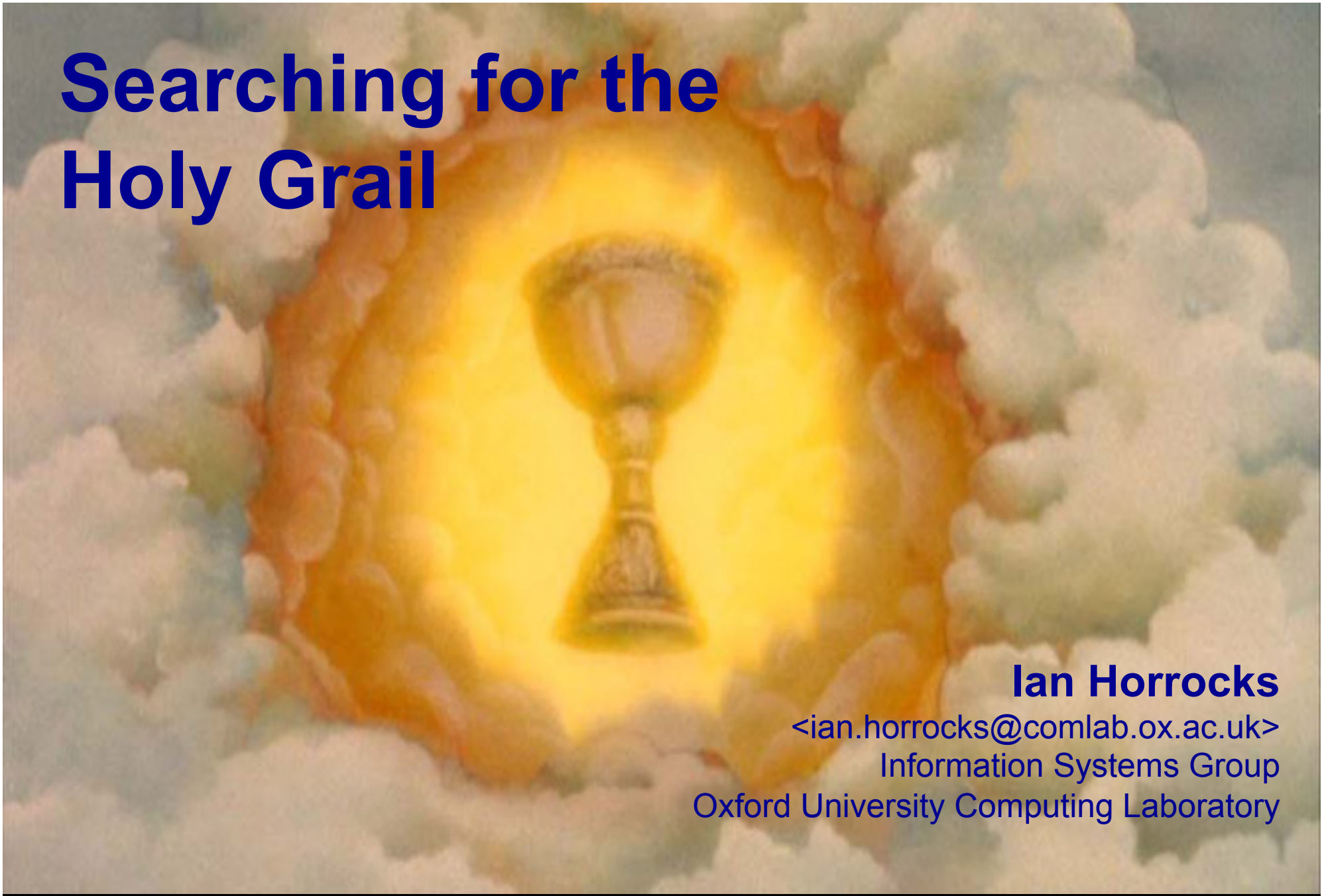
Searching for the Holy Grail



Ian Horrocks
<ian.horrocks@comlab.ox.ac.uk>
Information Systems Group
Oxford University Computing Laboratory

Background and Motivation

- Medicine has a large and complex vocabulary
- Long history of “formalising” and codifying medical vocabulary
  - Numerous medical “controlled vocabularies” of various types
- Large size of static coding schemes makes them difficult to build and maintain
  - Many terminologies specific to purpose (statistical analysis, bibliographic retrieval), specialty (epidemiology, pathology) or even database
  - Ad hoc terms frequently added to cover fine detail required for clinical care

Background and Motivation

Schemes such as **SNOMED** tackled some of these problems by allowing codes to be constructed, but this introduced its own problems:

- **Vague semantics**, e.g., conflating different relations:

  T-1X500 = bone
  T-1X501 = long bone (kind-of)
  T-1X505 = shaft of bone (part-of)
  T-1X520 = cortex of bone (constituent-of)

Background and Motivation

Schemes such as **SNOMED** tackled some of these problems by allowing codes to be constructed, but this introduced its own problems:

- **Redundancy**, e.g.:

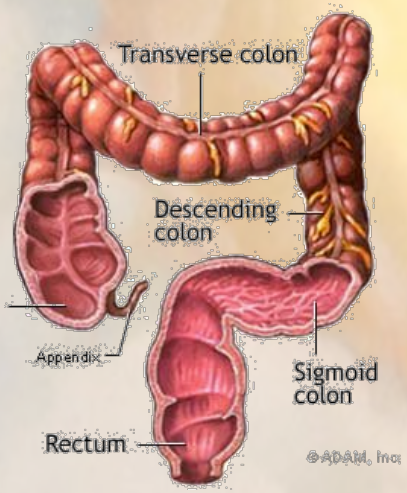
  T-28000 + E-2001 + F-03003 + D-0188 =
  tuberculosis in lung caused by M.tuberculosis together with fever

