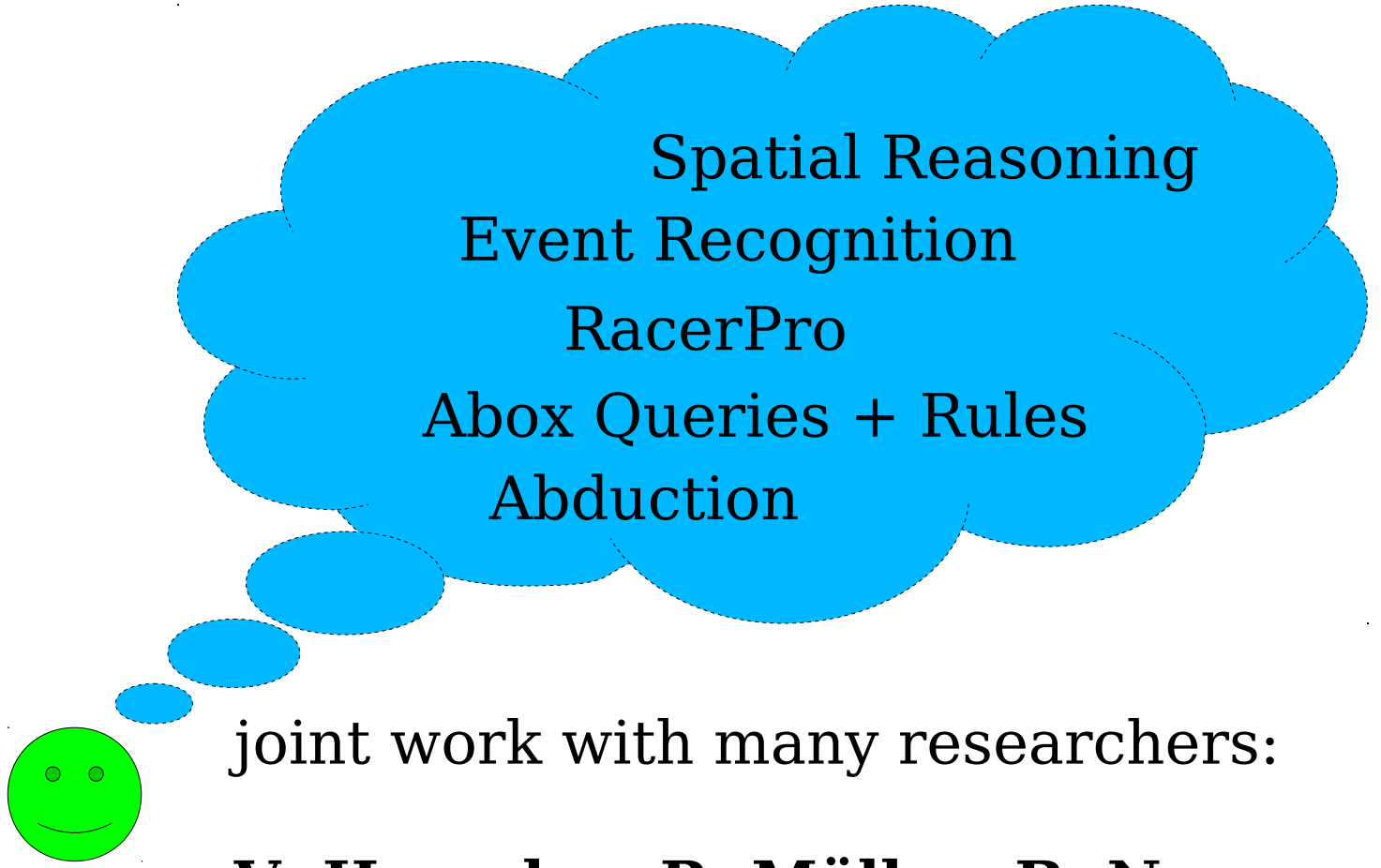
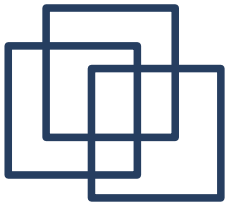
Applications of Description Logics



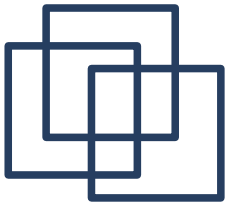
joint work with many researchers:

V. Haarslev, R. Möller, B. Neumann,
A. Kaya, S. Espinosa-Peraldi,
A. Kaplunova, BOEMIE, CASAM, TONES



Why Description Logics?

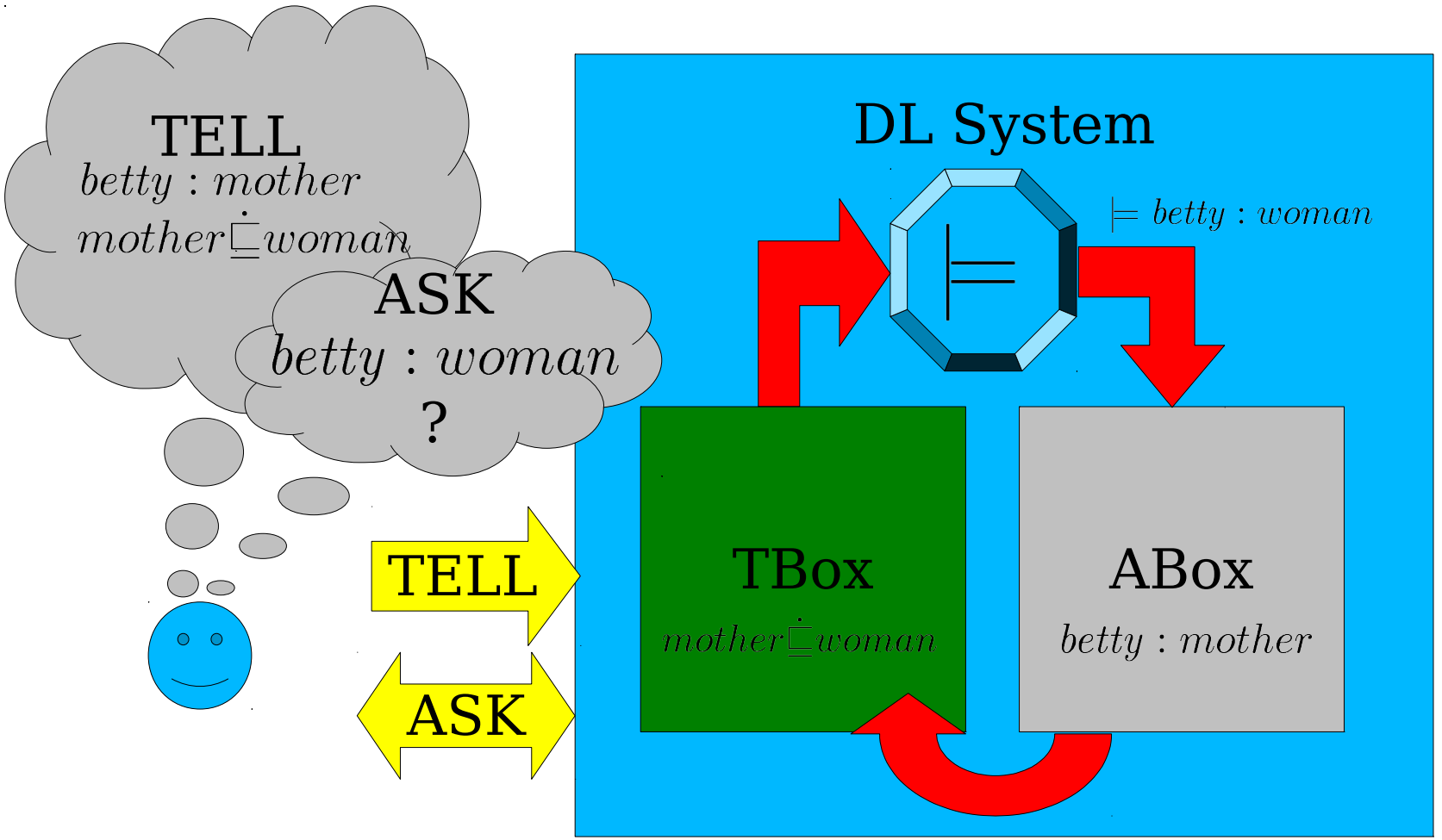
- Formal
  - suitable as **ontology** languages (Gruber definition)
- Well-understood
  - semantics, complexity, implemention techniques
- Family
  - somplexity scalable (e.g., lightweight DLs)
- Decidable
  - unlike FOPL
- Basis for Semantic Web (OWL2, …)
- Many tools available

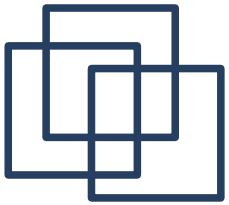


Description Logics : Basic Notions

- Based on first order-logic
  - but variable-free and decidable
  - concept languages, class-based KR
- Central notions:
  - concept (OWL: class): unary predicate
  - role (OWL: property, RDF: predicate): binary predicate / relation
  - Abox individual: constant
  - Container data structures:
    - TBox: Set of terminological + role axioms: $C(x) \rightarrow D(x)$, $C(x) \leftrightarrow D(x)$
    - ABox: individuals and relations: C(i), R(i,j)

Architecture of a DL System





ABox Inference Services

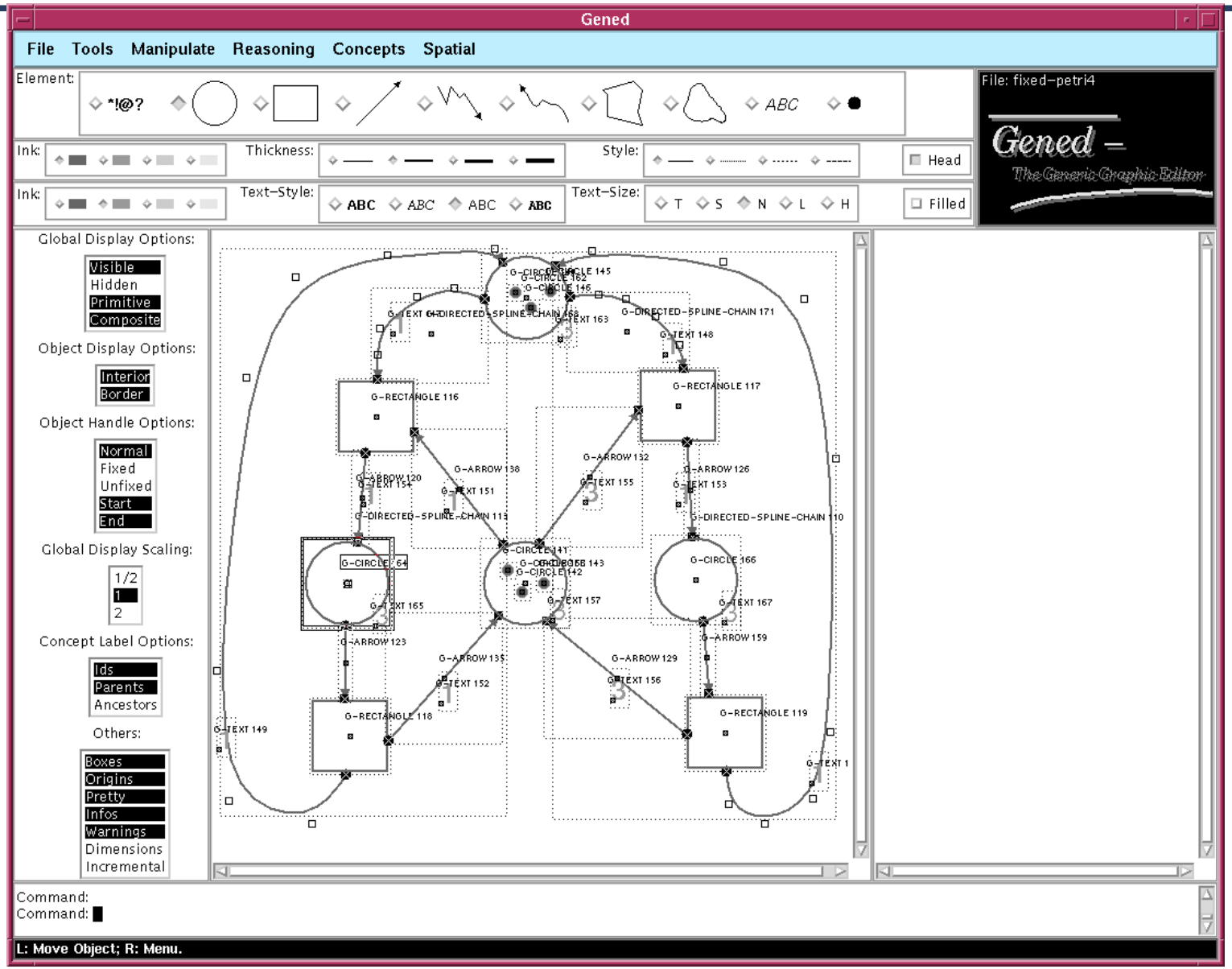
- Abox satisfiability (w.r.t. a possibly empty TBox)
  - does the Abox have a model?

$\{betty : \neg parent, betty : person, (betty, charles) : has\_child\}$

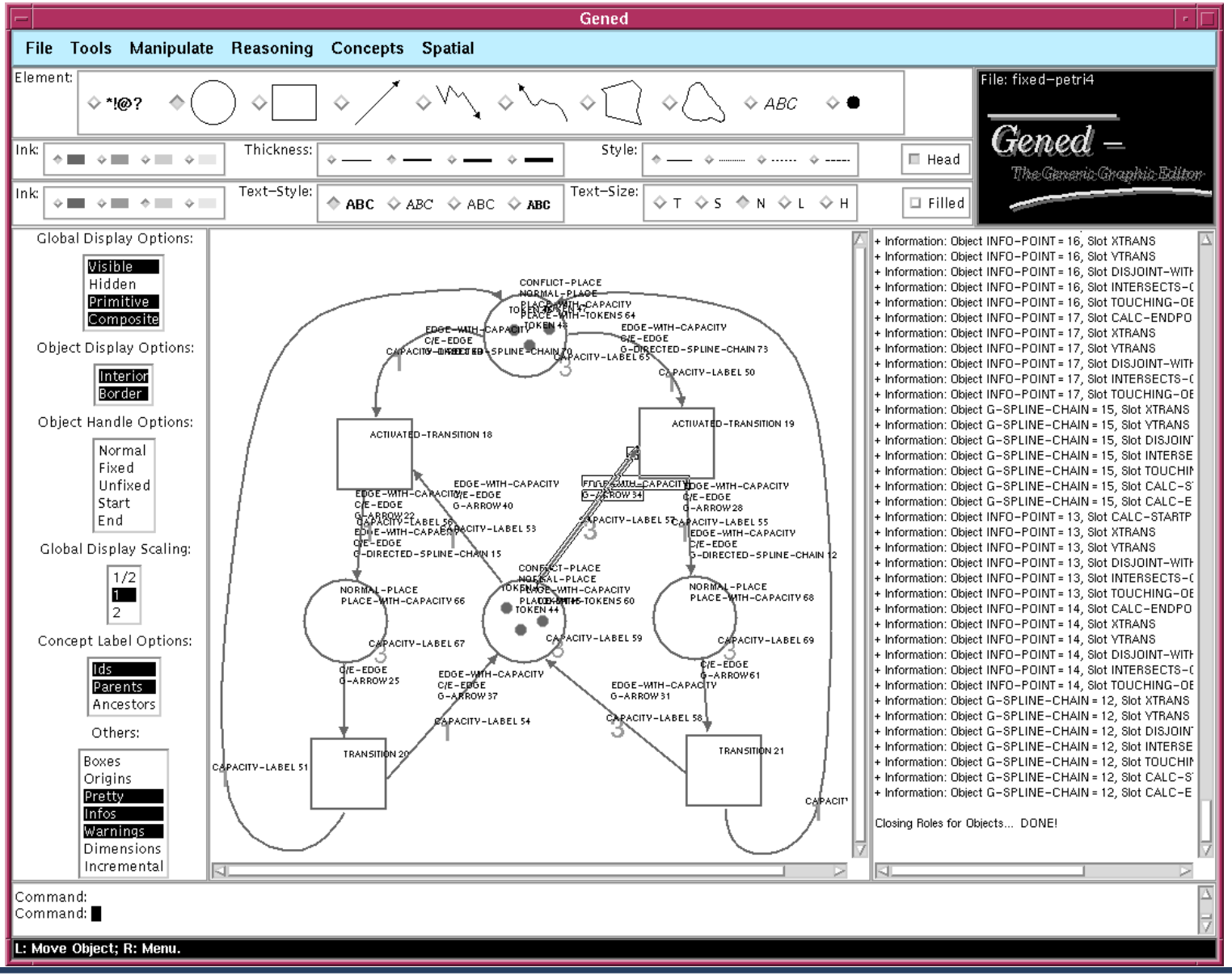
- Individual / ABox realization
  - compute the (most specific) concept names an individual is an instance of, e.g. in
    $\{betty : person, (betty, charles) : has\_child\}$

    it is realized that $betty$ is an instance of $parent$
- Instance checking: is $betty$ and instance of $parent$?
- Role filler checking: is $charles$ a filler (successor) of the has_child role of $betty$?

