A Flexible DL-based Architecture for Deductive Information Systems

Michael Wessel and Ralf Möller

Software Systems Group (STS)
Hamburg University of Technology (TUHH)
Hamburg, Germany
{mi.wessel | r.f.moeller}@tuhh.de
Background and Motivation

- Description logics (DLs) provide widely accepted standards for decidable knowledge representation.
- Benefits: Scalability of expressivity, foundations for ontology languages (Semantic Web), impressive theory, performant DL systems (DLS).
- Question: A basis for ontology-based information systems (IS)?
- Evaluation: How to use and extend a DL system for building an ontology-based query answering system for city maps?
- Contribution: Pragmatic solutions to tackle the design, representation and query answering problem in domains for which standard DLs are not well-suited.
Drawbacks of Today’s DLS for IS Building

- Scalability for ABoxes not easy to achieve (see LUBM with NRQL + RACERPRO)
- Few have expressive and practical ABox QL
- Persistency for large ABoxes? Use a DB, but:
  - Query answering requires inference – DB access during reasoning? Pre-loading of (which portions of the) ABox into memory?
  - Thesis: as long as RPC is much slower, representation layer should include query answering engine and reasoning engine
- ABoxes not good for “data representation” (Strings, Polygons), Concrete Domains: overkill in many cases
Underlying Thesis

• Pragmatic solutions needed for today's IS designers
  • Existing DLS must be reused (cannot be reimplemented in a short period of time)
  • Pragmatic ways to extend DLS in case representational deficiencies are encountered should be identified → this paper
  • Extensions must / should be easier to implement than DLS
  • Longtime perspective: DLS with “open architecture”? Plug in mechanism?
  • Even if this is achieved, extensibility is hard due to inherent intellectual complexity
Contributions

- “Road map for a difficult terrain”
- Three pragmatic settings for city maps IS
  - Setting 1: What can be done with RACERPRO ABox left alone
  - Setting 2: (RACERPRO external) spatial representation (“spatial DB”) + ABox
  - Setting 3: (RACERPRO internal) spatial representation + ABox
- Base the IS on software abstractions
  - Abstract from remote vs. local procedure call
  - Abstract from ABox representation of the DLs
  - Provide more flexible means for data / information representation than ABoxes
Contributions (2)

- Base the IS on software abstractions
  - Allow for hybrid representations (layering of representations)
  - Use one internal data model for the IS which allows all this: “substrate data model”
  - Flexible, extensible query language needed: “substrate query language framework”

⇒ “Semantic Middleware”

- Thesis: flexible way to build DL-based IS
- Example: NRQL was first external to RACERPRO; since it was implemented on top of “substrate” middleware, it could be migrated easily into RACERPRO
Digitale Stadtkarte (DISK)

- ©“Amt für Geoinformation und Vermessung”
- Two digital vector maps in SQD format
- Objects are “classified” according to a fixed list of categories (“Objektschlüssel-Katalog”):
  - 5164 ⇒ lake, navigable
  - 4128 ⇒ meadow
  - 2224 ⇒ park
  - 2119 ⇒ living area, ...
- Taxonomic relationships (“is-a”) implicitly present, but not explicitly modeled ⇒ needs remodeling
- Very specific categories, no generalizing categories: cemetry_for_non_christians, but cemetry is missing
Digitale Stadtkarte (DISK)

- Map 1: 2694 geo objects, 361 primary
Digitale Stadtkarte (DISK)

- Map 2: 18,039 geo objects, 5,418 primary
Digitale Stadtkarte (DISK)

- © “Amt für Geoinformation und Vermessung”
- Two digital vector maps in SQD format
- Objects are “classified” according to a fixed list of categories (“Objektschlüssel-Katalog”):
  - 5164 ⇒ lake, navigable, 4128 ⇒ meadow,
  - 2224 ⇒ park, 2119 ⇒ living area, …
- Taxonomic relationships (“is-a”) implicitly present, but not explicitly modeled ⇒ needs remodeling
- Very specific categories, no generalizing categories: cemetry_for_non_christians, but cemetry is missing
Aspects of Geographic Data

- Two groups of aspects
  - Thematic: Geographic Category (Semantics)
  - Spatial: Area, Shape, Relationships, ...

