subseteq$ Person

person wearing hat $\subseteq$ Person

Automatic Classification

- 

New concept or query: person wearing hat ???

Identifying objects in an image

- Segment sections of images and associate them with concepts

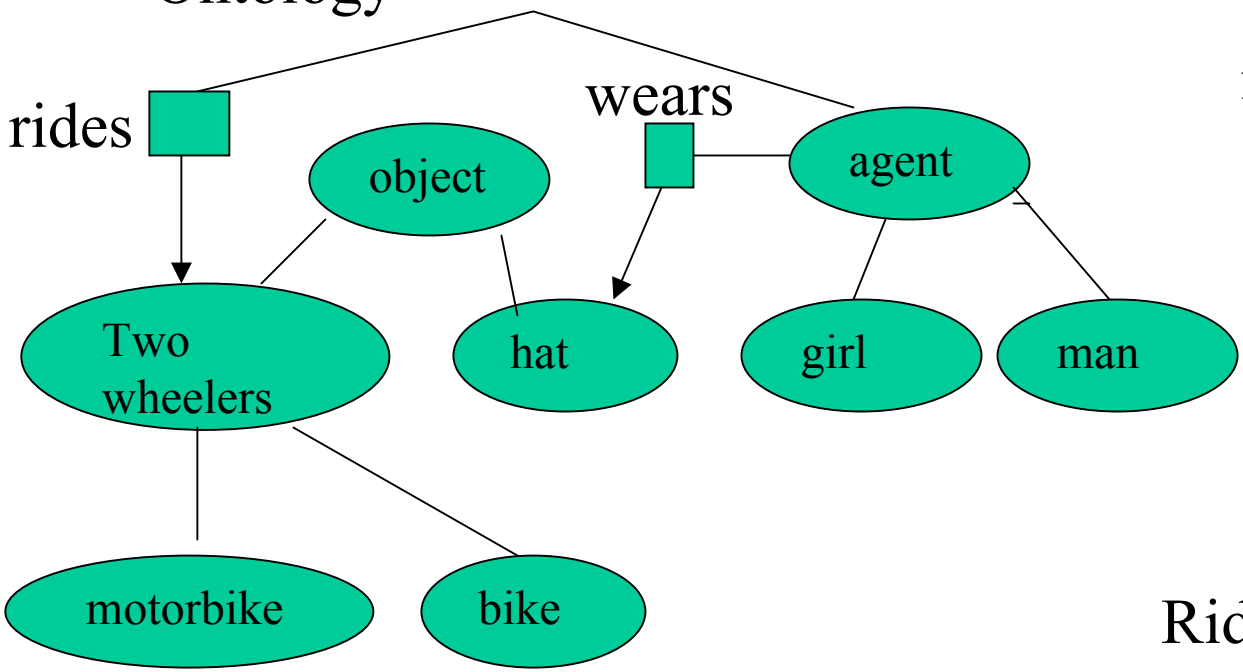
Issues with DL

- Reclassification : As an individual changes it's description it is reclassified with respect to the concept type system we have developed and this can lead to reclassification fo of other individuals.
- Can't say what's the content of a particular image

Ex: what does image2.jpg contain ?

Resolving queries

Ontology



Sample database

man

hat

Rides(Girl, two-wheeler)

Role

- Wears(agent, hat) = hat above agent
- Rides(agent,two-wheeler) = agent above two wheeler

Resolving queries

Query: girl rides a bike and wears a hat

1- attempt to find an image describe with the query (no synthesizing needed), if not found then

2- break query into components (synthesizing needed)

Girl rides bike, girl wears a hat, if not found

3- break query into components (synthesizing needed)

Girl, bike, hats,actions…(for actions, we can have a geo-spatial modeling which maps the action..can use Bob's definitions here.

Synthesizing images

- Issues

- find a method to segment out the image content regions
- Scaling issues.

Ex

we have an image of a big hat and a small man…..