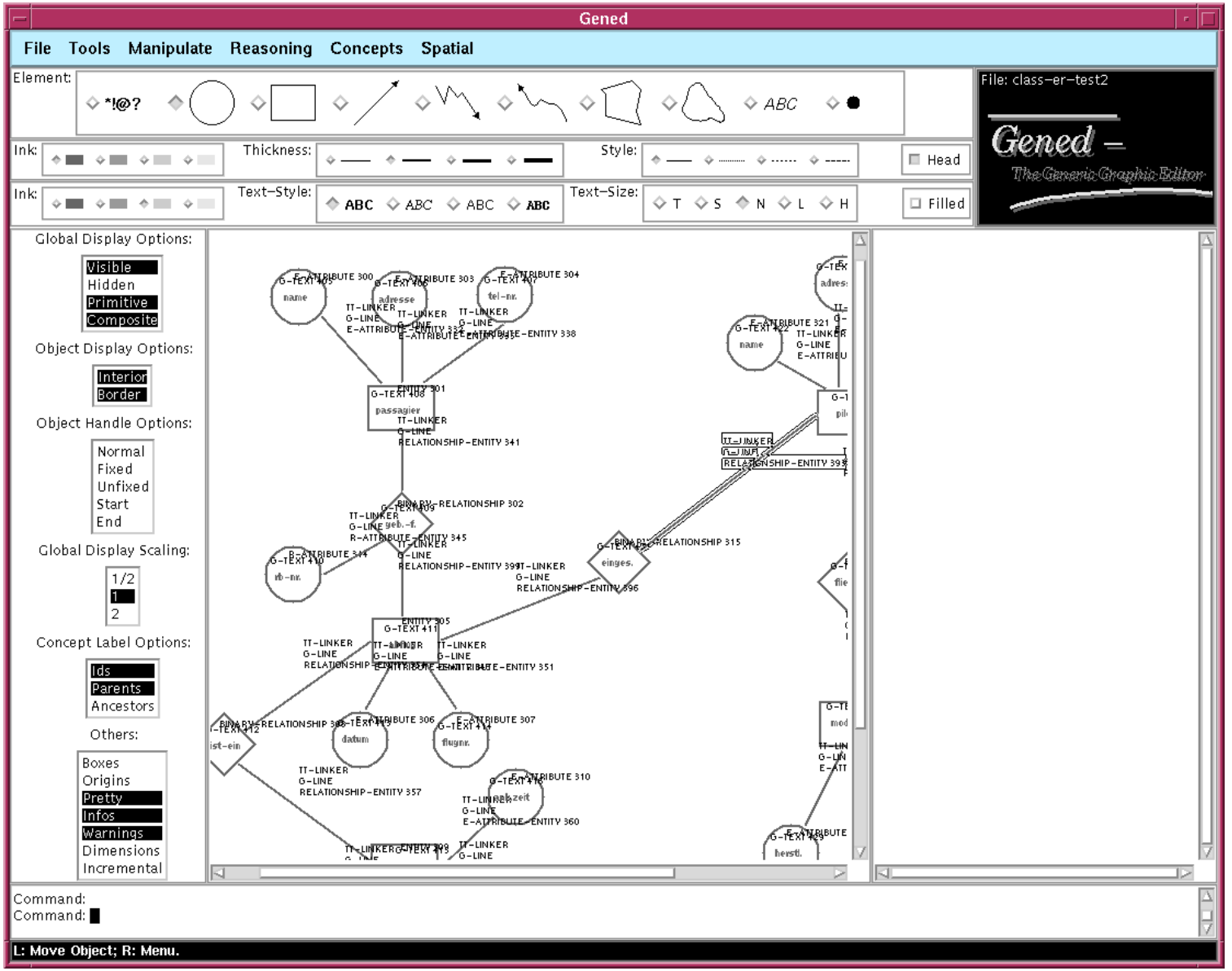
Recognition of Spatial Concepts (1996, Classic DL)



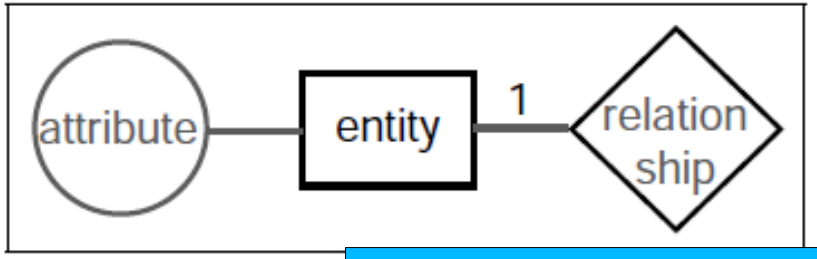
Petri Nets


Gened
File Tools Manipulate Reasoning Concepts Spatial
Element:
Ink:
Thickness:
Style:
Head
Ink:
Text-Style:
Text-Size:
Filled
File: fixed-petri4
Gened –
The Generic Graphic Editor
Global Display Options:
Visible
Hidden
Primitive
Composite
Object Display Options:
Interior
Border
Object Handle Options:
Normal
Fixed
Unfixed
Start
End
Global Display Scaling:
1/2
1
2
Concept Label Options:
Ids
Parents
Ancestors
Others:
Boxes
Origins
Pretty
Infos
Warnings
Dimensions
Incremental
ACTIVATED-TRANSITION 18
ACTIVATED-TRANSITION 19
TRANSITION 20
TRANSITION 21
NORMAL-PLACE
PLACE-WITH-CAPACITY 66
PLACE-WITH-CAPACITY 68
CAPACITY-LABEL 67
CAPACITY-LABEL 59
CAPACITY-LABEL 69
CAPACITY-LABEL 54
CAPACITY-LABEL 58
CAPACITY-LABEL 51
CAPACITY-LABEL 50
CAPACITY-LABEL 63
CAPACITY-LABEL 53
CAPACITY-LABEL 52
CAPACITY-LABEL 55
EDGE-WITH-CAPACITY
C/E-EDGE
G-ARROW 22
G-ARROW 25
G-ARROW 28
G-ARROW 31
G-ARROW 34
G-ARROW 37
G-ARROW 40
G-ARROW 61
G-DIRECTED-SPLINE-CHAIN 15
G-DIRECTED-SPLINE-CHAIN 73
CONFLICT-PLACE
NORMAL-PLACE
TOKEN 44
TOKEN 43
TOKENS 60
TOKENS 64
+ Information: Object INFO-POINT = 16, Slot XTRANS
+ Information: Object INFO-POINT = 16, Slot YTRANS
+ Information: Object INFO-POINT = 16, Slot DISJOINT-WITH
+ Information: Object INFO-POINT = 16, Slot INTERSECTS-C
+ Information: Object INFO-POINT = 16, Slot TOUCHING-OE
+ Information: Object INFO-POINT = 17, Slot CALC-ENDPO
+ Information: Object INFO-POINT = 17, Slot XTRANS
+ Information: Object INFO-POINT = 17, Slot YTRANS
+ Information: Object INFO-POINT = 17, Slot DISJOINT-WITH
+ Information: Object INFO-POINT = 17, Slot INTERSECTS-C
+ Information: Object INFO-POINT = 17, Slot TOUCHING-OE
+ Information: Object G-SPLINE-CHAIN = 15, Slot XTRANS
+ Information: Object G-SPLINE-CHAIN = 15, Slot YTRANS
+ Information: Object G-SPLINE-CHAIN = 15, Slot DISJOINT
+ Information: Object G-SPLINE-CHAIN = 15, Slot INTERSE
+ Information: Object G-SPLINE-CHAIN = 15, Slot TOUCHI
+ Information: Object G-SPLINE-CHAIN = 15, Slot CALC-S
+ Information: Object G-SPLINE-CHAIN = 15, Slot CALC-E
+ Information: Object INFO-POINT = 13, Slot CALC-STARTP
+ Information: Object INFO-POINT = 13, Slot XTRANS
+ Information: Object INFO-POINT = 13, Slot YTRANS
+ Information: Object INFO-POINT = 13, Slot DISJOINT-WITH
+ Information: Object INFO-POINT = 13, Slot INTERSECTS-C
+ Information: Object INFO-POINT = 13, Slot TOUCHING-OE
+ Information: Object INFO-POINT = 14, Slot CALC-ENDPO
+ Information: Object INFO-POINT = 14, Slot XTRANS
+ Information: Object INFO-POINT = 14, Slot YTRANS
+ Information: Object INFO-POINT = 14, Slot DISJOINT-WITH
+ Information: Object INFO-POINT = 14, Slot INTERSECTS-C
+ Information: Object INFO-POINT = 14, Slot TOUCHING-OE
+ Information: Object G-SPLINE-CHAIN = 12, Slot XTRANS
+ Information: Object G-SPLINE-CHAIN = 12, Slot YTRANS
+ Information: Object G-SPLINE-CHAIN = 12, Slot DISJOINT
+ Information: Object G-SPLINE-CHAIN = 12, Slot INTERSE
+ Information: Object G-SPLINE-CHAIN = 12, Slot TOUCHI
+ Information: Object G-SPLINE-CHAIN = 12, Slot CALC-S
+ Information: Object G-SPLINE-CHAIN = 12, Slot CALC-E
Closing Roles for Objects... DONE!
Command:
Command:
L: Move Object; R: Menu.

ER Diagrams


Gened
File Tools Manipulate Reasoning Concepts Spatial
Element:
Ink:
Thickness:
Style:
Head
Text-Style:
Text-Size:
Filled
File: class-er-test2
Gened
The Generic Graphic Editor
Global Display Options:
Visible
Hidden
Primitive
Composite
Object Display Options:
Interior
Border
Object Handle Options:
Normal
Fixed
Unfixed
Start
End
Global Display Scaling:
1/2
1
2
Concept Label Options:
Ids
Parents
Ancestors
Others:
Boxes
Origins
Pretty
Infos
Warnings
Dimensions
Incremental
Command:
Command:
L: Move Object; R: Menu.

Example TBox Definitions – Abox Realization for Recognition



An *entity* is a rectangl
touches at least one rela
some attribute-entity co
least one diamond.

named_region ≡
(*region* ∧ (∃$_{=1}$ containin

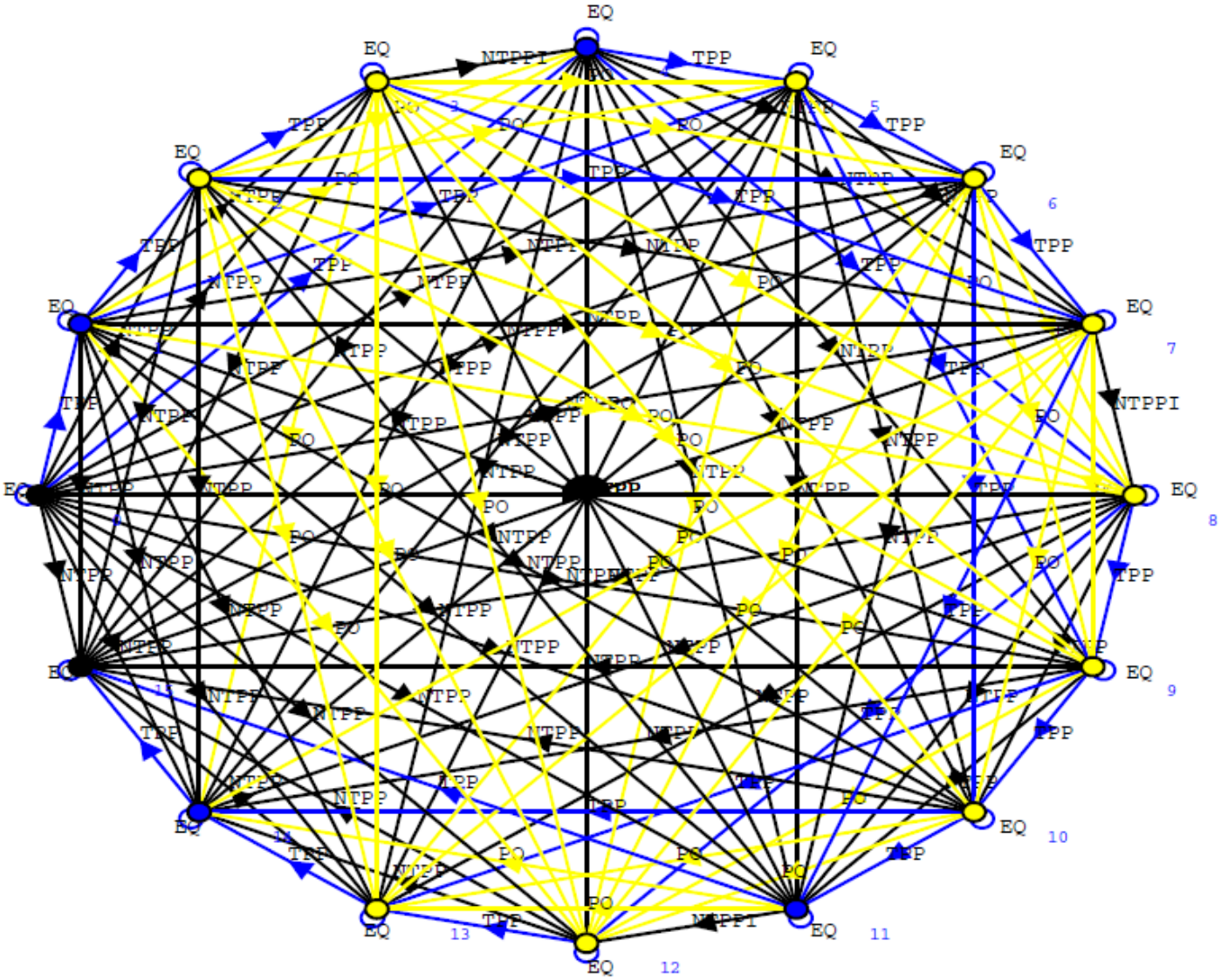
entity ≡
(*rectangle* ∧ named_region ∧
(∃$_{\geq 1}$ touching relationship_entity) ∧
(∀ touching (attribute_entity ∨ relationship_entity)) ∧
(∃$_{\geq 1}$ linked_with *diamond*) ∧
(∀ linked_with (*circle* ∨ *diamond*)))

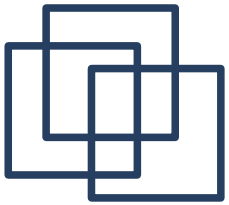
```
(defconcept entity
   (and rectangle
        named-region
        (some touching
              relationship-entity)
        (all touching
             attribute-or-relationship-entity)
        (some linked-with diamond)
        (all linked-with circle-or-diamond)))
```

Quantification over spatial relations

+ ABox for concrete diagram

ABoxes with Spatial Relations





Lessons Learned

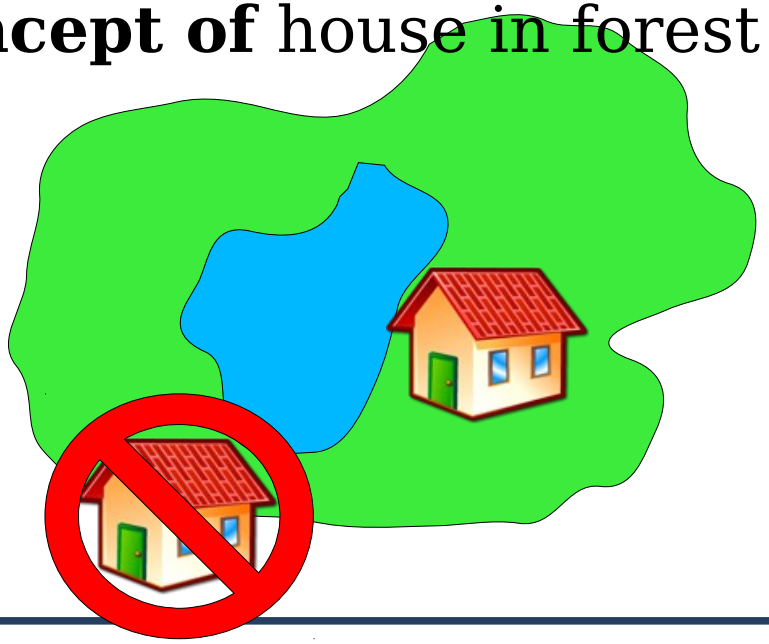
- Inexpressivity of CLASSIC: hacks required (disjunction, qualified subroles, rules for „qualified subroles“, …) → motivation for Racer
- DL problem: we cannot recognize composite aggregates (e.g., whole ER diagrams or petri nets), only „placeholder individuals“
- CLASSIC performance: lots of spatial role relationships (disjoint!) → 40 minutes for realization… → motivation for Racer
- DL problem: open world assumption - forall usually not satisfied
- Criticism - why use DLs at all for that??
  - just used SQL queries or Datalog
  - we don't get the desired subsumption relationships between spatial concepts, as spatial relationships not properly axiomatized

    → work on spatial DLs

DLs with Spatial Semantics

- Example of a desired subsumption relationship
  - Lake strictly contained in forest (NTPP)
  - House at (EC) lake strictly contained in forest not overlapping forest („PO“)

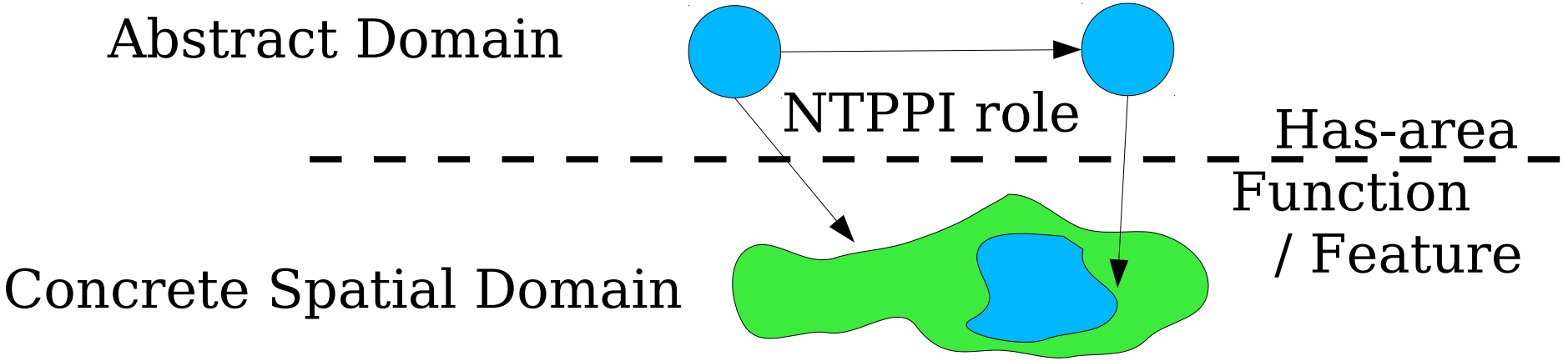
  - House at lake not overlapping forest **is subconcept of** house in forest

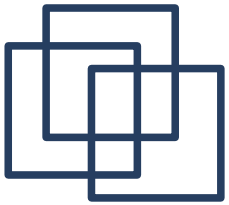


Quantification Over RCC relations

1998 - DLs with Spatial Semantics

- Lutz, Moeller, Haarslev: $\mathcal{ALCRP}(\mathcal{S}_2)$
  - a DL with spatial concrete domains
    - concrete domain objects = spatial regions are connected to abstract domain objects of the DL via functions
    - properties between spatial objects induced roles / relations between corresponding abstract objects
    - the concrete geometry does not need to be known!