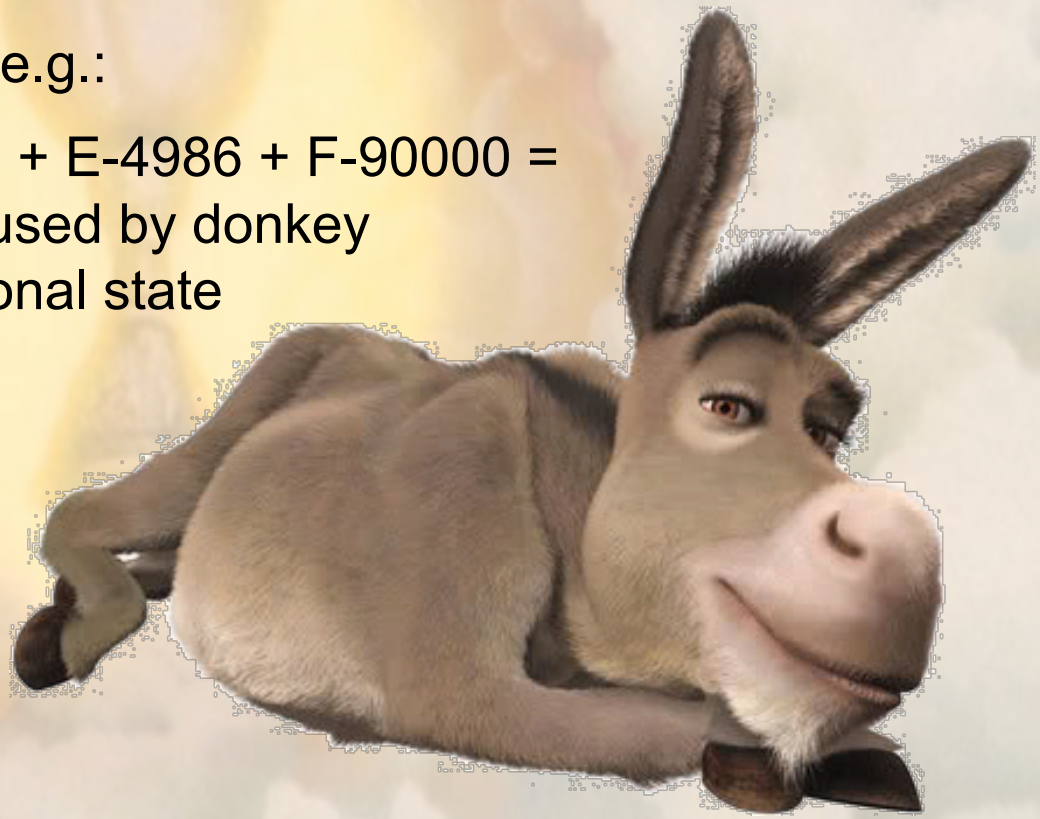
Background and Motivation

Schemes such as **SNOMED** tackled some of these problems by allowing codes to be constructed, but this introduced its own problems:

- **Nonsensical terms**, e.g.:

  T-67000 + M-12000 + E-4986 + F-90000 =
  fracture in colon caused by donkey
  together with emotional state





Proposed Solution

Use a **conceptual model**

- Detailed descriptions with clear semantics and principled extensibility
- Can use tools to support development and deployment, e.g.:
  - Consistency checking and schema enrichment through the computation of implicit subsumption relationships
  - Intensional and extensional query answering and query optimisation

GALEN Project

Goals of the project were:

- Design/select an appropriate (for medical terminology) modelling language: **GRAIL**
- Develop tools to support conceptual modelling in this language: **GRAIL classifier** (amongst others)
- Use these tools to develop a suitable model of medical terminology: **GALEN terminology** (aka ontology)

Recognised Problems

- **Classifier too slow**
  - Over 24 hours to classify ontology
- **My mission**: make it go faster

Hint: DL research might be relevant

Unrecognised Problems

- Vague semantics
  - no formal specification or mapping to (description) logic
- Language lacked many features
  - cardinality restrictions (other than functional roles)
  - negation and disjunction (not even disjointness)
- Reasoning via ad hoc structural approach
  - incorrect w.r.t. any reasonable semantics

Why Not Use a DL?

- Formalise semantics
  - establish mapping from GRAIL to a suitable DL
- Use suitable DL reasoner to classify resulting TBox
  - must support transitive roles, GCIs, etc.
- Does such a reasoner exist?
  - Yes: **LOOM**

Idea: translate GALEN ontology into LOOM DL and use LOOM classifier

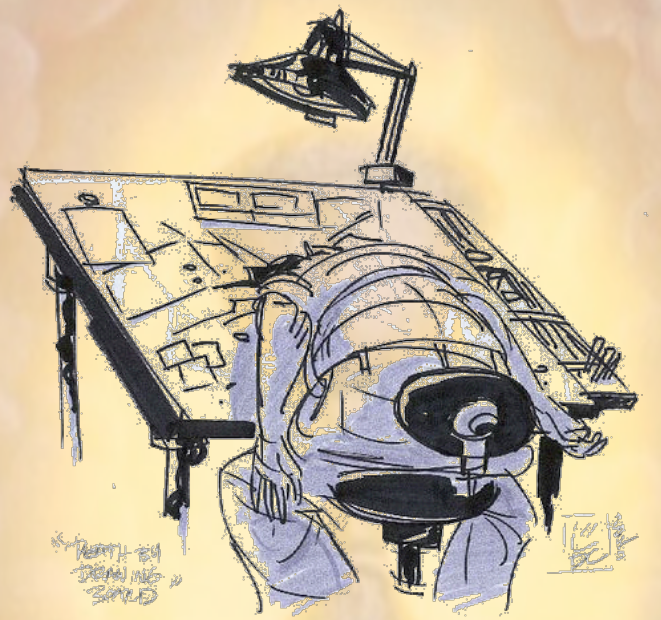
The False Grail

Results less than 100% satisfying:

- It gets the wrong answer (fails to find obvious subsumptions)
- It's even slower than the GRAIL classifier

Lesson: No such thing as a free lunch!