- Space has specific properties

- Instances of spatial data types (polygons etc.) automatically preserve many important spatial aspects of the represented geo object (a polygon represents both shape and area)

- A map intrinsically represents spatial relationships ⇒ rich, “analogical” representation

- In an ABox, everything must be represented symbolically ⇒ “symbolic bottleneck”
Spatio-Thematic Querying

- DISK ontology in a TBox
- Remodelling of thematic DISK categories:
  \[ \text{cemetery} \subseteq \text{cemetery} \]
- Additional \textbf{spatio-thematic} vocabulary:
  \[ \text{park}_\text{with}_\text{lake} \equiv \text{park} \land \exists \text{contains}\.\text{lake} \]
- Use \textbf{spatio-thematic} vocabulary in queries:
  \[
  \text{ans}(\text{?lake}, \text{?park}, \text{?creek}, \text{?industrial}_\text{area}, \text{?chemical}_\text{plant}) \leftarrow \\
  \text{lake}(\text{?lake}), \text{chemically}_\text{contaminated}(\text{?lake}), \text{park}(\text{?park}), \\
  \text{contains}(\text{?park}, \text{?lake}), \text{creek}(\text{?creek}), \\
  \text{flows}_\text{in}(\text{?creek}, \text{?lake}), \text{crosses}(\text{?creek}, \text{?industrial}_\text{area}), \\
  \text{contains}(\text{?industrial}_\text{area}, \text{?chemical}_\text{plant}), \\
  \text{unreliable}(\text{?chemical}_\text{plant}).
  \]
Representing DISK: Setting 1

- What can be done with RacerPro left alone?
- Remodeling of thematic categories in a TBox
  - \( \text{concept}_\text{for_key}(2224) = \text{def } park \)
Representing DISK: Setting 1

- What can be done with RacerPro left alone?
- Remodeling of thematic categories in a TBox
  - \( \text{concept}_{\text{for_key}}(2224) =_{\text{def}} \text{park} \)
- Representation of the map as an ABox \( \mathcal{A} \)
  - For each map object \( i \) with key \( n \), add \( i : \text{concept}_{\text{for_key}}(n) \) to \( \mathcal{A} \)
  - Represent dedicated metric aspects in the CD \( (i : \exists(area). =_{123456}) \)
Representing DISK: Setting 1

- What can be done with RacerPro left alone?
- Remodeling of thematic categories in a TBox
  - $\text{concept}_{\text{for}_{\text{key}}}(2224) = \text{def } \text{park}$
- Representation of the map as an ABox $\mathcal{A}$
  - For each map object $i$ with key $n$, add $i : \text{concept}_{\text{for}_{\text{key}}}(n)$ to $\mathcal{A}$
  - Represent dedicated metric aspects in the CD $(i : \exists \text{(area)}. =_{123456})$
- Compute qualitative RCC8 relationships: $(i, j) : \text{EC}, (i, k) : \text{TPPI}, \ldots$
Representing DISK: Setting 1

• What can be done with RacerPro left alone?
• Remodeling of thematic categories in a TBox
  • $\text{concept}_{\text{for key}}(2224) =_{\text{def}} park$
• Representation of the map as an ABox $\mathcal{A}$
  • For each map object $i$ with key $n$, add $i : \text{concept}_{\text{for key}}(n)$ to $\mathcal{A}$
  • Represent dedicated metric aspects in the CD $(i : \exists(area). =_{123456})$
  • Compute qualitative RCC8 relationships: $(i, j) : \text{EC}, (i, k) : \text{TPPI}, \ldots$
  ⇒ RCC8 network in the ABox (network is always consistent)
Illustration: DISK ABox
Querying the DISK (Setting 1)

- Simple spatio-thematic QL: instance retrieval queries
- \( \text{concept\_instances}(park\_with\_lake) = \{i, \ldots\} \)
- \( i \in \text{concept\_instances}(park \sqcap (\exists NTPPI.lake \sqcup \exists TPPPI.lake)) \)
- If \( \{ i : park, k : lake, j : meadow, (i, j) : TPPI, (j, k) : NTPPI \} \subseteq A \), then also \( (i, k) : NTPPI \in A \) (due to map)
- RCC roles in ABox must be closed: for each \( i \), add \( i : (\leq n R) \sqcap (\geq n R) \)
- \( n =_{def} |\{ j \mid (i, j) : R \in A \}| \)
Querying the DISK (Setting 1) (2)