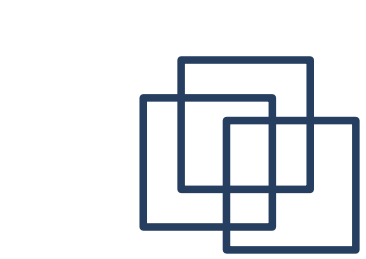
2001 - DLs with Spatial Semantics

- Wessel
  - $\mathcal{ALCI}_{\mathcal{RCC}}$ family
    - DLs with complex role axioms of the form

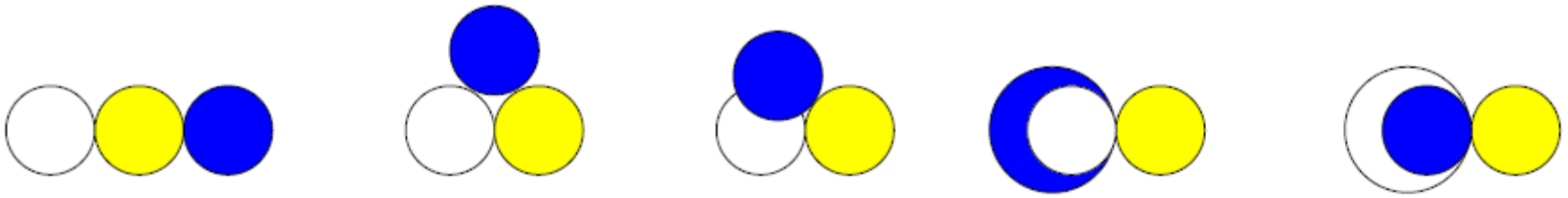
      forall x,y,z : R(x,y) /\ S(y,z) → T1(x,z) \/ T2 …

      (undecidable in general)

    - no concrete domains, but RCC relations axiomatized according to the RCC composition tables
    - some decidable fragments found (Rcc1 … Rcc3)
    - RCC5 and RCC8 undecidable (Lutz + Wolter)



Composition Axioms



$DC(a,c)$ $EC(a,c)$ $PO(a,c)$ $TPP(a,c)$ $TPPI(a,c)$

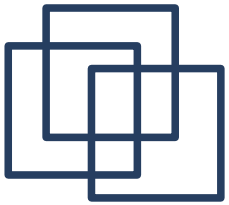
Given $EC(a,b)$, $EC(b,c)$, what do we know about the relationship between a and c? Lookup $EC \circ EC$ in the RCC8 composition-table:

$$\forall x,y,z : EC(x,y) \wedge EC(y,z) \Rightarrow (DC(x,z) \vee EC(x,z) \vee PO(x,z) \vee TPP(x,z) \vee TPPI(x,z))$$

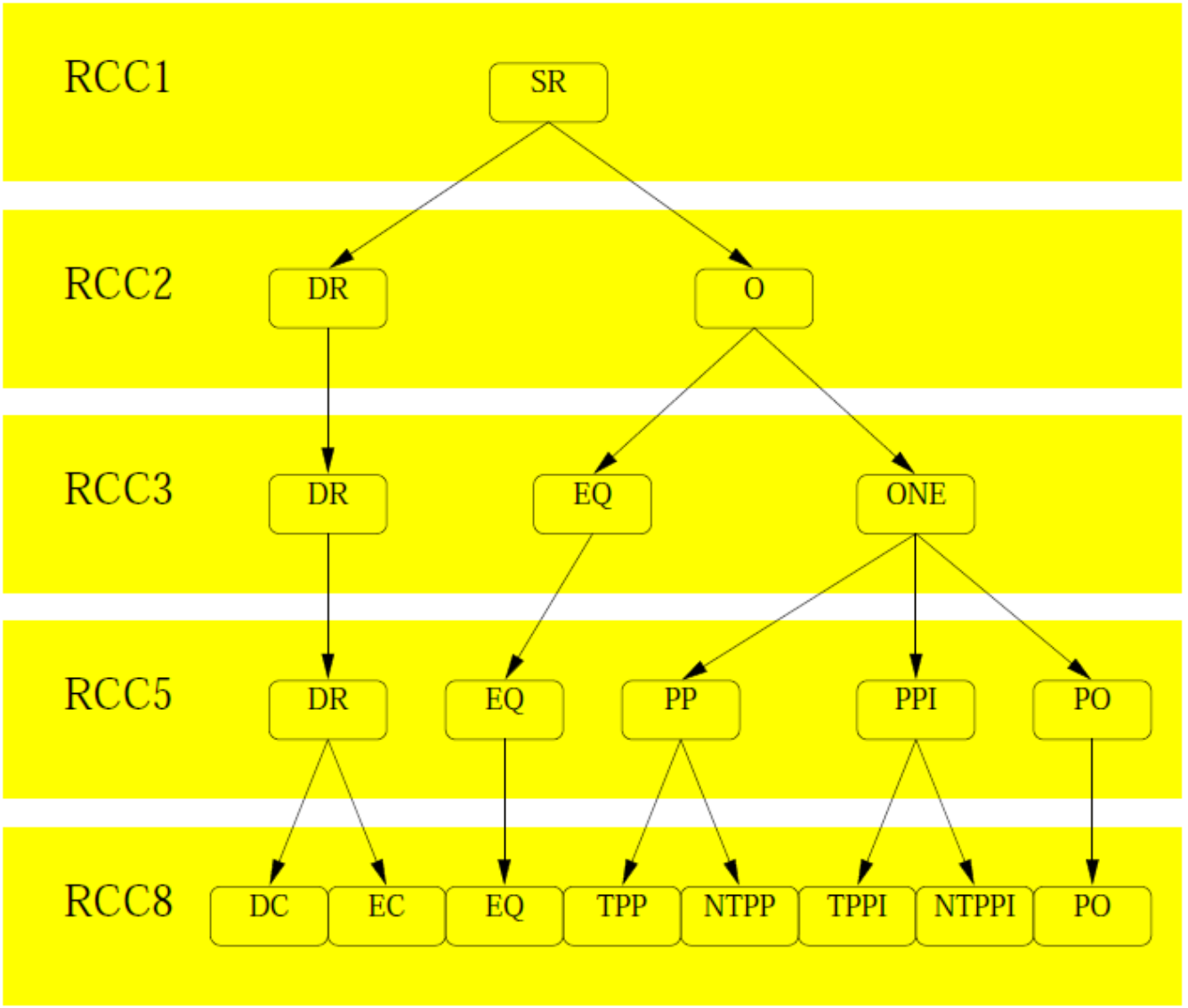
$$EC \circ EC \sqsubseteq DC \sqcup EC \sqcup PO \sqcup TPP \sqcup TPPI$$

RCC8 Composition Table

RCC8	DC(a,b)	EC(a,b)	PO(a,b)	TPP(a,b)	NTPP(a,b)	TPPI(a,b)	NTPPI(a,b)	EQ(a,b)
DC(b,c)	*	DC EC PO TPPI NT- PPI	DC EC PO TPPI NT- PPI	DC	DC	DC EC PO TPPI NT- PPI	DC EC TPPI NT- PPI	DC
EC(b,c)	DC EC PO TPP NTPP	DC EC PO TPP TPPI EQ	DC EC PO TPPI NT- PPI	DC EC	DC	EC PO TPPI NT- PPI	PO TPPI NT- PPI	EC
PO(b,c)	DC EC PO TPP NTPP	DC EC PO TPP NTPP	*	DC EC PO TPP NTPP	DC EC PO TPP NTPP	PO TPPI NT- PPI	PO TPPI NT- PPI	PO
TPP(b,c)	DC EC PO TPP NTPP	EC PO TPP NTPP	PO TPP NTPP	TPP NTPP	NTPP	PO EQ TPP TPPI	PO TPPI NT- PPI	TPP
NTPP(b,c)	DC EC PO TPP NTPP	PO TPP NTPP	PO TPP NTPP	NTPP	NTPP	PO TPP NTPP	PO TPPI TPP NTPP NT- PPI EQ	NTPP
TPPI(b,c)	DC	DC EC	DC EC PO TPPI NT- PPI	DC EC PO TPP TPPI EQ	DC EC PO TPP NTPP	TPPI NT- PPI	NTPPI	TPPI
NTPPI(b,c)	DC	DC	DC EC PO TPPI NT- PPI	DC EC PO TPPI NT- PPI	*	NTPPI	NTPPI	NTPPI
EQ(b,c)	DC	EC	PO	TPP	NTPP	TPPI	NTPPI	EQ

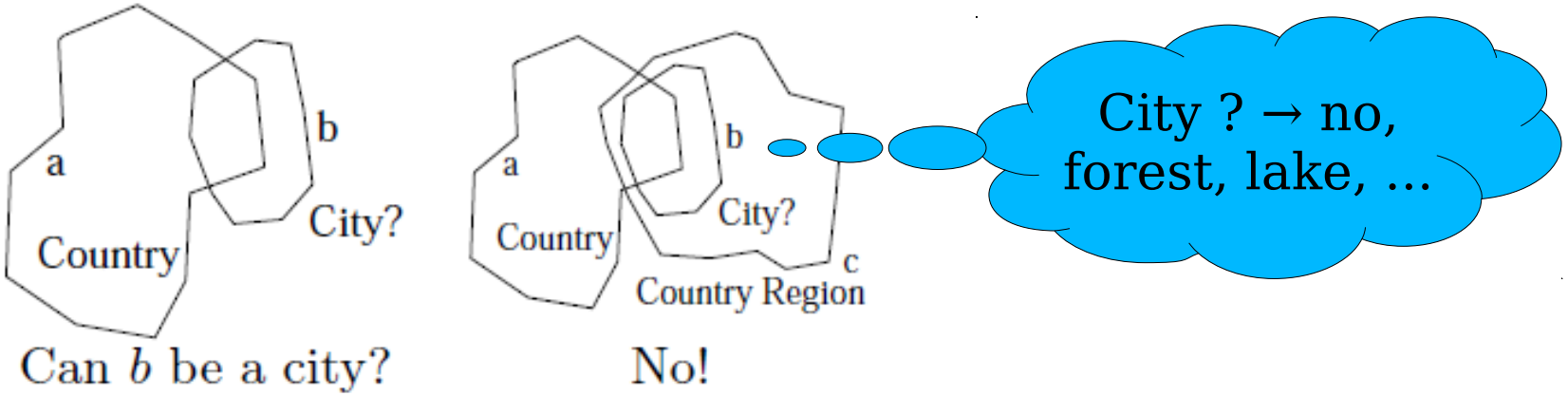


$\mathcal{ALCI}_{\mathcal{RCC}}$ Family

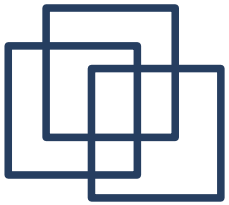


2000: Image Interpretation with Default Rules and $\mathcal{ALCRP}(\mathcal{S}_2)$

- In real life image interpretation applications, „recognition by entailment (e.g., Abox realization)“ does not work
- b is an unknown region – what can it be?



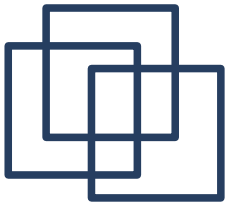
  - each city must be contained in exactly one country
  - countries don't overlap → b cannot be a city
- „Image Interpretation as Model Construction“ (completion)
- or unsound reasoning, e.g., default rules that hypothesize



$\mathcal{ALCRP}(\mathcal{S}_2)$ KB and Default Rules

$$
\begin{aligned}
country &\doteq country\_region \sqcap \forall contains.\neg country\_region \sqcap \\
&\quad \forall overlaps.\neg country\_region \sqcap \forall inside.\neg country\_region \\
city &\doteq city\_region \sqcap \exists inside.country\_region \\
lake &\dot{\sqsubseteq} lake\_region \\
river &\doteq river\_region \sqcap \forall overlaps.\neg lake\_region \sqcap \forall inside.\neg lake\_region
\end{aligned}
$$

$$
d_1 = \frac{area \ : \ city}{city}, \quad d_2 = \frac{area \ : \ lake}{lake}, \quad d_3 = \frac{area \ : \ country}{country}
$$

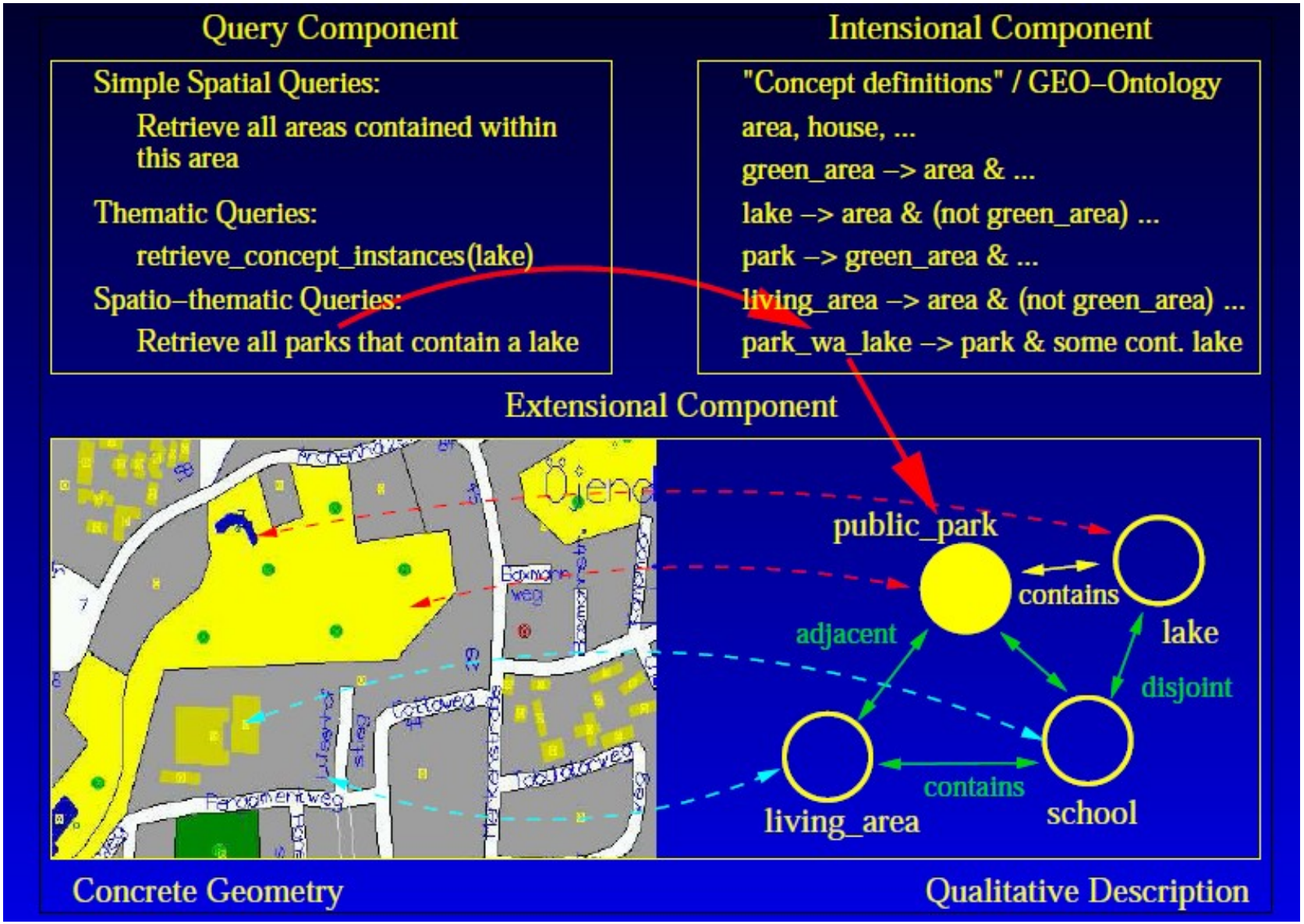


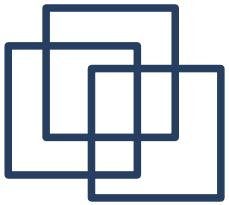
2003 - Deductive GIS with DLs

- Lessons learned:
  - spatial reasoning on the intensional (Tbox) level too difficult if quantification over spatial relations permitted …
  - … or to inexpressive

  → work on the extensional level instead ;-)
  - many interesting problems on the extensional level too
  - large Aboxes – many regions, many relations → use Racer (by then, Racer had a reputation of dealing well with large ABoxes)
  - to recognize spatial constellations: expressive Abox query language (→ nRQL)
    - spatio-thematic complex queries
    - hybrid system with, „on thy fly evaluation“ of QSR from geometry, tightly integrated into a DL reasoner, …