Back to the Drawing Board



Idea: Implement my own fast and correct reasoner for a very expressive DL!

Implementing a DL Reasoner

- What algorithm is implemented in LOOM?

  "... utilizes forward-chaining, semantic unification and object-oriented truth maintenance technologies ..."

- Alternative approaches?

  tableau algorithms

Implementing a Tableau Reasoner

- Advantages:
  - algorithms relatively simple, precisely described and available for a range of different logics
  - formal correctness proofs, and even some work on implementation & optimisation (KRIS)
- Disadvantages:
  - only relatively simple DLs have so far been implemented
  - need transitive and functional roles, role hierarchy and GCIs

Idea: extend Baader/Sattler transitive orbits to (transitive and functional) role hierarchy, and internalise GCIs

Implementing a Tableau Reasoner

Results less than 100% satisfying:

- It fails to get *any* answer
  - effectively non-terminating
- Discouraged? – not a bit of it!
  - Sustained by ignorance and naivety, the quest continues

Idea: Implement a highly optimised tableau reasoner

Optimising (Tableau) Reasoners

Performance problems mainly caused by GCIs

- standard "theoretical" technique is to use internalisation:

  $C \sqsubseteq D \rightsquigarrow \top \sqsubseteq (D \sqcup \neg C)$, and

  $(D \sqcup \neg C)$ applied to every individual using a "universal role"

- convenient for proofs (TBox satisfiability can be reduced to concept satisfiability), but hopelessly inefficient in practice
  - over 1,200 GCIs in GALEN ontology
  - resulting search space is impossibly large

Lesson: Theory is not the same as practice!

Optimising (Tableau) Reasoners

Idea: suggested by structure of GALEN KB

- GCIs all of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq D$
- can be rewritten as $C_1 \sqsubseteq D \sqcup \neg(C_2 \sqcap \ldots \sqcap C_n)$
- and "absorbed" into primitive "definition" axiom for C_1
- resulting TBox is "definitorial"
  - no GCIs
  - dealt with via lazy unfolding

Result: close, but no cigar

- search space still too large
- effective non-termination

Optimising (Tableau) Reasoners

Idea: Investigate other optimisations, e.g., from SAT

- simplifications (e.g., Boolean Constraint Propagation)
- semantic branching
- caching
- heuristics
- smart backtracking

Result: (qualified) success!

- "FaCT" reasoner classified GALEN core in <400s

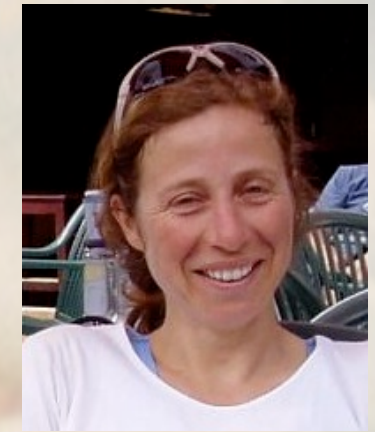
Qualifications

- Only works for GALEN “core”
  - full ontology is much larger & couldn’t be classified by FaCT
- No support for complex roles
  - GRAIL allows for axioms of form $(r \circ s) \sqsubseteq r$
- Weak (cheating?) semantics for inverse roles
  - GRAIL treats them as pre-processing macros: $(r \circ s) \sqsubseteq r \rightsquigarrow (s^- \circ r^-) \sqsubseteq r^-$

Result: progress, but still searching for the Holy Grail!

Extending the Logic

- Qualified Cardinality Restrictions
  - relatively trivial extension to functional roles
- Inverse roles
  - new “double blocking” technique



Result: $\mathcal{SHIQ}$ is born!

- But...
  - still can’t classify GALEN
  - relatively few other applications

Testing and Optimisation

Few ontologies, so testing focused on synthetic data

- hand crafted “hard” tests
- randomly generated tests
- most hand crafted tests easy for optimised systems, so attention focused on randomly generated tests

Result: semantic branching is a crucial optimisation

Semantic Branching

Technique derived from SAT testing

- guess truth values for predicates occurring in disjunctions; use heuristics to select predicate and valuation; e.g.:

  given $\{a : (B \sqcup C), a : (B \sqcup D)\} \subseteq \mathcal{A}$
  guess $a : \neg B$ which implies $a : C$ and $a : D$

Result:

- great for random data, but useless/harmful for ontologies
- e.g., given $\{B \sqsubseteq (C_1 \sqcap C_2)\} \subseteq \mathcal{T}$ we get $a : (\neg C_1 \sqcup \neg C_2)$
- heuristics assume sat:unsat ≈ 50:50; far from true in ontologies

Lesson: careful study of *typical inputs* crucial for successful optimisation

Applications?

