Flexible Tools for the Semantic Web

Michael Wessel
(instead of Jans Aasman from Franz Inc.)

Software Systems Group (STS)
Hamburg University of Technology (TUHH)
Hamburg-Harburg, Germany

(and Racer Systems GmbH & Co. KG)
Flexible Tools for the Semantic Web

- Portfolio of tools offered by Franz Inc. & Racer Systems
  - RacerPro, RacerPorter
  - AllegroGraph, Gruff
- These tools
  - are mature
  - are open for users (extensible)
  - are compatible with W3C standards and OpenSource frameworks
  - offer flexible and pragmatic solutions for which there are no standards yet (e.g., spatial queries via SPARQL extensions in AllegroGraph)
  - can be combined
STS, Racer Systems, Franz Inc.

- **Racer Systems** GmbH & Co. KG
  - Private, founded 2004
    - Prof. V. Haarslev (Concordia)
    - Prof. R. Möller (STS)
    - Kay Hidde
    - Michael Wessel (STS)
  - Commercial home of RacerPro
  - RacerPro, RacerPorter
  - Consulting

- **STS**
  - Databases, Software Systems, Semantic Web
  - The second „academic home“ of RacerPro

- **Franz Inc.**
  - Private, founded 1984 - U.C. Berkeley
  - CEO Dr. Jans Aasman
  - Lisp Vendor, AllegroGraph
  - Consulting

Offer expertise in semantics-based applications and mature and scalable tools
- Architecture & History

- Started as Racer at the University of Hamburg in 1998
- Working on large Aboxes since 1998 as a DL Abox reasoner
- One of the first OWL DL (-) system (2002), DL $SHIQ(D)$
- Expressive Abox query lang. nRQL
- First DL system that passed the „LUBM” (2004)
- First DL system with DL-aware SWRL (-) & SPARQL
- Interfaces: native, DIG (Protege 3), OWLAPI, OWLLink
- Main memory based, but interface to AllegroGraph
- Some special-purpose representations and reasoning
- Free for education & research
Demo of Tools ...

... using a well-known „standard“ OWL ontology

- Lehigh University Benchmark (LUBM)
- Standard inferences via RacerPro & RacerPorter
  - Taxonomy computation
  - Classification
  - Consistency checks
- Query answering
  - LUBM queries (chair example)
  - Queries with aggregation operators
  - Pragmatic stuff: HTML report generation via MiniLisp
Introduce LUBM...
Explain Chair...
Show Property Hierarchy...
Explain Class Reasoning...

RacerEditor

([#]:Professor)
[children
([#]:bottom bottom))

[5] > NIL

[6] > t

[7] > NIL

[8] ?

Class / Concept / IBox Reasoning

(concept-satisfiable? (and [#]:PostDoc (not [#]:Employee)))
(concept-subsumes? [#]:Employee [#]:PostDoc)
(concept-satisfiable? (and [#]:PostDoc (not (some [#]:worksFor [#]:Organization)))

(define-concept [#]:StudentTeacher
 (and [#]:GraduateStudent
 (some [#]:TeachingAssistantTo
 [#]:UndergraduateCourse)))

Query Answering

finished evaluating

RacerPro 1.3.3 running on localhost:3088 (core: preserve) — demo racer
Explain Classification.
Explain Chair Query...
Browse Abox for retrieved Chair Prof7...
Explain Prof7 satisfies Chair definition due to HeadOf Dep0
Check that Dep0 is indeed a Department (Direct Types)...
Show some nRQL aggregation queries...
Count number of publications of a Prof...
Generate a HTML query result page with MiniLisp...

Query Head

Query Body

Query Answer

<table>
<thead>
<tr>
<th>Result Tuple No. 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>?x <a href="http://www.Department0.University0.edu/FullProfessor7">http://www.Department0.University0.edu/FullProfessor7</a></td>
</tr>
<tr>
<td>?y <a href="http://www.Department0.University0.edu/Course10">http://www.Department0.University0.edu/Course10</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result Tuple No. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>?x <a href="http://www.Department0.University0.edu/FullProfessor7">http://www.Department0.University0.edu/FullProfessor7</a></td>
</tr>
<tr>
<td>?y <a href="http://www.Department0.University0.edu/Course11">http://www.Department0.University0.edu/Course11</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result Tuple No. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>?x <a href="http://www.Department0.University0.edu/FullProfessor7">http://www.Department0.University0.edu/FullProfessor7</a></td>
</tr>
<tr>
<td>?y <a href="http://www.Department0.University0.edu/GraduateCourse11">http://www.Department0.University0.edu/GraduateCourse11</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result Tuple No. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>?x <a href="http://www.Department0.University0.edu/FullProfessor7">http://www.Department0.University0.edu/FullProfessor7</a></td>
</tr>
<tr>
<td>?y <a href="http://www.Department0.University0.edu/GraduateCourse12">http://www.Department0.University0.edu/GraduateCourse12</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result Tuple No. 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>?x <a href="http://www.Department1.University0.edu/FullProfessor4">http://www.Department1.University0.edu/FullProfessor4</a></td>
</tr>
<tr>
<td>?y <a href="http://www.Department1.University0.edu/Course7">http://www.Department1.University0.edu/Course7</a></td>
</tr>
</tbody>
</table>