DLMAPS – Hybrid Architecture



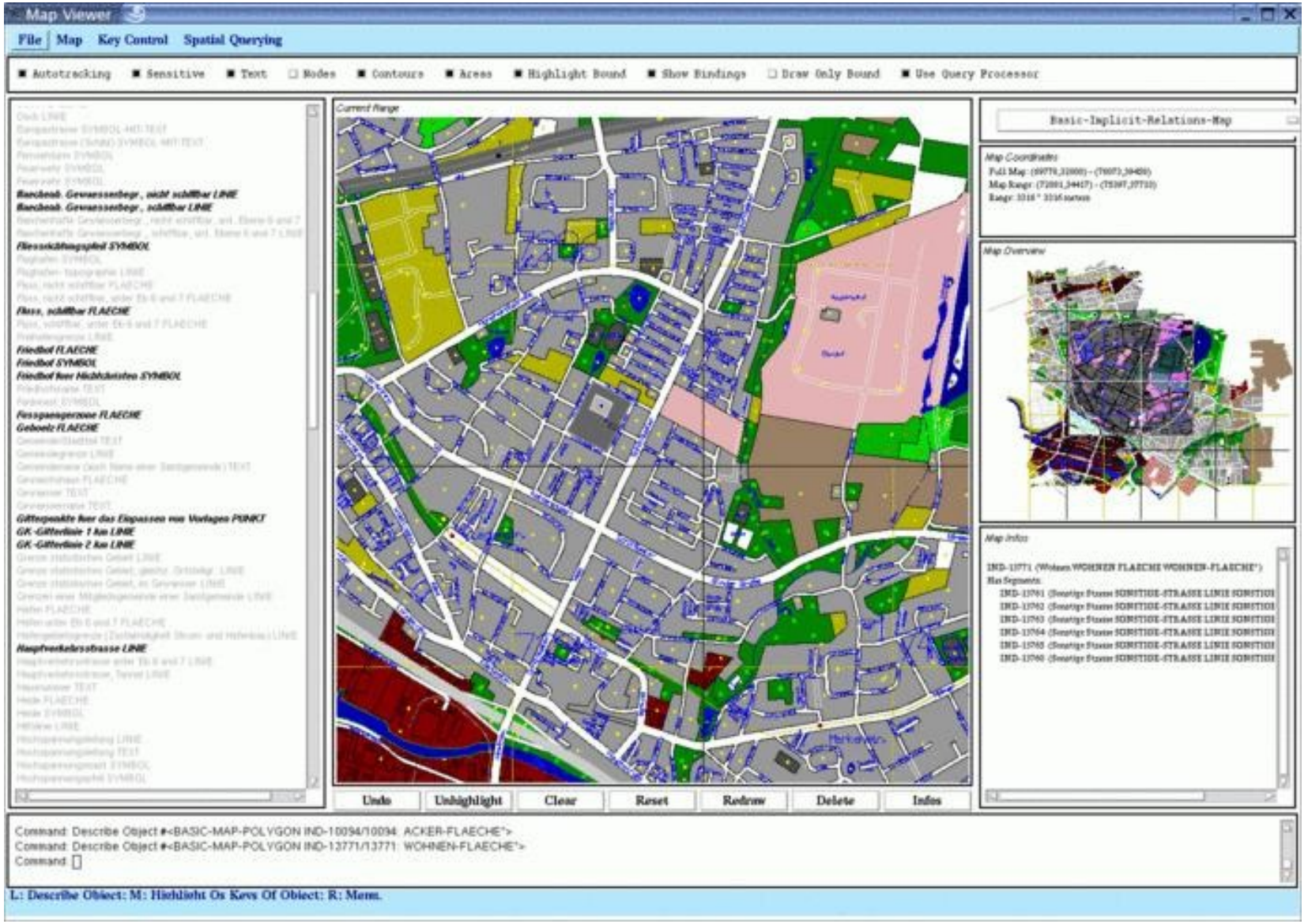


Spatio-Thematic Hybrid Queries

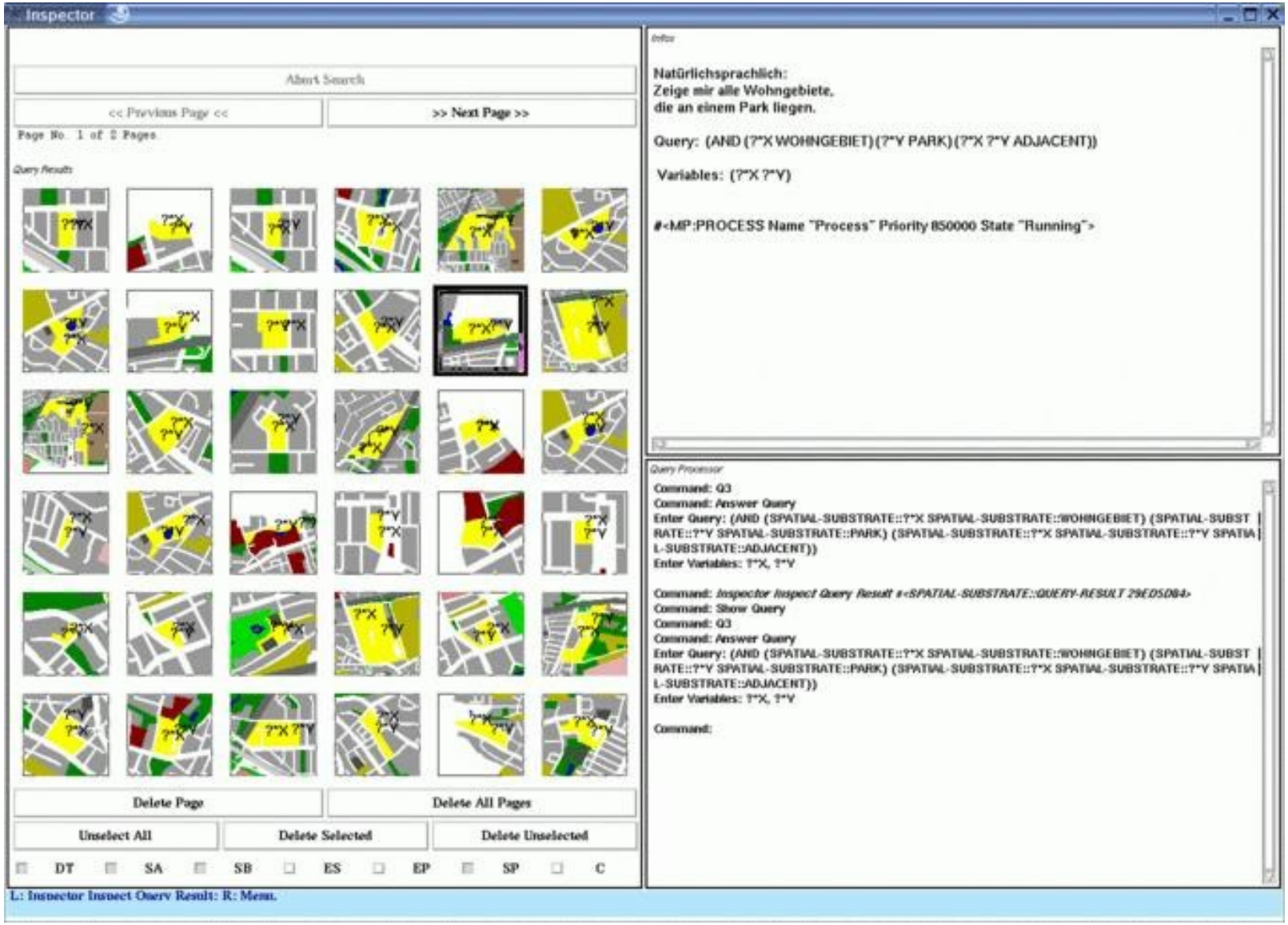
```
(retrieve (?*x ?*y)
      (and (?*x
                (and living-area
                     (all living-quality
                          first-class-area)))
           (?y ?x (:inside-distance 750))
           (?*y subway-station)
           (?x ?y :adjacent)
           (?*y golf-club)
           (?y (:area 10000000 nil))))
```

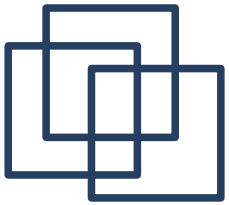
- hybrid: ?*x DL query atom, ?x spatial atom
- thematic, qualitative + quantiative spatial aspects
- composition-table based RCC reasoning
- cost-based optimizer
  (reused for „Abox-based“ conjunctive queries in AURA)

DLMAPS – Map Viewer



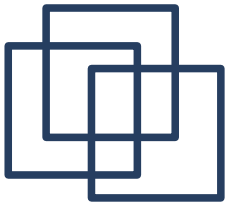
DLMAPS – Query Answering



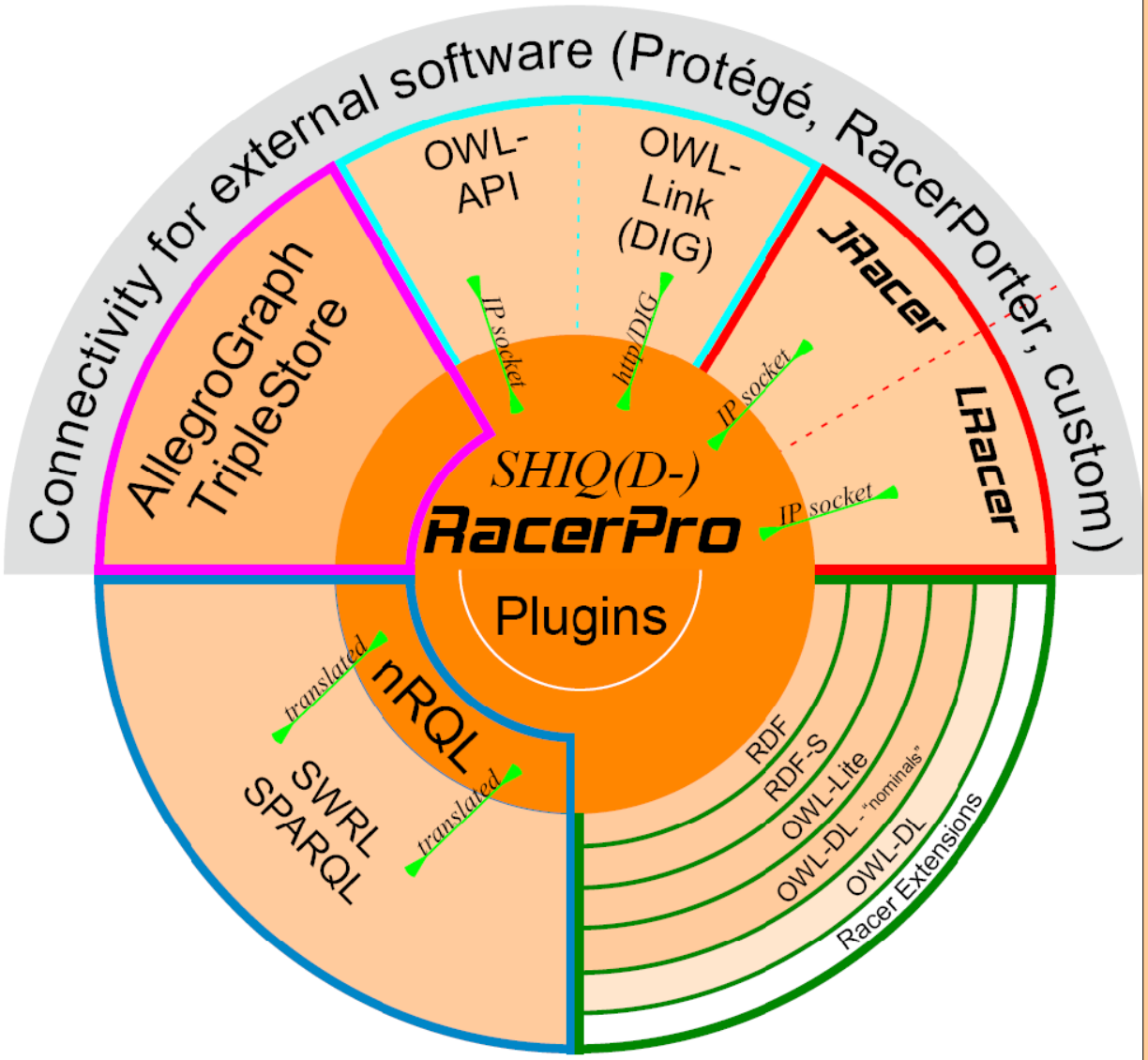


Racer Systems – Partners & Friends

October 2004 – today



RacerPro - Architecture & History



- Started as **Racer** at the University of Hamburg in **1998**, project of **Volker Haarslev & Ralf Möller**
- First description logic (DL) reasoner „of the new generation“ of highly optimized DL systems **with ABox (individuals, relations)**
- One of the **first OWL DL (-)** systems (**2002**), DL $\mathcal{SHIQ}(D)$
- Commercial offspring **RacerPro** by **Racer Systems** (2004 - today)
- Expressive query language **nRQL**
- First DL system that could give **complete answers** to the **LUBM Benchmark** queries (2004)
- First DL system with **inference-aware SWRL & SPARQL**
- Main memory-based
- Recently: Integrated **AllegroGraph**
- Some **special-purpose representations and reasoning**
- **Free for education & research**

How Do I Get RacerPro ?

- www.racer-systems.com

  there is the 2.0 preview version

  - no license required
  - to be finalized soon

- Latest activities
  - Protege 4.3 plugin (Ulm / Derivo GmbH)
  - EL consequence-based reasoning
  - OntoLisp framework

Solving Sudokus by Reasoning

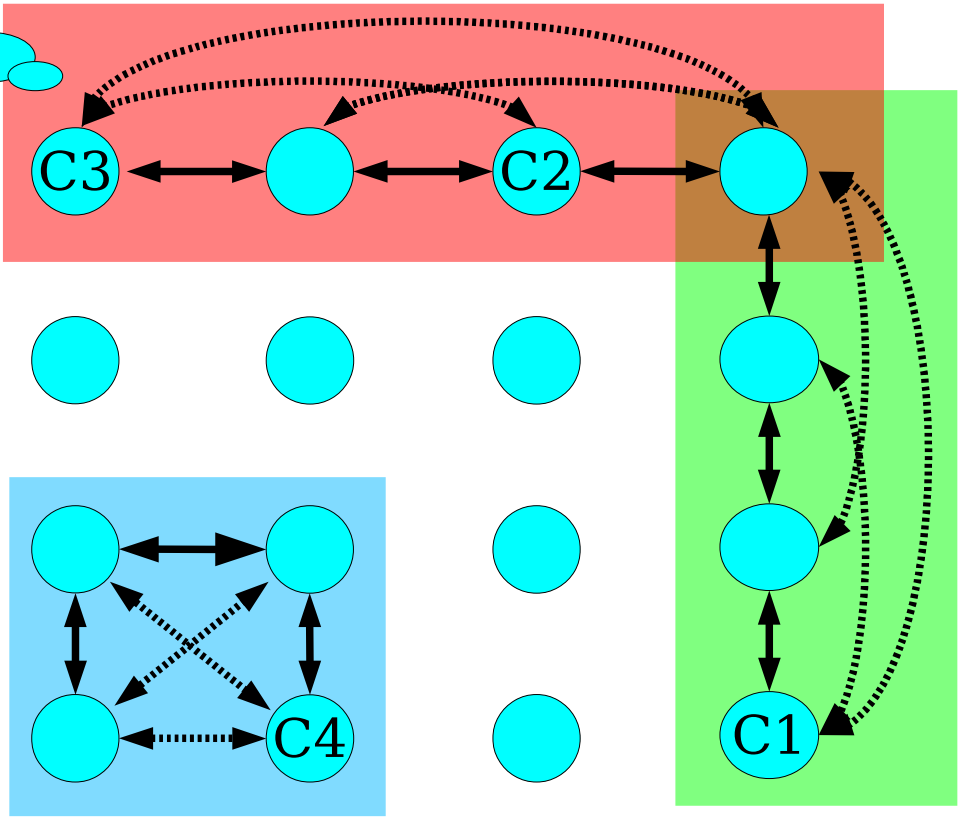
Create a KB whose logical models represent all possible Sudoku solutions. A good Sudoku has only ONE solution → entailed facts = solution!

pairwise_disjoint(C_1, C_2, C_3, C_4)

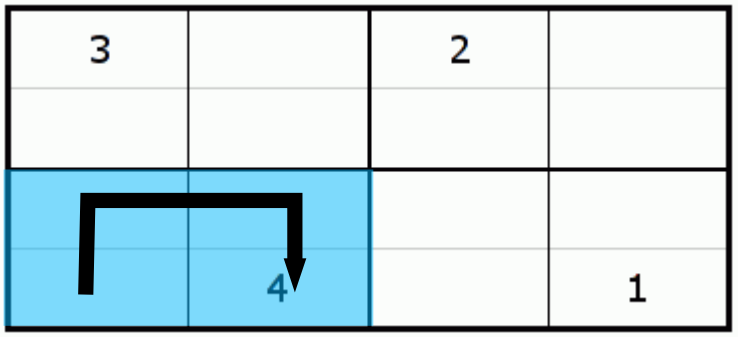
$$\top \doteq (C_1 \sqcup C_2 \sqcup C_3 \sqcup C_4) \sqcap$$
$$(C_1 \rightarrow \forall R.\neg C_1) \sqcap (C_2 \rightarrow \forall R.\neg C_2) \sqcap$$
$$(C_3 \rightarrow \forall R.\neg C_3) \sqcap (C_4 \rightarrow \forall R.\neg C_4) \sqcap$$
$$\ldots$$

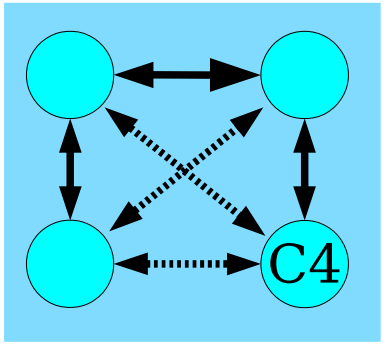
3		2	
	4		1

3		2	
	4		1



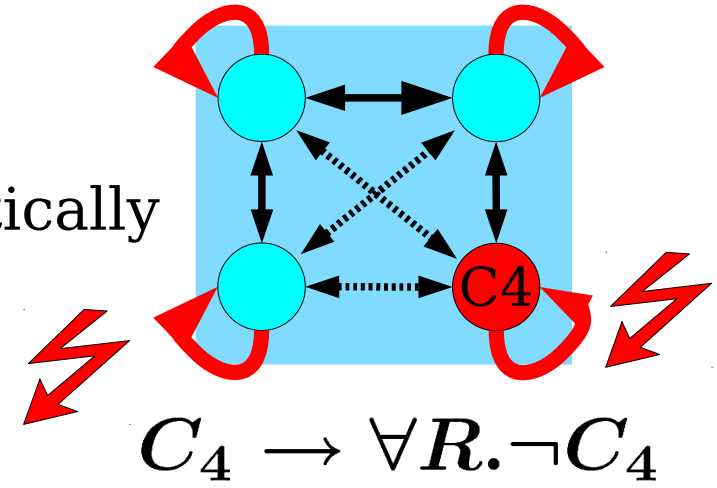
Sudoku – ABox Construction



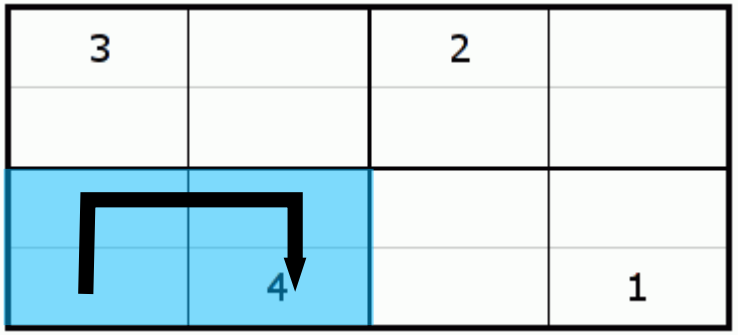


ABox construction

- by hand? OK for 4x4, but for 9x9?
  → create the structure programmatically (MiniLisp)
- transitive & symmetric property →
  - use different backward property instead of a symmetric property
  - quantification over common parent property
  - different props. for different rows, columns, diagonals