- Medical terminologies
- Configuration?
- DB schema design and integration?

Semantic Web: Killer App for DLs

- According to **TBL**, the Semantic Web is

  "... a **consistent logical web of data** ..." in which
  "... information is given **well-defined meaning** …"

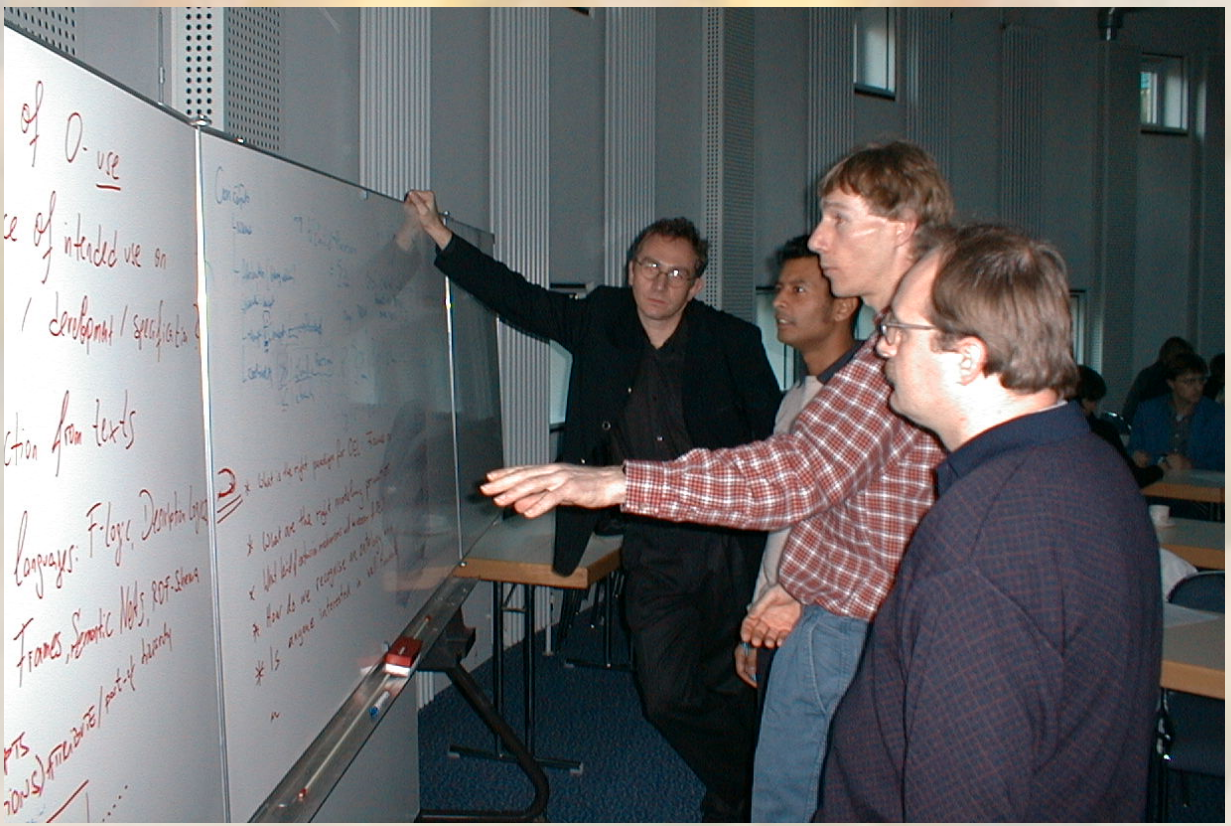
- Idea was to achieve this by adding semantic annotations
  - **RDF** used to provide annotation mechanism
  - **Ontologies** used to provide vocabulary for annotations
- Evolved goal is to transform web into a platform for distributed applications and sharing (linking) data
  - **RDF** provides uniform syntactic structure for data
  - **Ontologies** provide machine readable schemas

Web Ontology Languages

- RDF extended to **RDFS**, a primitive ontology language
  - classes and properties; sub/super-classes (and properties); range and domain (of properties)
- But RDFS **lacks** important **features**, e.g.:
  - existence/cardinality constraints; transitive or inverse properties; localised range and domain constraints, …
- And RDF(S) has “higher order flavour” with no (later **non-standard**) **formal semantics**
  - meaning not well defined (e.g., argument over range/domain)
  - difficult to provide reasoning support

From RDFS to OIL

At **DFKI** in Kaiserslautern at a “Sharing Day on Ontologies” for projects of the ESPRIT LTI programme



From RDFS to OIL

At **DFKI** in Kaiserslautern at a “Sharing Day on Ontologies” for projects of the ESPRIT LTI programme

- Started working with **Deiter Fensel** on development of an “ontology language”
  - On-To-Knowledge project developing web ontology language
  - initially rather informal and based on frames
  - were persuaded to use DL to formalise and provide reasoning

From RDFS to OIL

At **DFKI** in Kaiserslautern at a "Sharing Day on Ontologies" for projects of the ESPRIT LTI programme

- Started working with **Deiter Fensel** on development of an "ontology language"
  - On-To-Knowledge project developing web ontology language
  - initially rather informal and based on frames
  - were persuaded to use DL to formalise and provide reasoning
- Soon joined by **Frank van Harmelen**, and together we developed **OIL**
  - basically just $\mathcal{SHIQ}$ DL with frame-like syntax
  - initially "Manchester" style syntax, but later XML and RDF

From OIL to OWL

- **DARPA DAML** program also developed DAML-ONT
- Efforts “merged” to produce **DAML+OIL**
  - Further development carried out by “Joint EU/US Committee on Agent Markup Languages”

From OIL to OWL

- **DARPA DAML** program also developed DAML-ONT
- Efforts "merged" to produce **DAML+OIL**
  - Further development carried out by "Joint EU/US Committee on Agent Markup Languages"
- DAML+OIL submitted to W3C as basis for standardisation
- **WebOnt** Working Group formed
  - WebOnt developed OWL language based on DAML+OIL
  - OWL became a W3C recommendation
  - OWL extended DAML+OIL with nominals: "Web-friendly" syntax for $\mathcal{SHOIN}$

Was it Worth It?