Finished evaluating

RacerPro 1.9.3 running on localhost:8088 (easy-preserve) — demo racen (Li)
First-Order Features of nRQL

• NAF and projection
  - e.g., for closed-domain universal quantification
  - „find the oldest known professor that still gives lectures“ (all other known lecture giving professors are younger)
  - makes sense, even with Open World Assumption
  - Non-DL-safe non-monotonic rules that construct new individuals and auto-disable (termination)

• In principle, this is also possible with SPARQL
  - equivalent to relational Algebra (Angles, ISWC '08)
  - … but not handy
    (no explicit NAF and projection, have to use bound)
Add random age1 datatype fillers to chair profs with a rule and retrieve those ages...
Retrieve the oldest (known)
Chair prof. that still gives
lectures...
Application – Linear Time Event Recognition

- 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
Ensure Maximum Duration & Homogeneity

... e.g., for the „staying in office event“
Ensure Maximum Duration & Homogeneity

... and construct a new aggregating event, similar to SPARQL `construct`
nRQL Event Rule Example

(prepare-abox-rule
  (and (?s1 state) (?s2 state) (?s1 ?s2 future)
   (?s1 ?r in-region) (?s2 ?r in-region) (?r office)
   (neg (project-to (?s2 ?r)
     (and (?s2 ?s next) (?s ?r in-region))))
   (neg (project-to (?s1 ?r)
     (and (?s ?s1 next) (?s ?r in-region))))
   (neg (project-to (?s1 ?s2 ?r)
     (and (?s1 ?s future) (?s ?s2 future)
       (neg (?s ?r in-region))))))
   (neg (project-to (?s1 ?s2)
     (and (?e in-office-event)
       (?e ?s1 start-state)
       (?e ?s2 end-state)))))

((instance (new-ind new-simple-event ?s1 ?s2) in-office-event)
 (related (new-ind new-simple-event ?s1 ?s2) ?s1 start-state)
 (related (new-ind new-simple-event ?s1 ?s2) ?s2 end-state)))
Solving Constraint Satisfaction Problems by Reasoning - Sudoku

Create a KB whose logical models represent all possible Sudoku solutions. If a Sudoku has only ONE solution, then the solution can be obtained with a query (solution = entailed atoms)

\[
pairwise\_disjoint(C_1, C_2, C_3, C_4)
\]

\[
\bigcap \subseteq (C_1 \cup C_2 \cup C_3 \cup C_4) \cap (C_1 \rightarrow \forall R. \neg C_1) \cap (C_2 \rightarrow \forall R. \neg C_2) \cap (C_3 \rightarrow \forall R. \neg C_3) \cap (C_4 \rightarrow \forall R. \neg C_4) \cap ... 
\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C1

C2

C3

C4
Sudoku – Abox Construction

ABox construction

- by hand? OK for 4x4, but for 9x9?
  → create the structure programmatically
    (MiniLisp or OWLAPI)
- or via rules from asserted part
- transitive & symmetric property →
  - use different backward property instead of a symmetric property
  - qualification over common parent property

$C_4 \rightarrow \forall R. \neg C_4$
Sudoku (3)

\[ Q_1 \sqsubseteq R \]
\[ Q_2 \sqsubseteq R \]
transitive(\( Q_1 \))
transitive(\( Q_2 \))
\[ Q_1(x, y) \leftrightarrow Q_2(y, x) \]
Explain DL modeling...
Add a new server function to RacerPro ...)
Franz Inc. - AllegroGraph 3.2 & Gruff