Sudoku – Relational Structure



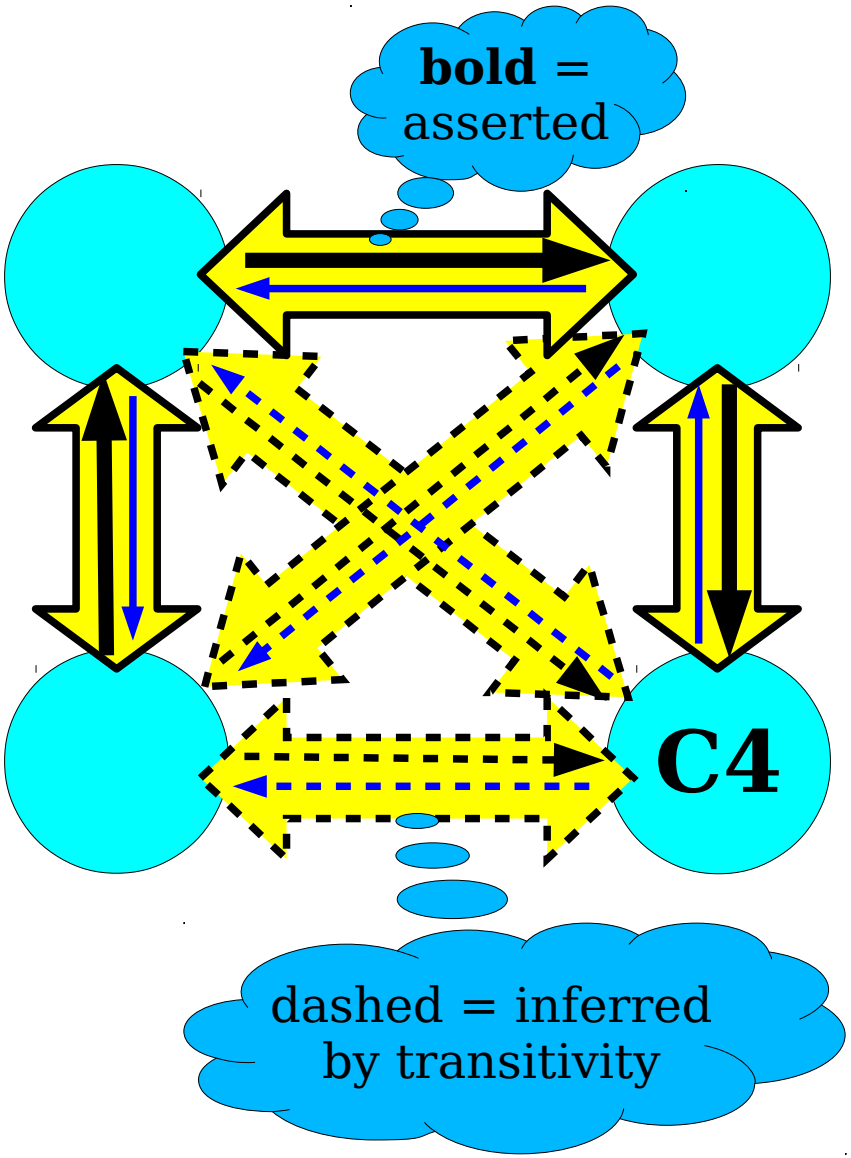
$Q_1 \dot{\sqsubseteq} R$

$Q_2 \dot{\sqsubseteq} R$

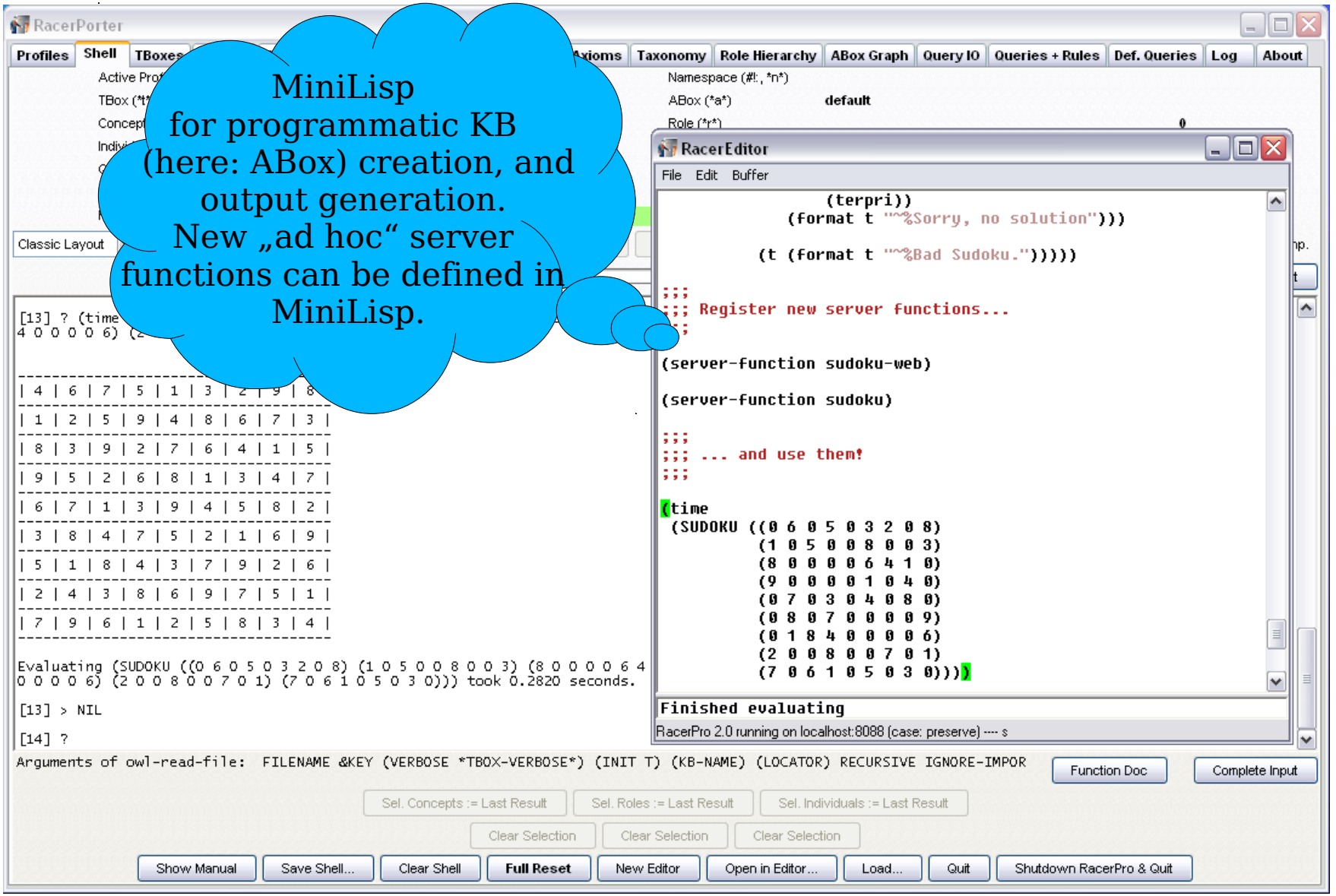
$\mathsf{transitive}(Q_1)$

$\mathsf{transitive}(Q_2)$

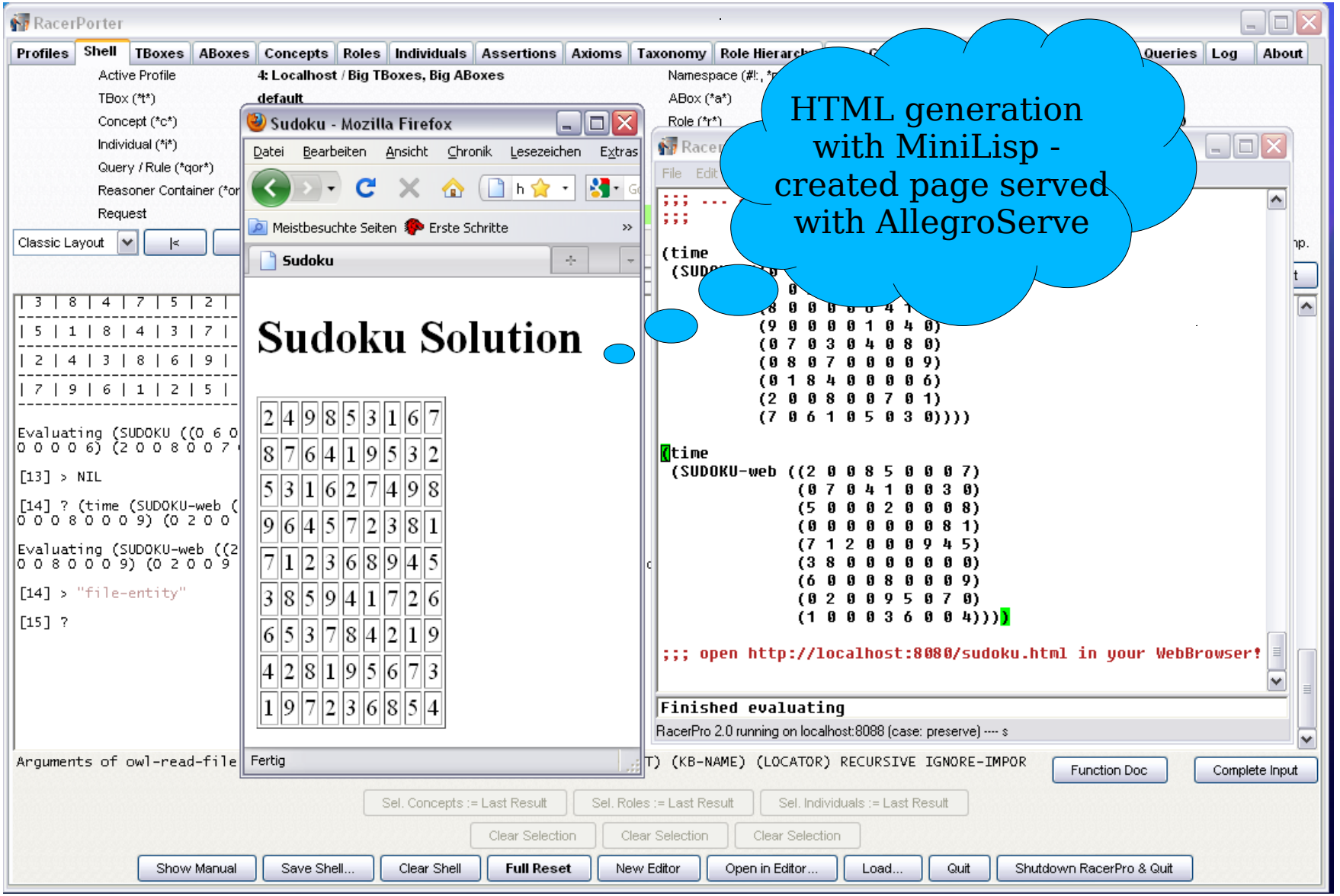
$Q_1(x, y) \leftrightarrow Q_2(y, x)$

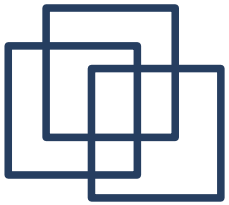


Solving Sudokus by Reasoning



MiniLisp HTML Generation

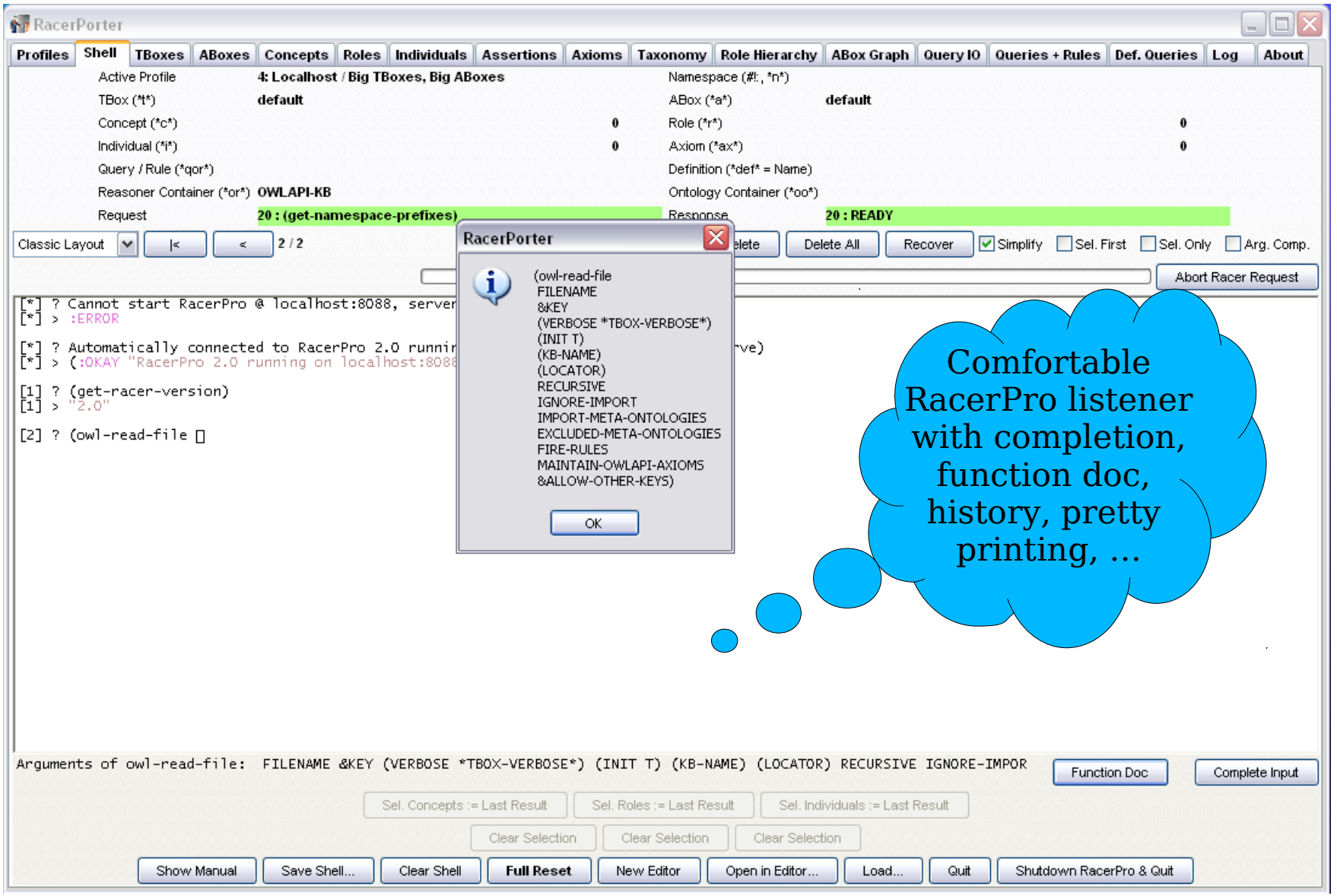




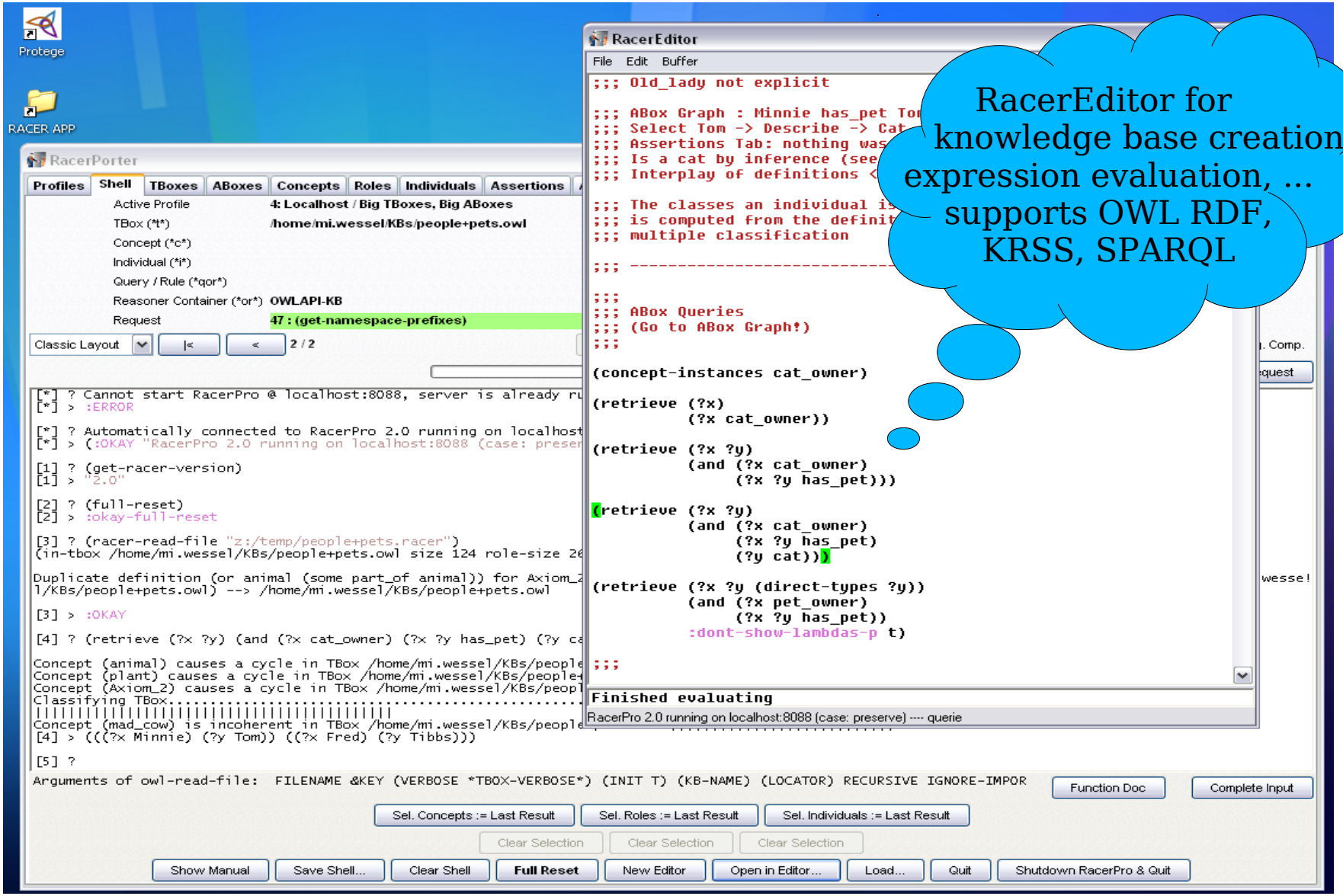
Demos

- Sudoku
- People + Pets.owl Protege and RacerPorter
- JRacer + Eclipse
- LRacer...