Was it Worth It?

Ontologies before:

Name	Original Language	de-fined	primi-tive	arti-ficial	$\sum$	de-fined	primi-tive
		concepts				roles	
CKB	SB-ONE	23	57	58	138	2	46
Companies	BACK	70	45	81	196	1	39
FSS	SB-ONE	34	98	75	207	0	47
Espresso	SB-ONE	0	145	79	224	11	41
Wisber	TURQ	50	81	152	283	6	18
Wines	CLASSIC	50	148	237	435	0	10

and of course Galen!

Was it Worth It?

Ontologies after:

Was it Worth It?

Ontologies after:

Welcome to the Protege Ontology Library!

OWL ontologies

- AIM@SHAPE Ontologies: Ontologies pertaining to digital shapes. Source: AIM@SHAPE NoE - Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects.
- amino-acid.owl: A small OWL ontology of amino acids and their properties. Source: Amino Acid Ontology Web site.
- Basic Formal Ontology (BFO)
- bhakti.owl: An OWL ontology for the transcendental states of consciousness experienced by practitioners of bhakti-yoga, a form of Vedic consciousness engineering.
- Biochemical Ontologies: Over 30 ontologies for knowledge representation and reasoning across scientific domains. Ontologies are normalized into non-disjoint primitive skeletons and

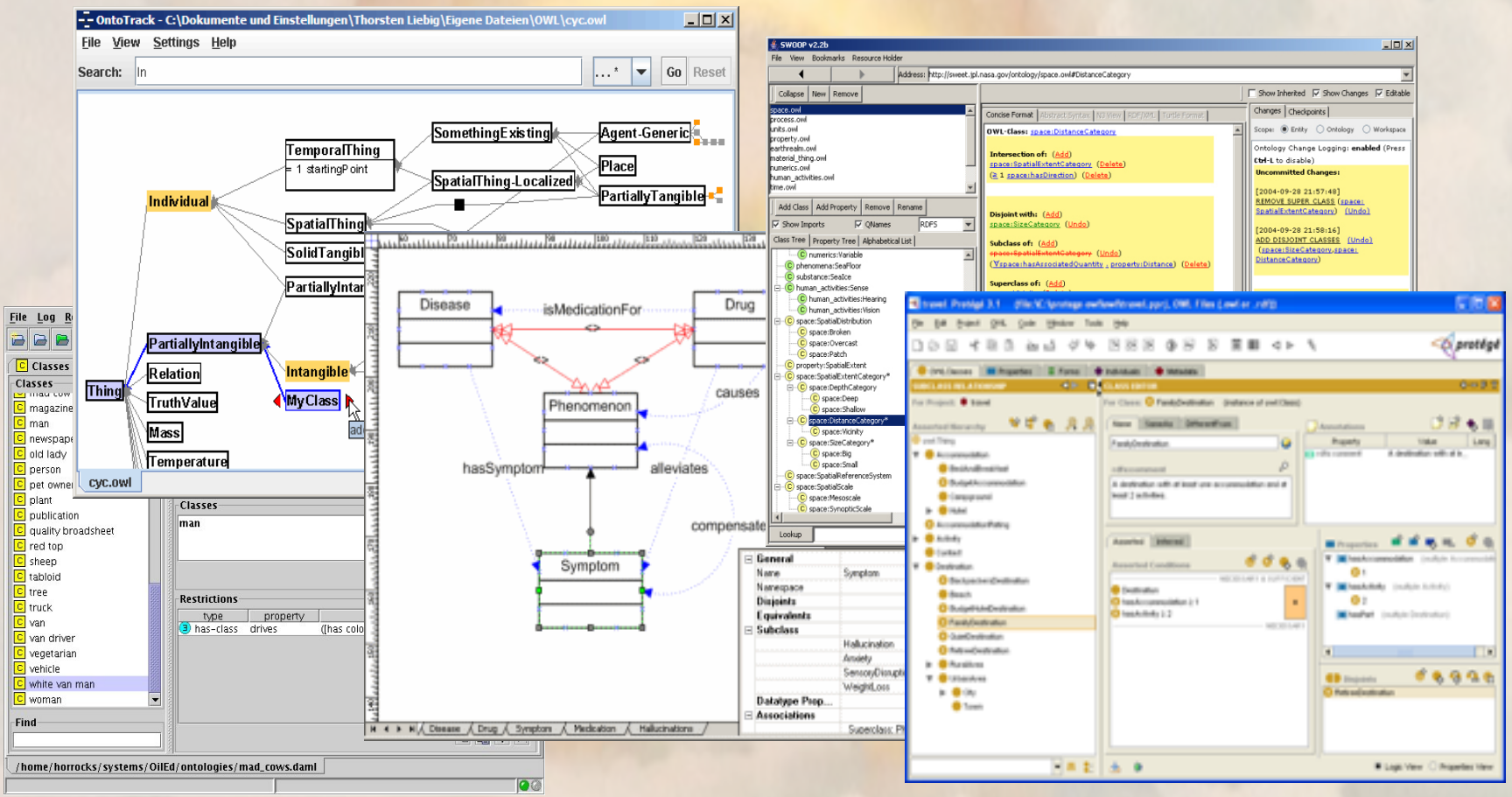
Was it Worth It?