- Scalable and persistent RDF Database (Triple Store)
  - Loads a Billion triples in 8 hours on a 4 processor AMD machine
  - Load 10 Billion triples on EC2 (Amazon) on 10 machines in 10 hours
- Free version: 50.000.000 triples
- RDFS++ SPARQL / Prolog query answering
  - `rdf:type, rdfs:domain, rdfs:range, rdfs:subClassOf, rdfs:subPropertyOf, owl:inverseOf, owl:sameAs, owl:TransitiveProperty`
- Federated
  - Create an abstract store that is the union of other triple stores
  - Query answering works transparently against abstract store
- Geo-spatial-temporal and SNA predicates
  - Accessible from Prolog / SPARQL (extension mechanism)
Open the LUBM triple store...
Display it...
Explain that FullProfessors are directly asserted...
But RDFS++ reasoning is required to get the Professors...
Enable the RDFS++ reasoning for query answering...
We then get also the professors...
And also the persons...
... but not the Chairs, so let RacerPro do some materialization later...
LUBM(8000) = 1,105,993,401 Triples

![Bar chart showing LUBM(8000) Total query time with AllegroGraph 3.2 and Other categories.](image-url)
So what is the big deal?

LUBM(8000) Total Time

- Other / Static
- AllegroGraph 3.2
- AllegroGraph 3.2 Federated

- Total Query Time
- Type Materializations
- Loading and Indexing
Activity Recognition

• Our customers use AllegroGraph as an event database with social network analysis and geospatial and temporal reasoning.

Find all meetings that happened in November within 5 miles of Berkeley that was attended by the most important person in Jans’ friends and friends of friends.

(select (?x)
(ego-group person:jans knows ?group 2) SNA
(actor-centrality-members ?group knows ?x ?num) SNA
(q ?event fr:actor ?x) DB Lookup
(qs ?event rdf:type fr:Meeting) RDFS
(interval-during ?event “2008-11-01” “2008-11-06”) Temporal
(geo-box-around geoname:Berkeley ?event 5 miles) Spatial
!)


Social Network Analysis
Answers 4 questions

• How far is P1 from P2 (and how strong is the relation?)
• To what groups does this person belong (ego groups, cliques?)
• How important is this person in the group?
• Does this group have a leader, how cohesive are they?
GeoSpatial

• Make the following super efficient
  – Where did something happen?
  – How far was event1 from event2?
  – Find all the events that occurred in a bounding box or radius of M miles?
  – Do these two shapes overlap?
  – Find all the objects in the intersection of two shapes
• On a very large scale
  – when things don’t fit in memory
  – millions of events and polygons
Temporal Reasoning

- Adhere to our convention to encode StartTimes and EndTimes and enjoy efficient temporal primitives
- Implementation of Allen’s interval logic primitives

<table>
<thead>
<tr>
<th>Allen's Interval Logic</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(interval-before ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-meets ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-overlaps ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-starts ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-during ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-finishes ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-after ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-met-by ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-overlapped-by ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-started-by ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-contains ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-finished-by ?e1 ?e2)</td>
<td></td>
</tr>
<tr>
<td>(interval-cotemporal ?e1 ?e2)</td>
<td></td>
</tr>
</tbody>
</table>
A SPARQL query spanning 4 sources

```
prefix go: <http://purl.org/obo/owl/GO#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix mesh: <http://purl.org/commons/record/mesh/>
prefix sc: <http://purl.org/science/owl/sciencecommons/>
prefix ro: <http://www.obofoundry.org/ro/ro.owl#>

select ?genename ?processname
where
{ graph <http://purl.org/commons/hcls/pubmesh> 
    ?article sc:identified_by_pmid ?paper.
    ?gene sc:describes_gene_or_gene_product_mentioned_by ?article.
  } 
  graph <http://purl.org/commons/hcls/goa> 
    ?res owl:onProperty res:has_function.
    ?res2 owl:onProperty res:realized_as.
    ?res2 owl:someValuesFrom ?process.
  } 
  graph <http://purl.org/commons/hcls/20070416/classrelations> 
  { (?process rdfs:subClassOf go:GO_0007166) 
    ?res3 owl:hasValue ?gene.
  } 
  graph <http://purl.org/commons/hcls/gene> 
  { ?gene rdfs:label ?genename }
  graph <http://purl.org/commons/hcls/20070416> 
  { ?process rdfs:label ?processname}
}
```

Mesh: Pyramidal Neurons

Pubmed: Journal Articles

Entrez Gene: Genes

GO: Signal Transduction

Inference required
AllegroGraph and RacerPro

- Racer can do type & property materialization for AllegroGraph
  - Racer can read OWL RDF from triple store, web or file
  - materialize inferences („type & property materialization“)
  - write back
    → e.g., ontology repository of the EU project
      - Much better than relational DB, same data model
      - Integrated with RacerPro
      - SPARQL queries from AllegroGraph, Gruff, …
- Drawback: type materialization only possible for Aboxes that fit in main memory
  - future work: RacerPro reasoning directly on triple store
    → not limited w.r.t. main-memory size
Let RacerPro do the materialization for LUBM...
Open the LUBMMAT triple store...
Browser for Chair predicate...
Check that the Chair instances have been materialized by RacerPro.
BOEMIE – Ontology-Based Information Extraction from Athletics New WebPages