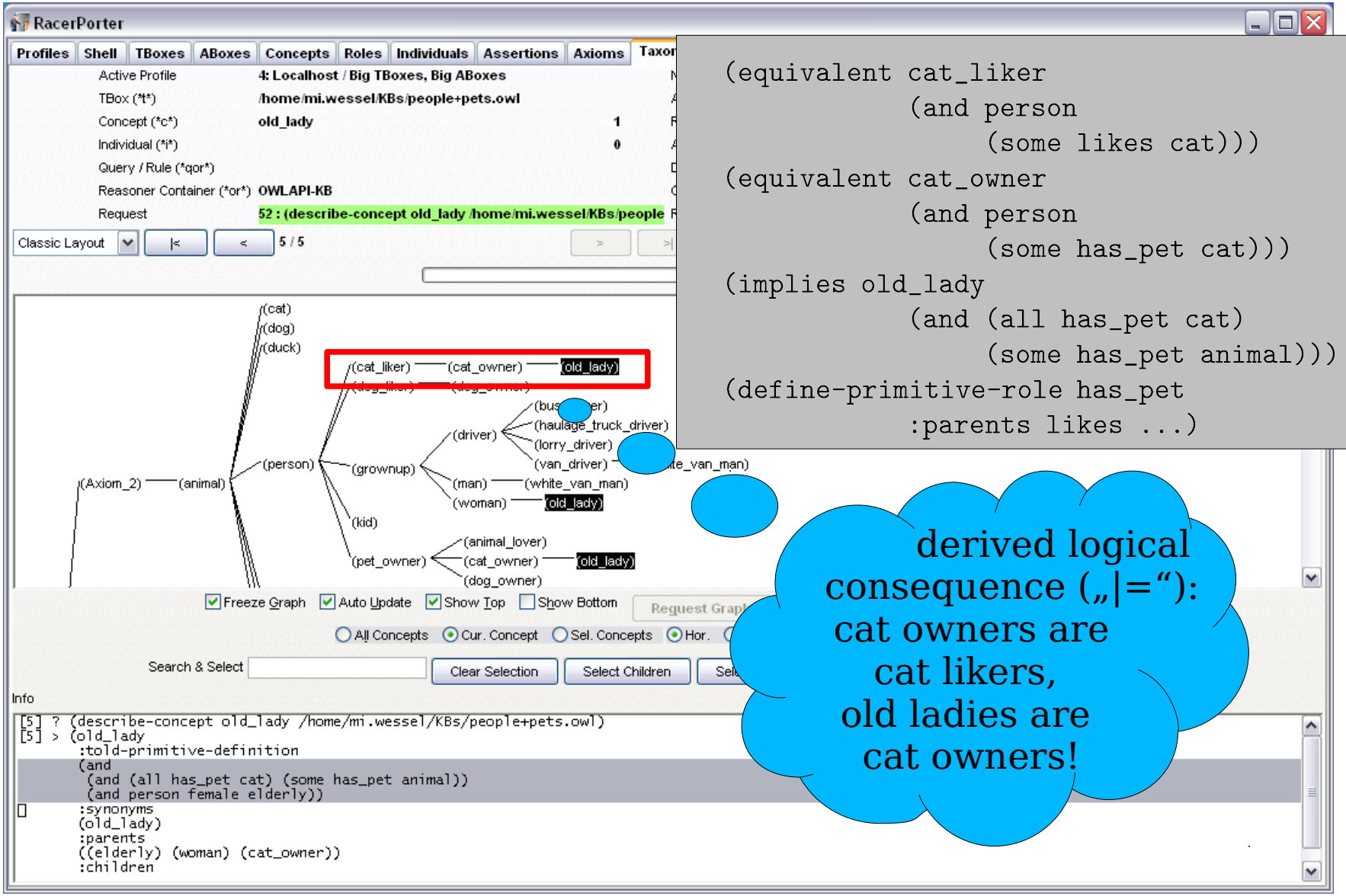
(2006 – today) RacerPorter – The Listener



The Racer Editor with Some Example Queries

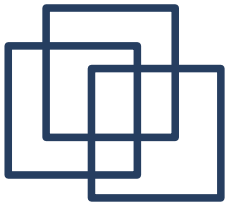


The Class (Concept) Hierarchy („Taxonomy“)




```
(equivalent cat_liker
            (and person
                 (some likes cat)))
(equivalent cat_owner
            (and person
                 (some has_pet cat)))
(implies old_lady
            (and (all has_pet cat)
                 (some has_pet animal)))
(define-primitive-role has_pet
            :parents likes ...)
```

derived logical consequence („|=“): cat owners are cat likers, old ladies are cat owners!



The Relation (Role / Property) Hierarchy

RacerPorter

Profiles | Shell | TBoxes | ABoxes | Concepts | Roles | Individuals | Assertions | Axioms | Taxor

Active Profile	4: Localhost / Big TBoxes, Big ABoxes	
TBox (*t*)	/home/mi.wessel/KBs/people+pets.owl	
Concept (*c*)	cat	4
Individual (*i*)	Tom	1
Query / Rule (*qor*)		
Reasoner Container (*or*)	OWLAPI-KB	
Request	63 : (describe-role has_pet /home/mi.wessel/KBs/people+pet	

Classic Layout | |< | < | 13 / 13 | > | >|

(top-role)
- (likes) — (has_pet)
- (eats)
- (drives)
- (works_for)
- (reads)
- (is_pet_of)
- (has_parent) — (has_father), (has_mother)
- (has_part)
- (has_child)
- (eaten_by)
- (service_number)

Freeze G… | Print Graph | Search & Select | Parents | Synonyms

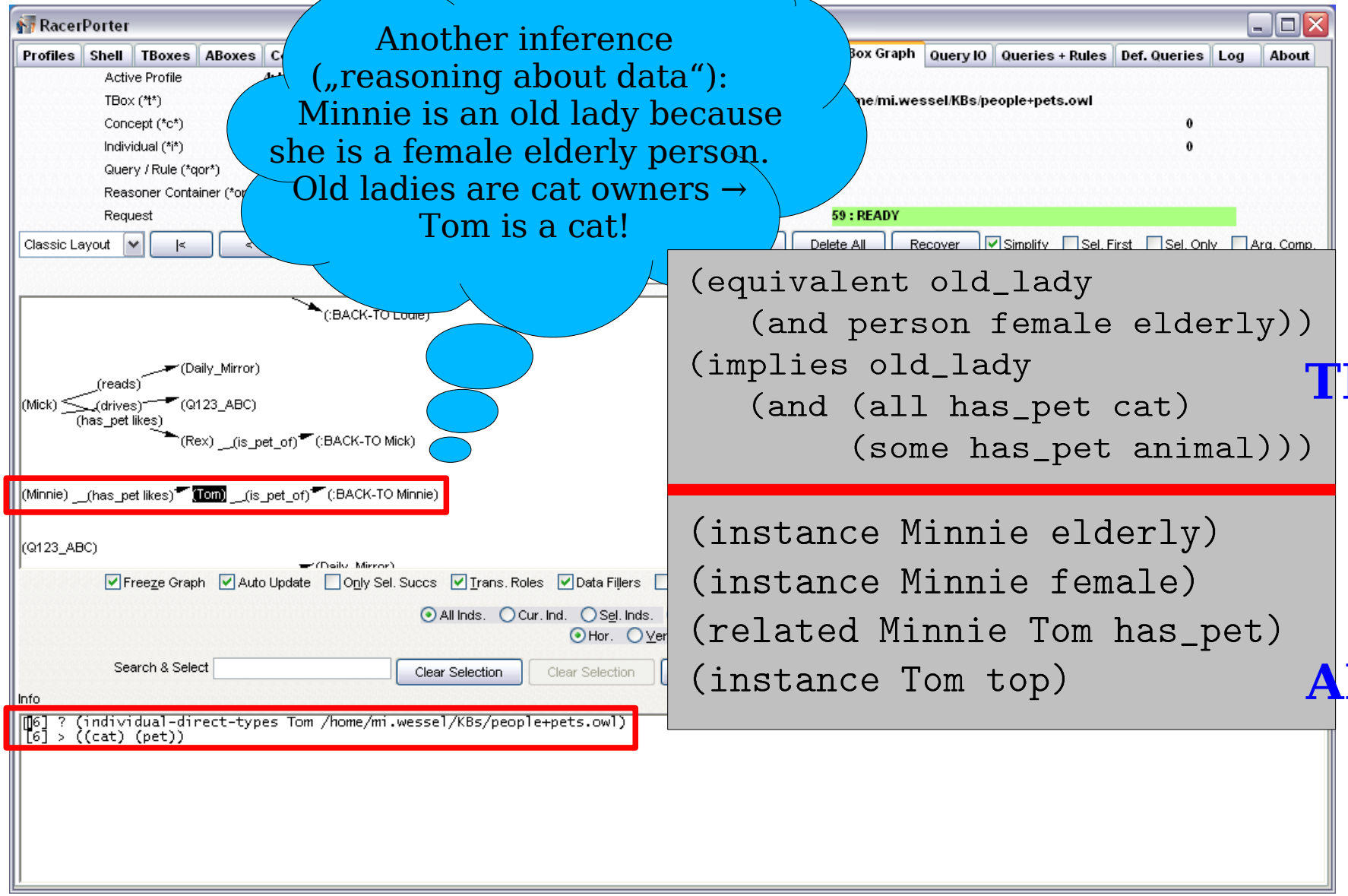
Info

```
[7] ? (describe-role has_pet /home/mi.wessel/KBs/…
[7] > (has_pet
       :synonyms
       (has_pet)
       :domain
       person
       :range
       animal
       :parents
       (likes)
       :children
```

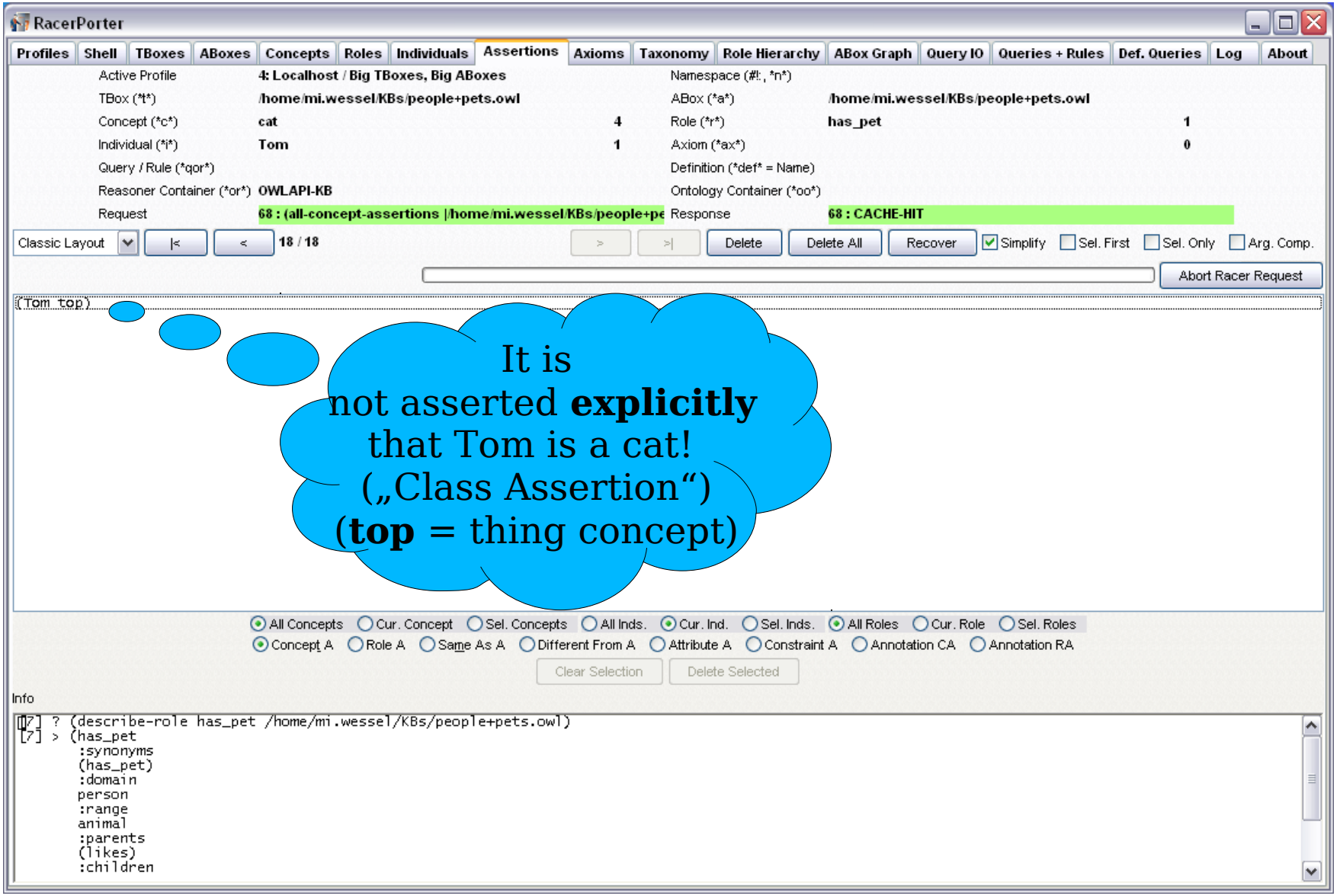
```
(equivalent cat_liker
            (and person
                 (some likes cat)))
(equivalent cat_owner
            (and person
                 (some has_pet cat)))
(implies old_lady
            (and (all has_pet cat)
                 (some has_pet animal)))
(define-primitive-role has_pet
            :parents likes ...)
```

Relation hierarchy - „has pet“ is a subrelation of „likes“ (having a pet implies that you like it)

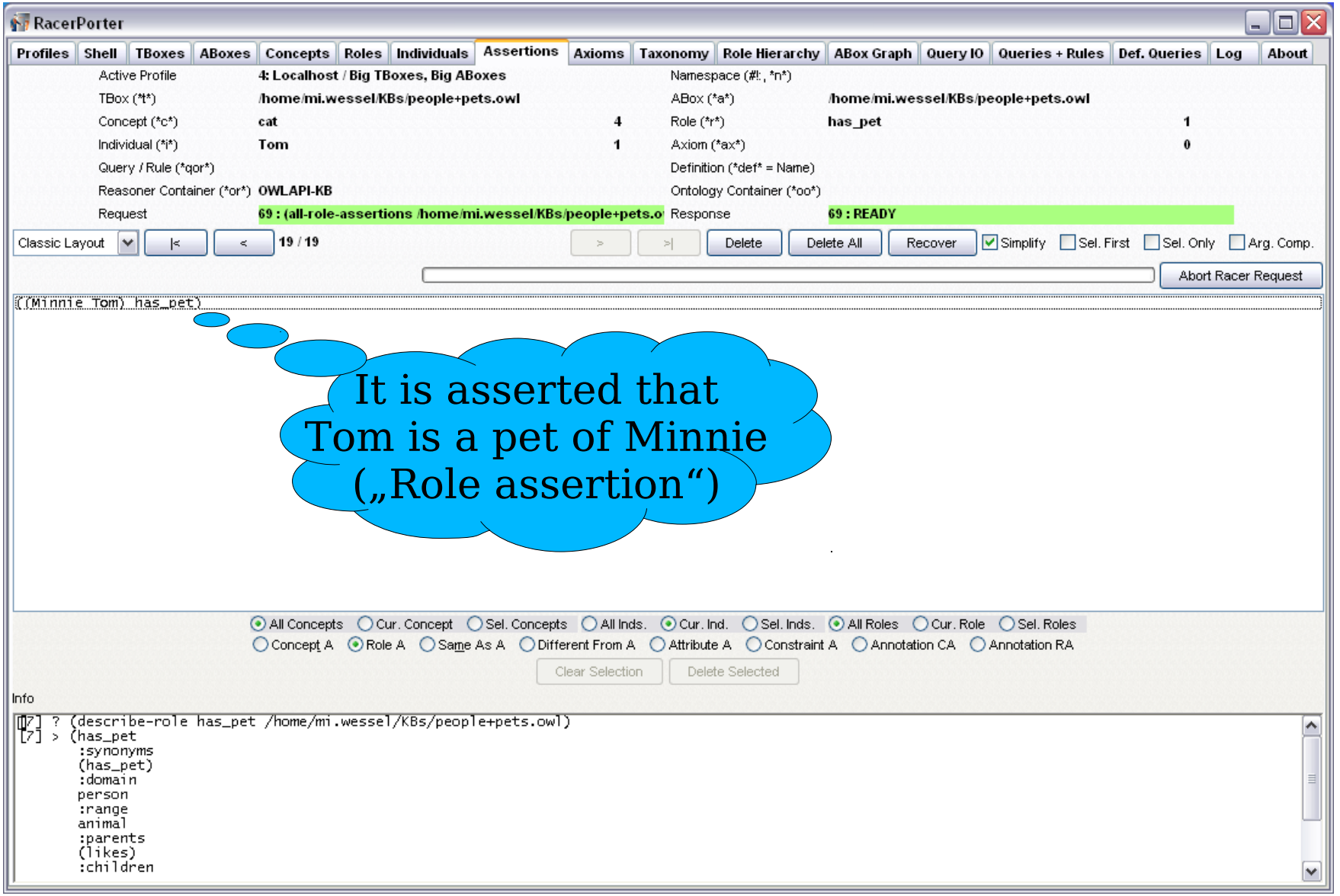
Individuals & Relationships – ABox Graph

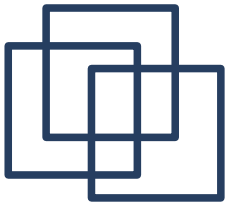


Inspecting Class Assertions for Tom



Relation („Role“) Assertions for Tom





Syntaxes

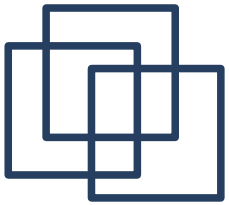
Racer has its own parsers

- Old lady concept in…
  - **KRSS / Racer native:**

```
(equivalent old_lady (and person female elderly))
(implies    old_lady (and (all has_pet cat)
                          (some has_pet animal)))
```

  - **OWL 2 Functional Syntax** (almost S-Expressions…)

```
EquivalentClasses(
    old+lady
    ObjectIntersectionOf(female person elderly))
SubClassOf(old+lady
    ObjectIntersectionOf(
        ObjectAllValuesFrom(has_pet cat)
        ObjectSomeValuesFrom(has_pet animal)))
```



Syntaxes (2)

- **Old lady** concept in **OWL RDF/XML:**

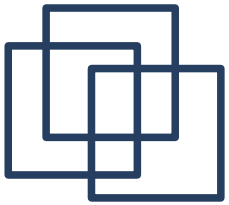
```
<owl:Class rdf:ID="old_lady">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf
        rdf:parseType="Collection">
            <owl:Class rdf:about="#person"/>
            <owl:Class rdf:about="#female"/>
            <owl:Class rdf:about="#elderly"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
...
```

(equivalent old_lady...