Tools before:

```
> (load-tkb "demo.kb" :verbose T)
  ..................................
  ..........................
> (classify-tkb :mode :stars)
  pppppppppppppppppccpcppcccpcppcpcppcccppccpcp
  pcccccppcpcppcccp
  T
> (direct-supers 'MAN)
  (c[HUMAN] c[MALE])
>
```

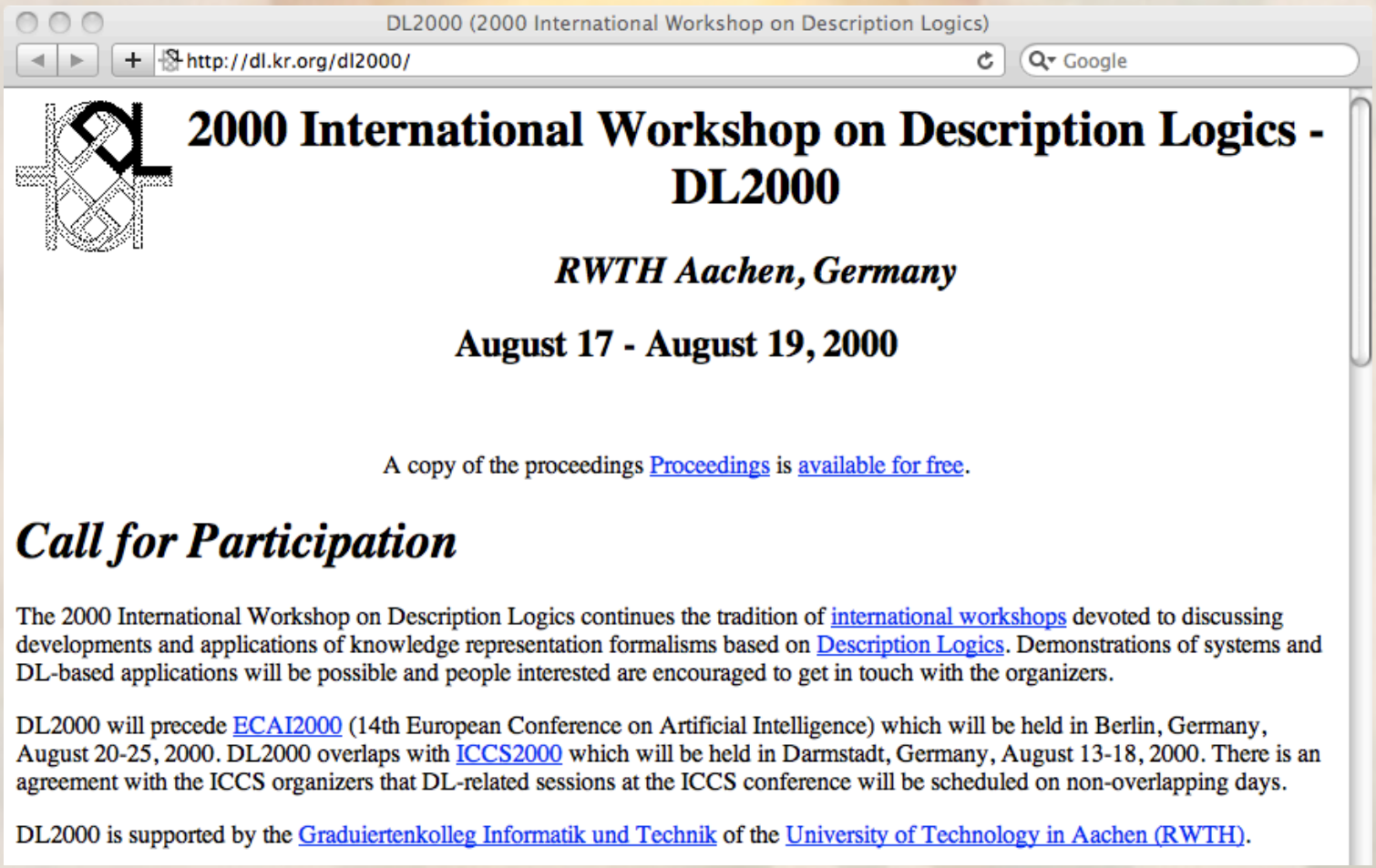
Was it Worth It?

Tools after:



Was it Worth It?

"**Profile**" before:



Was it Worth It?

"**Profile**" after:

Where the Rubber Meets the Road

- DL ontologies/reasoners only useful in practice if we can deal with large ontologies and/or large data sets

  We made a sale; can we deliver the goods?

- Unfortunately, OWL/$\mathcal{SHOIN}$ is highly intractable
  - satisfiability is **NEXPTIME-complete** w.r.t. schema
  - and **NP-Hard** w.r.t. data (upper bound open)
- Problem addressed in practice by
  - New algorithms and optimisations
  - Use of tractable fragments (aka **profiles**)

New Algorithms and Optimisations

- HyperTableau
- Completely defined concepts
- Algebraic methods
- Nominal absorption
- Heuristics
- Caching and individual reuse
- Optimised blocking
- ...

New Algorithms and Optimisations

- HyperTableau
- Completely defined concepts
- Algebraic methods
- Nominal absorption
- Heuristics
- Caching and individual reuse
- Optimised blocking
- ...