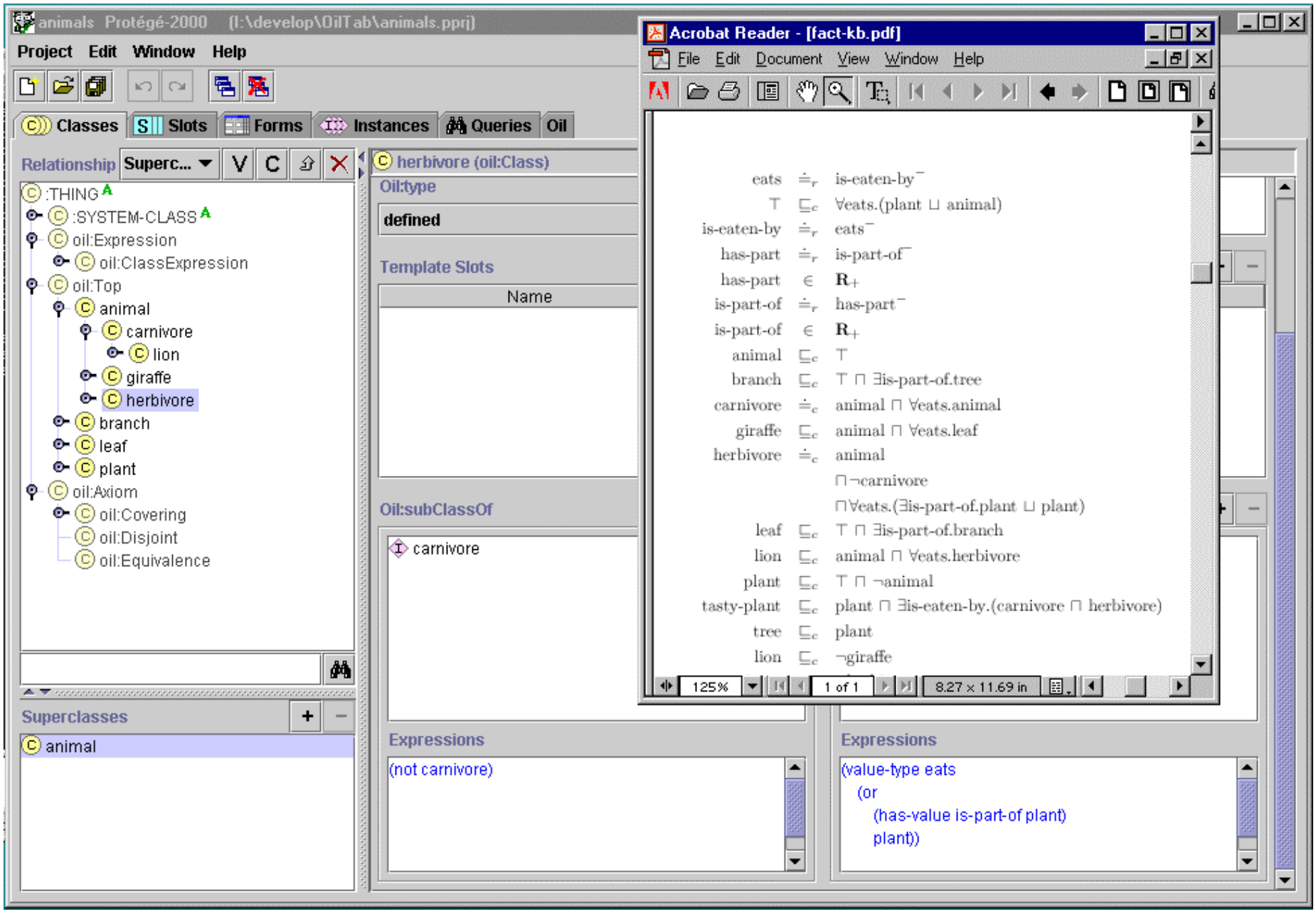
Still not solved…

To do

- Protégé + OIL

Adding OIL tag to Protégé, supports the transformation rules of knowledge representation as shown in the papers on DL. – Done.

- Query interface
- Automatic Selection of algorithms from Gift depending on the query
- Explore the idea of using image models instead of images.


animals Protégé-2000 (I:\develop\OilTab\animals.pprj)
Project Edit Window Help
Classes Slots Forms Instances Queries Oil
Relationship Superc...
:THING
:SYSTEM-CLASS
oil:Expression
oil:ClassExpression
oil:Top
animal
carnivore
lion
giraffe
herbivore
branch
leaf
plant
oil:Axiom
oil:Covering
oil:Disjoint
oil:Equivalence
Superclasses
animal
herbivore (oil:Class)
Oil:type
defined
Template Slots
Name
Oil:subClassOf
carnivore
Expressions
(not carnivore)
Expressions
(value-type eats
(or
(has-value is-part-of plant)
plant))
Acrobat Reader - [fact-kb.pdf]
File Edit Document View Window Help
eats ≐r is-eaten-by⁻
⊤ ⊑c ∀eats.(plant ⊔ animal)
is-eaten-by ≐r eats⁻
has-part ≐r is-part-of⁻
has-part ∈ R+
is-part-of ≐r has-part⁻
is-part-of ∈ R+
animal ⊑c ⊤
branch ⊑c ⊤ ⊓ ∃is-part-of.tree
carnivore ≐c animal ⊓ ∀eats.animal
giraffe ⊑c animal ⊓ ∀eats.leaf
herbivore ≐c animal
⊓¬carnivore
⊓∀eats.(∃is-part-of.plant ⊔ plant)
leaf ⊑c ⊤ ⊓ ∃is-part-of.branch
lion ⊑c animal ⊓ ∀eats.herbivore
plant ⊑c ⊤ ⊓ ¬animal
tasty-plant ⊑c plant ⊓ ∃is-eaten-by.(carnivore ⊓ herbivore)
tree ⊑c plant
lion ⊑c ¬giraffe
125%
1 of 1
8.27 x 11.69 in