... (and ...

... person female elderly ...

...))



Syntaxes (3)

- **Old lady** concept in **OWL RDF/XML** conti...

```
...
<rdfs:subClassOf>
  <owl:Class>
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Restriction>
        <owl:allValuesFrom rdf:resource="#cat"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#has_pet"/>
        </owl:onProperty>
      </owl:Restriction>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#has_pet"/>
        </owl:onProperty>
        <owl:someValuesFrom rdf:resource="#animal"/>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
</rdfs:subClassOf>
</owl:Class>
```

(implies old_lady ...

... (and ...

... (all has_pet cat) ...

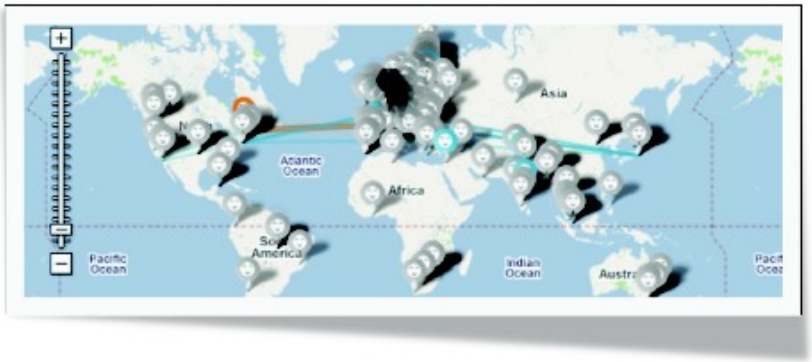
... (some has_pet cat) ...

DOCOMO ContextWatcher / IYouIt

www.iyouit.eu

iyouit

share life blog play



A mobile community service

- 1300+ users in 60+ countries
- Connected to emerging Web 2.0 services

Your digital life recorder

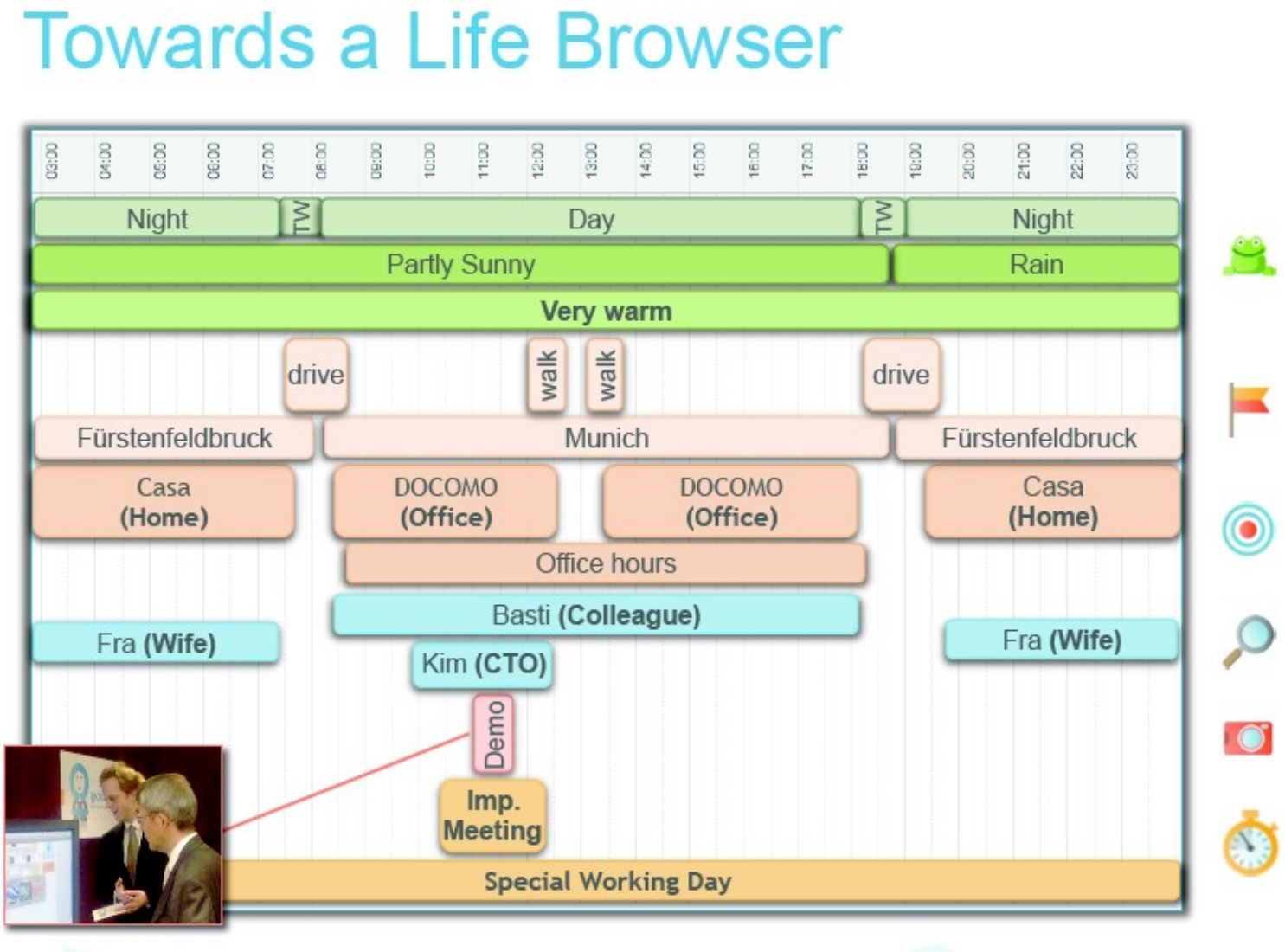
- Facilitating context awareness on standard phones
- Integration of key Semantic Web technologies

flickr

YouTube

twitter

DOPPLR


Towards a Life Browser
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Night
TW
Day
TW
Night
Partly Sunny
Rain
Very warm
drive
walk
walk
drive
Fürstenfeldbruck
Munich
Fürstenfeldbruck
Casa (Home)
DOCOMO (Office)
DOCOMO (Office)
Casa (Home)
Office hours
Basti (Colleague)
Fra (Wife)
Kim (CTO)
Fra (Wife)
Demo
Imp. Meeting
Special Working Day

Regonized Events for „Life Logging“ Blogging

ContextWatcher Blogs in 2005

FRIDAY, NOVEMBER 11, 2005

A busy ISWC day

Today was a busy ISWC day (84.1% covered). I took 9 pictures in Dublin and Galway.

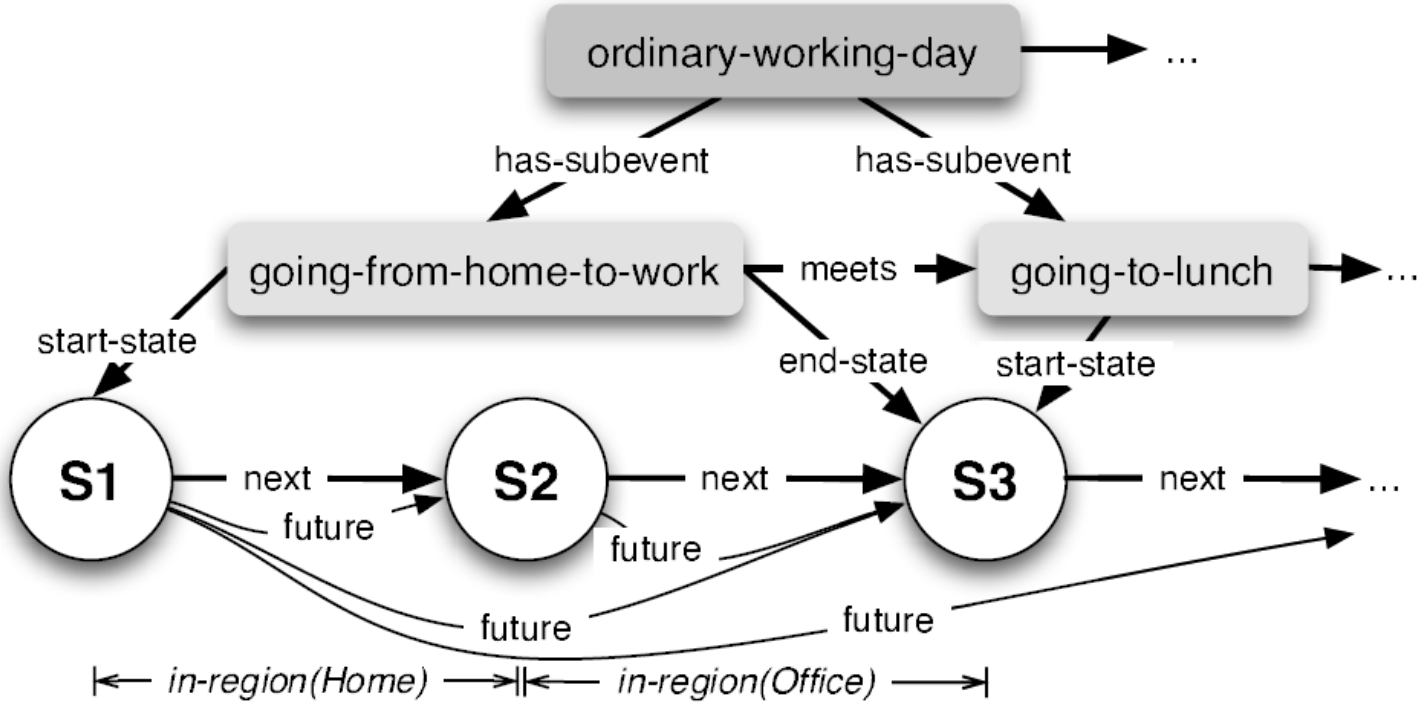



Today was the last day of my business trip to the ISWC'05 conference in Galway together with my colleague M. Luther. It was a cold and rainy day. In the afternoon I traveled back to Munich via Dublin by plane.

I visited Galway (51.8%), München (7.3%), Dublin (18.9%) and Offaly (9.9%), mainly Commute (5.2%) and ISWC (35.1%). I met luther (38.7%). My maximum speed was 131.2 km/h.

POSTED BY MATTHIAS AT 1:01 PM 0 COMMENTS

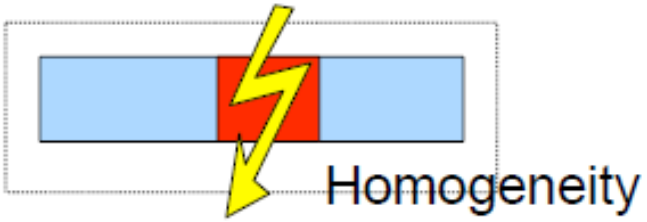
Situations, Simple & Complex Events

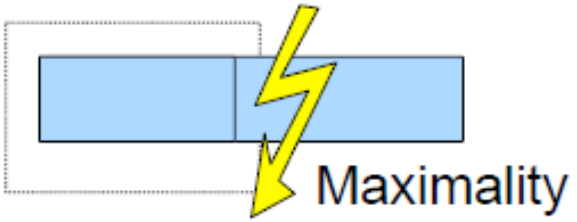


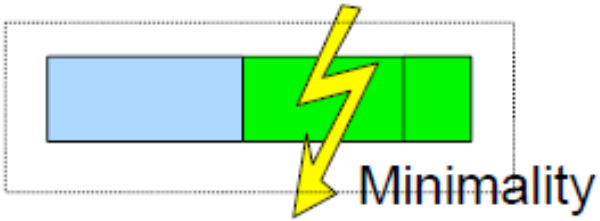
- States = situations → simple events → complex events
  - Linear time model (**next** role, transitive superrole **future**)
  - Allen temporal relation via concrete domain reasoning
  - Event constructions (aggregates!) via forward chaining rules

Realization of Event *Recognizers*

- Assuming events are already present as aggregates in the ABox, how to recognize them?
  - Defined concepts
    - Problems with relational expressivity for complex events
      - only tree-shaped temporal constraints expressible
      - Important event properties cannot be expressed (required for definitions of complex events!)



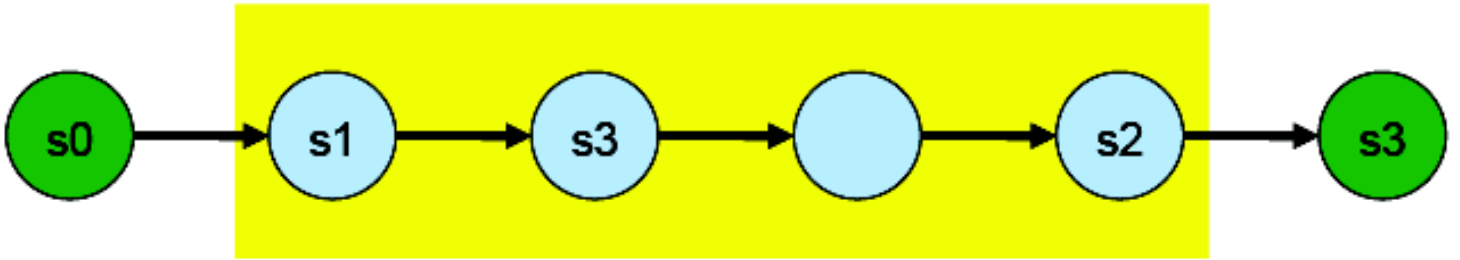




  - Queries and rules
    - High relational expressivity over the ABox
    - Universal closed-domain quantifier (SPARQL, SQL, nRQL)

Recognizing Homogeneous Events (1)

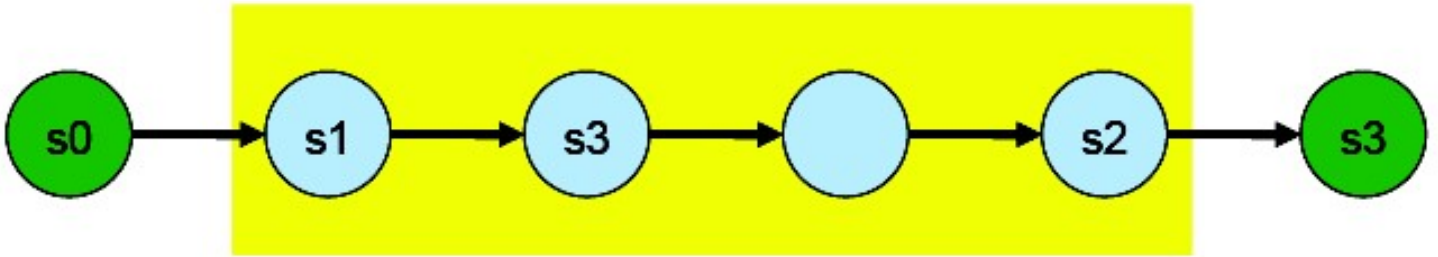
$$
\begin{aligned}
ans(s_1, s_2) \leftarrow \quad & state(s_1), state(s_2), future(s_1, s_2), \\
& P(s_1), P(s_2), \\
& \backslash\pi(s_1)\ (state(s_0), next(s_0, s_1), P(s_0)), \\
& \backslash\pi(s_2)\ (state(s_3), next(s_2, s_3), P(s_3)), \\
& \backslash\pi(s_1, s_2)\ (\ \ state(s_3), future(s_1, s_3), future(s_3, s_2), \\
& \qquad\qquad\qquad\qquad \backslash P(s_3))
\end{aligned}
$$




```
(retrieve (?s1 ?s2)
  (and (?s1 state) (?s2 state) (?s1 ?s2 future)
       (?s1 P) (?s2 P)
       (neg (project-to (?s1) (and (?s0 state) (?s0 ?s1 next) (?so P)))
       (neg (project-to (?s2) (and (?s3 state) (?s2 ?s3 next) (?s3 P)))
       (neg (project-to (?s1 ?s2) (and (?s3 state) (?s1 ?s3 future) (?s3 ?s2 future)
                                       (neg (?s3 P))))))
```

Recognizing Homogeneous Events (2)

$$\mathcal{S}_{\mathcal{A}}=(\Delta^{\mathcal{I}}, C^{\mathcal{I}}, ..., R^{\mathcal{I}}, ...), \text{ with}$$
$$\Delta^{\mathcal{I}} = \mathsf{inds}(\mathcal{A}),$$
$$C^{\mathcal{I}} = \{\, i \mid\ i \in \mathsf{inds}(\mathcal{A}), \mathcal{A} \models C(i) \,\} \ ,$$
$$R^{\mathcal{I}} = \{\, (i,j) \mid\ \mathsf{inds}(\mathcal{A}), \mathcal{A} \models R(i,j) \,\},$$



$$\{\, (s_1, s_2) \mid\ \exists s_1, s_2 : \ state(s_1) \wedge state(s_1) \wedge future(s_1, s_2) \wedge$$
$$P(s_1) \wedge P(s_2) \wedge$$
$$\neg\exists s_0 : \ state(s_0) \wedge next(s_0, s_1) \ \wedge P(s_0) \wedge$$
$$\neg\exists s_3 : \ state(s_3) \wedge next(s_2, s_3) \ \wedge P(s_3) \wedge$$
$$\neg\exists s_3 : \ state(s_3) \wedge future(s_1, s_3), future(s_3, s_2) \wedge$$
$$\neg P(s_3) \,\}$$