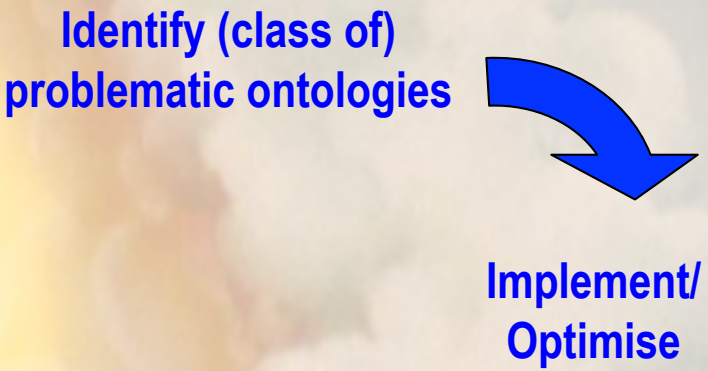
New Algorithms and Optimisations

- HyperTableau
- Completely defined concepts
- Algebraic methods
- Nominal absorption
- Heuristics
- Caching and individual reuse
- Optimised blocking
- ...

Identify (class of) problematic ontologies

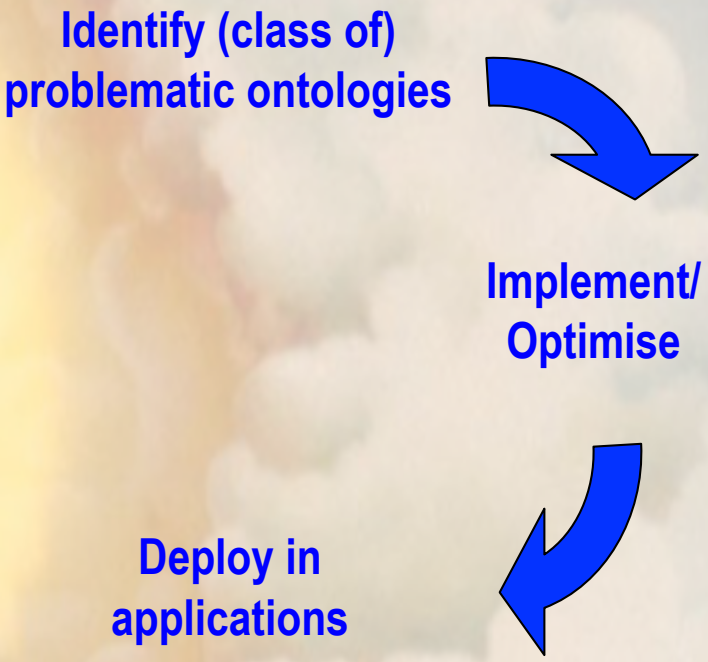
New Algorithms and Optimisations

- HyperTableau
- Completely defined concepts
- Algebraic methods
- Nominal absorption
- Heuristics
- Caching and individual reuse
- Optimised blocking
- ...



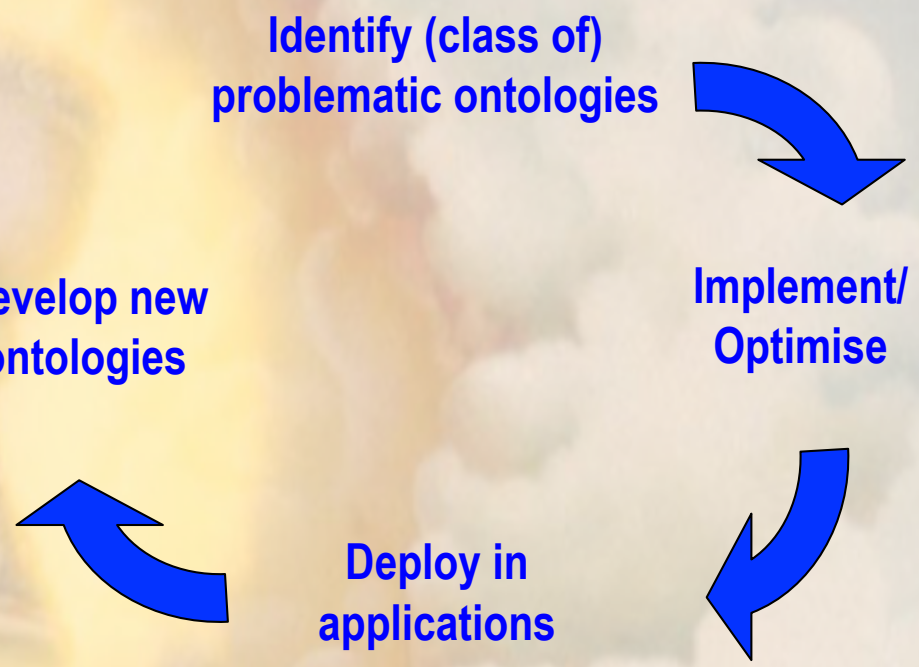
New Algorithms and Optimisations

- HyperTableau
- Completely defined concepts
- Algebraic methods
- Nominal absorption
- Heuristics
- Caching and individual reuse
- Optimised blocking
- ...