- \text{concept\_instances}(bird\_sanctuary) = \{i, \ldots\}
- \text{bird\_sanctuary} \equiv \text{park} \sqcap \forall \text{NTPPI}.\neg \text{building} \sqcap \forall \text{TPPI}.\neg \text{building}

\[ A \cup \{(i, k) : \text{NTPPI}\} \cup \{(i : (\leq_1 \text{TPPI}) \cap (\geq_1 \text{TPPI}), i : (\leq_1 \text{NTPPI}) \cap (\geq_1 \text{NTPPI}), \ldots) \cup \{i : (\neg \text{park} \sqcup ((\exists \text{TPPI}.\text{building}) \sqcap (\exists \text{NTPPI}.\text{building})))\}\}

- is unsatisfiable, if \{\text{building, park, meadow}\} are mutually disjoint
- Problems in the TBox: incomplete subsumption relationships (not a problem for query answering)
- Moreover, nRQL can be used
Querying the DISK with NRQL

- NRQL offers classical negation and NAF
- For which living areas can it be proven that there are no adjacent freeways?
  \[
  \text{ans}(\text{?living\_area}) \leftarrow \text{living\_area}(\text{?living\_area}), \\
  \forall \text{adjacent}. \neg \text{freeway}(\text{?living\_area})
  \]
- Requires RCC closed ABox and disjointness axioms, as just discussed (not so good ...)
- Living areas with no known adjacent freeways?
  \[
  \text{ans}(\text{?living\_area}) \leftarrow \\
  \text{living\_area}(\text{?living\_area}), \\
  \backslash (\pi(\text{?living\_area}) \text{ adjacent}(\text{?living\_area, ?freeway}), \\
  \text{freeway}(\text{?freeway}))
  \]
NRQL Concrete Syntax

Q1:
(retrieve (?x)
  (and (?x living-area)
   (?x (all adjacent
    (not freeway)))))

Q2:
(retrieve (?x)
  (and (?x living-area)
   (not (project-to (?x)
     (and (?x ?y adjacent)
      (?y freeway))))))
Extensible Substrate QL Framework

- NRQL is a specialized “Substrate QL” for substrates $S$ of type ABox
- Two kinds of atoms: unary $C(x)$, binary $R(x, y)$
- $x, y$: Individuals, variables (with act. dom. sem.)
- Extensions of atoms on substrate $S$:
  $C(x)^E = \{ i \mid S \models C(i) \}$,
  $R(x, y)^E = \{ (i, j) \mid S \models R(i, j) \}$.
- Due to NAF: $\models_{NAF}$ instead of $\models$
- Complex queries: Relational operators AND (“$\times$”), UNION ($\cup$), NEG ($\setminus$), PROJECT-TO ($\pi$)
- New atoms can be added: $\models$ must be defined for $S \times atom$, $\Rightarrow$ very flexible!
Problems with Setting 1

- $n^2$ size of generated ABoxes (29 million role assertions with $DC$, 19,880 without DC)
- Missing practically relevant query atoms (e.g., distance queries)
- Qualitative representation of “spatial data” in an ABox
- “Closed domain reasoning” required $\Rightarrow$ misuse of the DL system (open domain reasoning)
- Geometry needed anyway, at least for presentation purposes
- $\Rightarrow$ Motivates hybrid representation and query language
Representing the DISK: Setting 2

- Map Substrate: \((ABox, SBox, \ast)\)
  - Substrate 1: ABox - thematic aspects
  - Substrate 2: SBox - map geometry
  - \(\ast\): part. inject. mapping from nodes in \(S_1\) to \(S_2\)
Representing the DISK: Setting 2

- Map Substrate: \((ABox, SBox, *)\)
  - Substrate 1: ABox - thematic aspects
  - Substrate 2: SBox - map geometry
  - \(*\): part. inject. mapping from nodes in \(S_1\) to \(S_2\)
Representing the DISK: Setting 2

- Map Substrate: \((ABox, SBox, \ast)\)
  - Substrate 1: ABox - thematic aspects
  - Substrate 2: SBox - map geometry
  - \(\ast\): part. inject. mapping from nodes in \(S_1\) to \(S_2\)

- Non-symbolic spatial binary query atoms
- On-demand computation and inspection of spatial aspects
- Dedicated index structures
- Closed world reasoning in \(SBox\)
- Simple model checking
- No more “reasoning” on spatial aspects
Spatial $\text{SNRQL}$