Similar to OpenCalais, but with deeper and athletics-domain specific interpretation capabilities, e.g. can hypothesize complex Sports events from parts
By combining „clues“ from different modalities
(→ ontology-based abduction mechanism!)
Hestrie Cloete on her lap of honour in Zurich. Cloete and Ayhan to headline again.

As the penultimate stage of the IAAF Golden League takes on particular significance as the pressure gauge of form, just a week after the 9th IAAF World Championships in Athletics (23-31 August 2003), Hestrie Cloete and Sıriver Ayhan were the undoubted stars of last Sunday's ISTAF Golden League meeting in Berlin, and the South African World High Jump champion and European 1500m gold medallist are sure to be headline again in Zurich.

With a practically unblemished scorecard up to and including her 2:08m African record in Berlin, will have a World record 2:10 clearance in her sights once more today. Wearing the 22:02 bib number on her vest last Sunday was perhaps too weighty a burden to carry as her three heavily failed attempts moved but that heigt is surely with in.
BOEMIE Project

[Diagram showing a pyramid with layers labeled Fusion, Interpretation, Low-level Analysis, and Non-Visual Information. Arrows indicate flow and information gain.]
Conclusion

- I have demonstrated
  - RacerPro & RacerPorter
    - Standard reasoning
    - ABox query answering
    - Solving application problems with reasoning
    - Convenient & flexible ad hoc extensibility
  - AllegroGraph & Gruff
    - SPARQL query answering (Prolog also possible)
    - Visualization
    - RDFS++ query answering
  - „Type and inferred property materialization“ via RacerPro
- Thanks 😊
Abox Query Answering

Assuming all blocks are red or green - is there a green block on the table which is next to a red one? (Brachman & Levesque)

\[
\{ \text{block} \sqsubseteq \text{red} \sqcup \text{green}, \text{next-to} \equiv \text{next-to}^{-1} \} \\
\{ t : \text{table}, lb : \text{green} \sqcap \text{block}, rb : \text{red} \sqcap \text{block}, \text{mb} : \text{block}, ob : \text{green} \sqcap \text{block}, \\
(lb, t) : \text{on-table}, (ml, t) : \text{on-table}, (rb, t) : \text{on-table}, \\
(gb, ob) : \text{next-to}, (ob, rb) : \text{next-to} \}
\]

Ask for instances of the concept

\[
\text{table} \sqcap \\
\exists \text{on-table}^{-1}. (\text{block} \sqcap \text{green} \sqcap \\
\exists \text{next-to}. (\text{red} \sqcap \text{block}))
\]
There are two possibilities. If the middle block is red, then the green left block is next to a red one. But...

Model 1

\[ \exists_{on\_table}^{-1}. (\text{block} \sqcap \text{green} \sqcap \exists_{next\_to}. (\text{red} \sqcap \text{block})) \]
... if the middle block is green, then it is also next to the right Block, which is red. So, yes, there ALWAYS EXISTS such a block on the table!
Full Conjunctive Queries

\[ ans(x) \leftarrow table(x), on\_table(y, x), block(y), green(y), next\_to(y, z), red(z), block(z). \]

Answer: \( x = t \)

- However, no answer for head \( ans(y) \leftarrow \ldots \)
  - distinguished variables in head: binding must hold in ALL models ("certain answer")
  - other variables: treated as existentially quantified
Grounded Conjunctive Queries

\[ \text{ans}(x) \leftarrow \text{table}(x), \text{on_table}(y, x), \]
\[ \text{block}(y), \text{green}(y), \]
\[ \text{next_to}(y, z), \text{red}(z), \text{block}(z). \]

Gives no answer in nRQL!

- In grounded conjunctive queries
  - ALL variables are distinguished; a binding is only established iff it holds in ALL models
  - grounding: subst. variables \( \leftrightarrow \) entailed assertions
Grounded CQs vs. Full CQs

In this example, a complex concept can be used. But that's not always the case

- „Rolling up“ technique
  - note that variables may introduce coreferences
  - no automatic rolling up in nRQL
  - newer results for full conjunctive queries and OWL available, but not implemented AFAIK