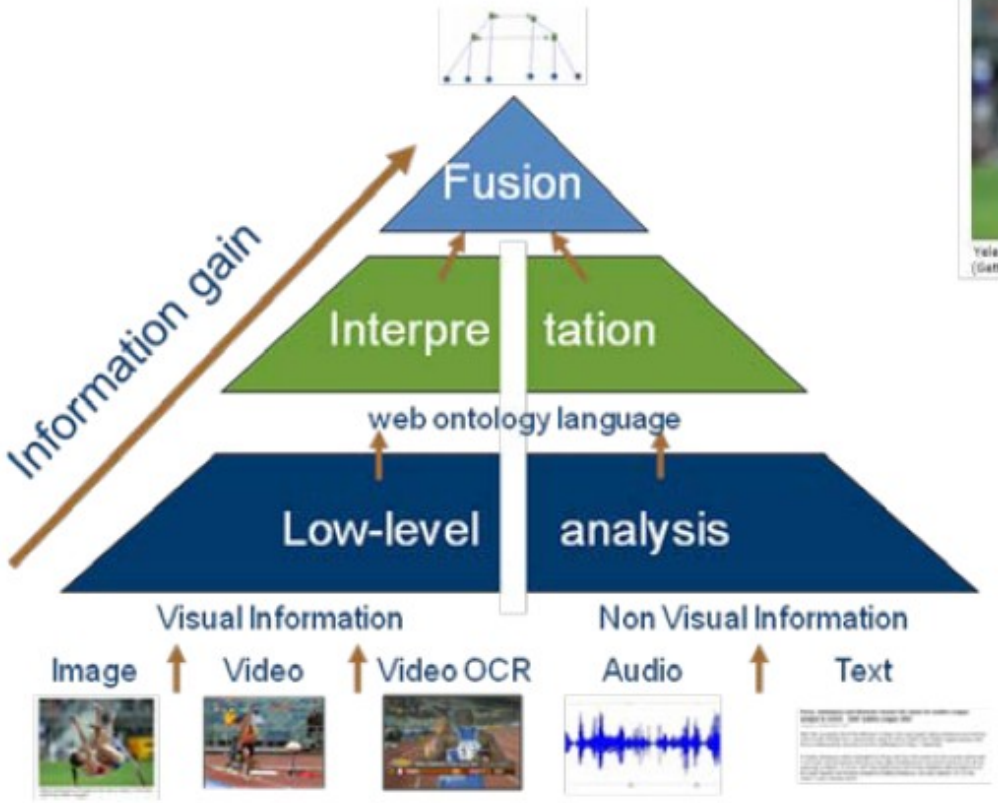
BOEMIE Project – Interprete / Understand Athletics News Pages

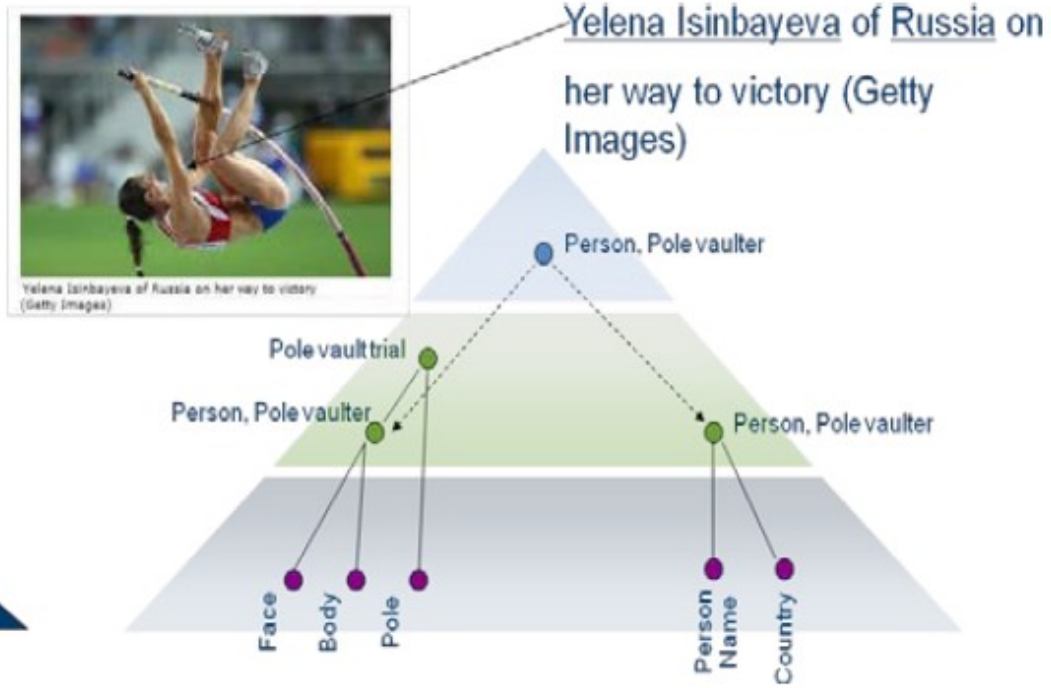
Goal:

use clues from different modalities (text, image, video, sound) and named entity recognition + background KB with athletics knowledge to interprete. Hypothesize and integrate (fusion). Label the paragraphs appropriately („text understanding“).

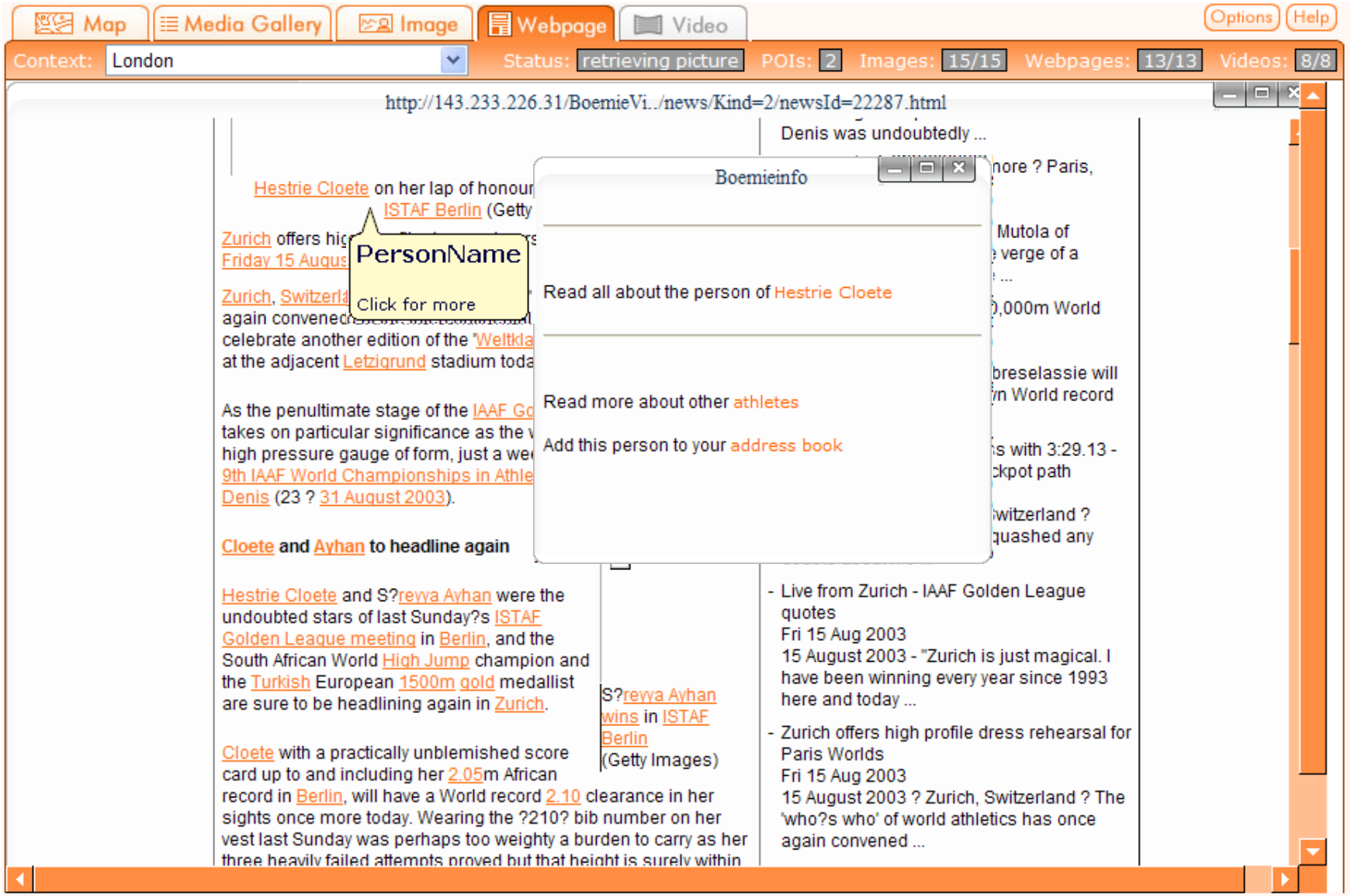
Jerry Hobbs

BOEMIE Project – „DIKW“ Pyramid



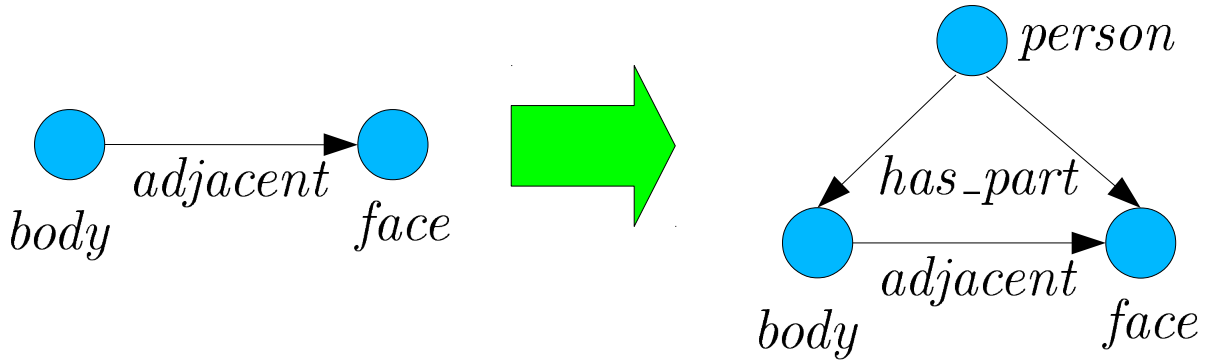


BOEMIE Project – Text Modality



Interpretation with OWL / SWRL?

- The low-level annotations are incomplete
  - no person instance, but face & body region adjacency
  - often, aggregates must be instantiated, which is challenging



$$body \sqcap \exists adjacent.face \dot{\sqsubseteq} \exists part\_of.person$$

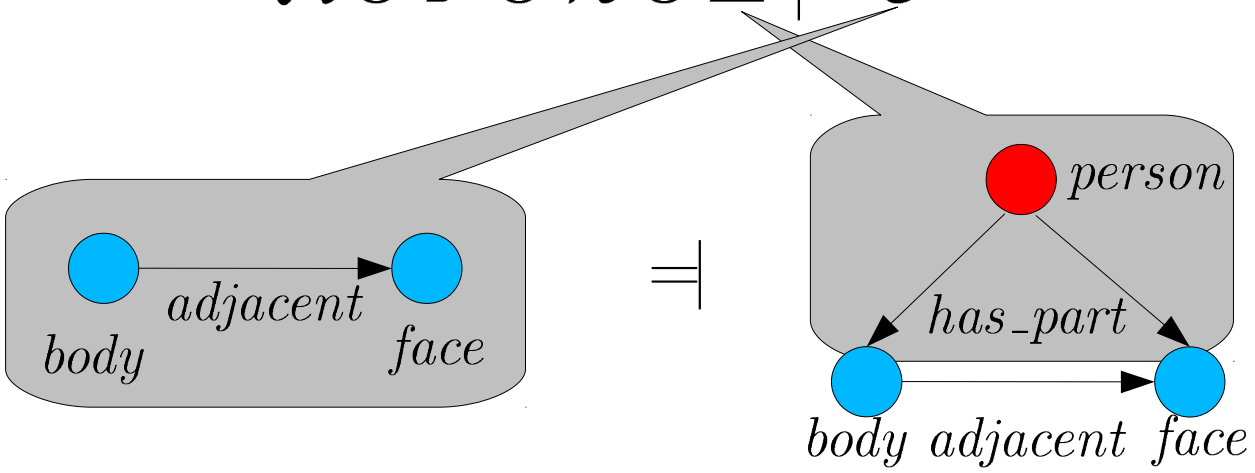
$$body(x), face(y), adjacent(x,y) \rightarrow person(z), has\_part(z,x), has\_part(z,y)$$

BOEMIE Abduction Framework

- abductive horn rules which „explain observations“ (Shanahan) – implemented in nRQL

$$adjacent(x, y) \leftarrow person(z), body(y), face(x), has\_part(z, x), has\_part(z, y)$$

$$\mathcal{A} \cup \mathcal{T} \cup \mathcal{R} \cup \Delta \models \mathcal{O}$$

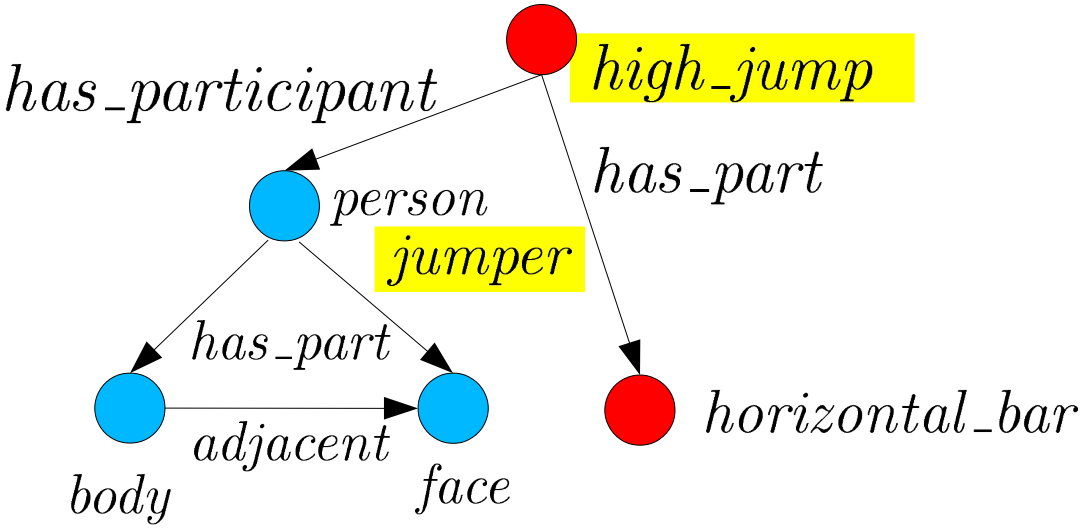


In general, more than one explanation possible

- minimality, consistency, preference, specificity
- probabilistic ranking (→ CASAM project)

BOEMIE Abduction Framework (2)

- Apply rules until fixpoint is reached

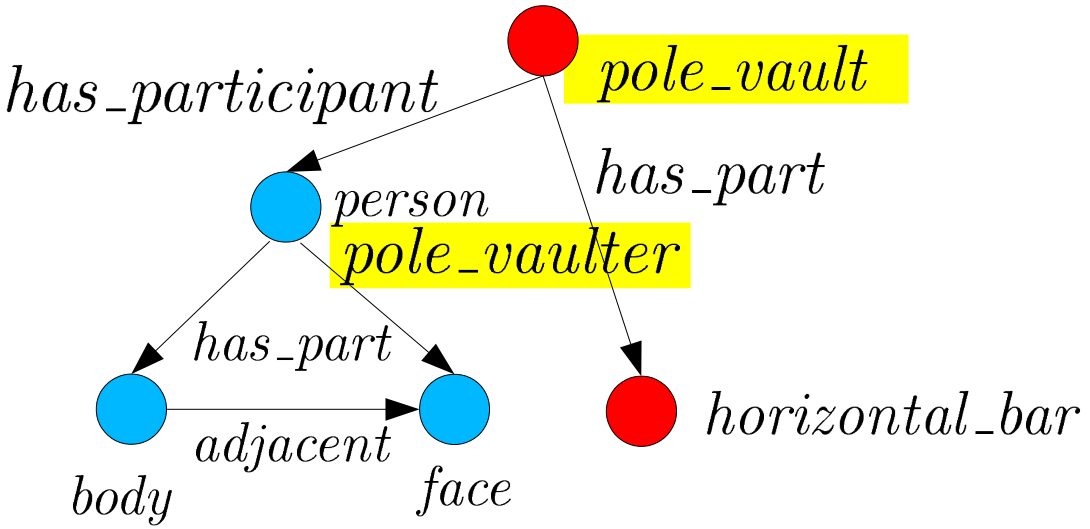


$$adjacent(y, z) \leftarrow high\_jump(x), jumper(y), horizontal\_bar(z), has\_participant(x, y), has\_part(x, z)$$

$$adjacent(y, z) \leftarrow pole\_vault(x), pole\_vaulter(y), horizontal\_bar(z), has\_participant(x, y), has\_part(x, z)$$

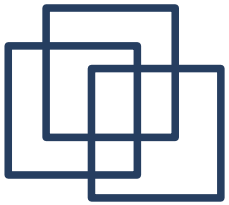
BOEMIE Abduction Framework (3)

- Apply rules until fixpoint is reached



$$adjacent(y, z) \leftarrow$$
$$high\_jump(x), jumper(y), horizontal\_bar(z),$$
$$has\_participant(x, y), has\_part(x, z)$$

$$adjacent(y, z) \leftarrow$$
$$pole\_vault(x), pole\_vaulter(y), horizontal\_bar(z),$$
$$has\_participant(x, y), has\_part(x, z)$$



Discussion and Opportunities

- Geo-spatial semantic web:
  - **GeoSPARQL** + Open Street Map + Map Server + Big Data + SemWeb → spatially aware semantic web → smarter SIRI, smarter QA, …
- I would like to participate in any project on logic-based image or scene understanding
  - some experience
  - probabilistic ranking of hypotheses (CASAM)
    - MARKO Description Logic + Rules
- Racer is still the best environment for Lisp-based ontology and SemWeb development
  - integration with Franz Allegro Graph technology