- Two sorts of atoms in $\text{SNRQL}$
  - ABox atoms:
    - Variables range over ABox individuals
    - Atoms as in $\text{NRQL}$ (concept, role, constraint query atoms)
  - Spatial atoms:
    - Variables range over SBox individuals
    - RCC atoms
    - Geometric attributes: area, length, ...
    - Metric relationships: range queries, epsilon queries, ...
- Variables are bound in parallel, bindings reflect the "*" mapping
Spatial NRQL - SNRQL

Range Query

Epsilon Query
SNRQL Concrete Syntax

(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)))))
Hybridness can be made transparent

- Add end-user syntax for DLMAPS system: user must not be aware of the details of the map representation

  - \( \text{ans}(x, y) \leftarrow \text{park}(x), \text{contains}(x, y), \text{lake} \_ \text{or} \_ \text{pond}(y) \)

  - \( \text{lake} \_ \text{or} \_ \text{pond}(y) \rightarrow (\text{lake} \sqcup \text{pond})(\ast y) \)

  - \( \text{contains}(x, y) \rightarrow \text{NTPPI}(x, y) \lor \text{TPPI}(x, y) \)
Problems with the Approach

- Theoretical problems: No spatial reasoning
- Practical problems (perspective: IS designer)
  - ABox / SBox separated, communication overhead (caches required)
  - Hybrid QL required
  - IS designers probably do not want to implement a query answering engine, thus:
    - Split hybrid query into subqueries, send to different sources, combine sub-results
    - Probably bad performance (no overall query optimization, communication overhead, combination of results, ...)

⇒ Shows a way, but too hard to realize
Hybrid Substrates in RACERPRO

- Thus, in order to avoid these problems, this functionality should be put into the DL system

⇒ Hybrid substrates for RACERPRO

- Makes functionality available for other IS designers

- Compensate for representational deficiencies of the ABox
  - Data substrate: stores told value data (from CD of ABox or OWL documents, enabled retrieval facilities)
  - RCC substrate: associate an RCC network with an ABox (next) ⇒ limited form of spatial reasoning
The Data Substrate

(retrieve (?x ?*name ?*age)
   (and (?x (and |http://...#person|
           (an |http://...#age|))))
   (?*x ?*name |http://...#name|)
   (?*name ( (:predicate (search "wessel")))
     ( (:predicate (search "michael")))
     (:predicate (search "achim"))))
   (?*x ?*age |http://...#age|)
   (?*age ( (:predicate (< 40))))))

• New sort of variables: *?x (*$?x), ranging over data nodes
• Data nodes can also be data values in OWL documents
• Data nodes/edges have descriptive labels: kind, role, property, …
• Notion of entailment for labels of nodes/edges
• Data query atoms are in pos. CNF & contain literals and predicates.
The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
The RCC Substrate

• Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")

• Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?

• Not so easy, since $\mathcal{R}$ (or $R$) can contain non-base relations
The RCC Substrate

A

B

C

D

(A,A) : EQ

(B,B) : EQ

(B,C) : PP

(B,D) : DR

(D,D) : EQ

(C,C) : EQ
The RCC Substrate
The RCC Substrate
The RCC Substrate

\[ \models DR(A, D) \]
\[ \models PP(A, C) \]
\[ \models \{PPi, PO, DR\}(C, D) \]
\[ \models \ldots \]
The RCC Substrate

• Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models (“certain answer”)

• Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?

• Not so easy, since $\mathcal{R}$ (or $R$) can contain non-base relations

• Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or $R$) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Setting 3 for DLMAPS
The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or $R$) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Setting 3 for DLMAAPS
- (Proof of concept implementation)
The RCC Substrate

• Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")

• Question: Holds \( R(x, y) \) in all models of the RCC network \( \mathcal{R} \)?

• Not so easy, since \( \mathcal{R} \) (or \( R \)) can contain non-base relations

• Reduction to satisfiability: \( R \models R(x, y) \) iff \( \mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y) \) unsatisfiable

• Setting 3 for DLMAPS

• (Proof of concept implementation)
Conclusion

• Spatial parts from spatio-thematic concepts removed from the TBox
• Instead, spatially aware query answering instead of TBox / concept reasoning
• Simple layering of representations can be of great value in practice
• (Possibly) Hybrid substrate QL framework allows for extensibility
• Base the IS on abstractions so that the representation can be changed easily
• (Theoretically) simple techniques can be successful in practice
Thanks for your attention!