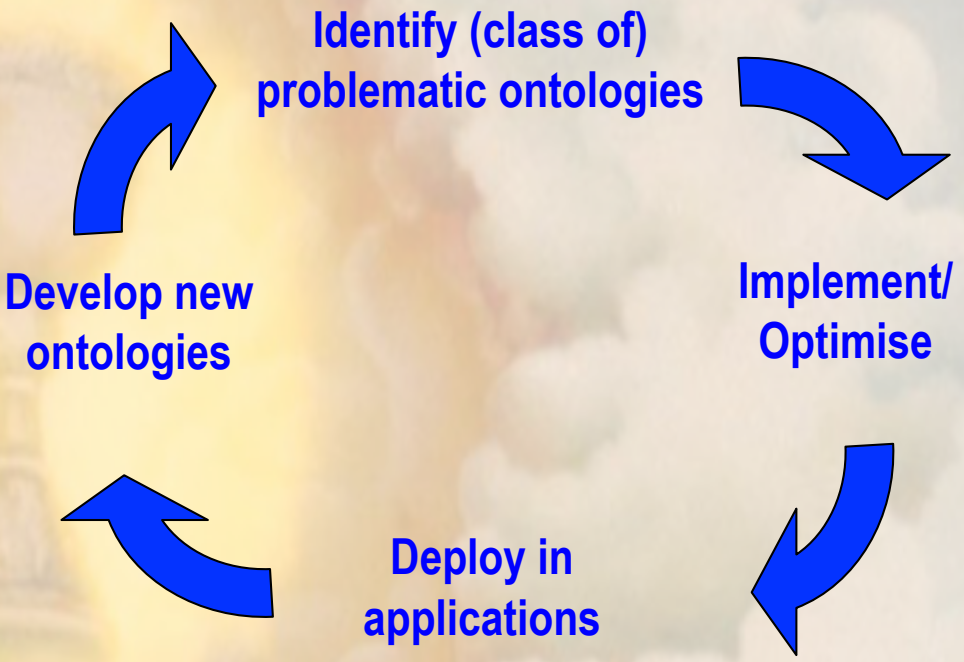
New Algorithms and Optimisations

- HyperTableau
- Completely defined concepts
- Algebraic methods
- Nominal absorption
- Heuristics
- Caching and individual reuse
- Optimised blocking
- ...



New Algorithms and Optimisations

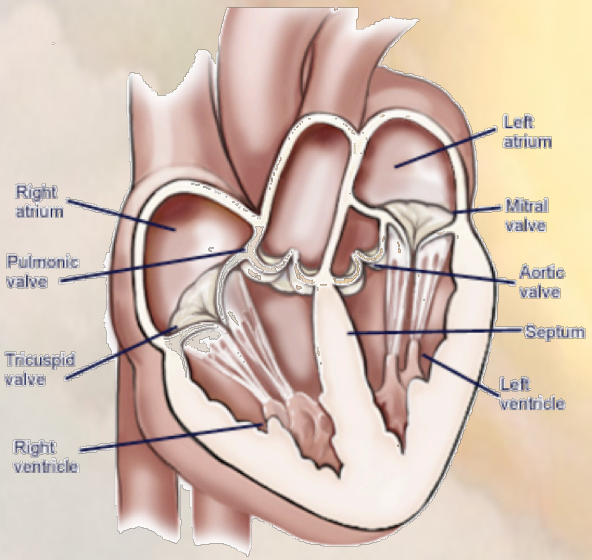
- HyperTableau
- Completely defined concepts
- Algebraic methods
- Nominal absorption
- Heuristics
- Caching and individual reuse
- Optimised blocking
- ...



Scalability Issues

Problems with very **large and/or cyclical ontologies**

- Ontologies may define 10s/100s of thousands of terms
- Potentially vast number (n^2) of tests needed for classification
- Each test can lead to construction of *very* large models



$$\mathsf{LeftSide} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{AorticValve}$$
$$\mathsf{LeftSide} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{MitralValve}$$
$$\mathsf{AorticValve} \sqsubseteq \exists \mathsf{hasConnection}.\mathsf{LeftVentircle}$$
$$\mathsf{MitralValve} \sqsubseteq \exists \mathsf{hasConnection}.\mathsf{LeftVentircle}$$
$$\mathsf{LeftVentricle} \sqsubseteq \exists \mathsf{isDivisionOf}.\mathsf{LeftSide}$$

Scalability Issues

Problems with **large data sets** (ABoxes)

- Main reasoning problem is (conjunctive) query answering, e.g., retrieve all patients suffering from vascular disease:
  $Q(x) \leftarrow \mathsf{Patient}(x) \wedge \mathsf{suffersFrom}(x, y) \wedge \mathsf{VascularDisease}(y)$
- Decidability still open for OWL, although minor restrictions (on cycles in non-distinguished variables) restore decidability
- Query answering reduced to standard decision problem, e.g., by checking for each individual x if $\mathcal{O} \models Q(x)$
- Model construction starts with *all* ground facts (data)

Typical applications may use data sets with **10s/100s of millions** of individuals (or more)

OWL 2

- OWL recommendation now updated to **OWL 2**
  (I didn't learn my lesson!)
- OWL 2 based on $\mathcal{SROIQ}$
  - includes complex role inclusions, so properly includes GRAIL
- OWL 2 also defines several **profiles** – fragments with desirable computational properties
  - **OWL 2 EL** targeted at very large ontologies
  - **OWL 2 QL** targeted at very large data sets

OWL 2 EL

- A (near maximal) fragment of OWL 2 such that
  - Satisfiability checking is in PTime (**PTime-Complete**)
  - Data complexity of query answering also PTime-Complete
- Based on $\mathcal{EL}$ family of description logics
- Can exploit **saturation** based reasoning techniques
  - Computes complete classification in “one pass”
  - Computationally optimal (PTime for EL)
  - Can be extended to Horn fragment of OWL DL

OWL 2 QL

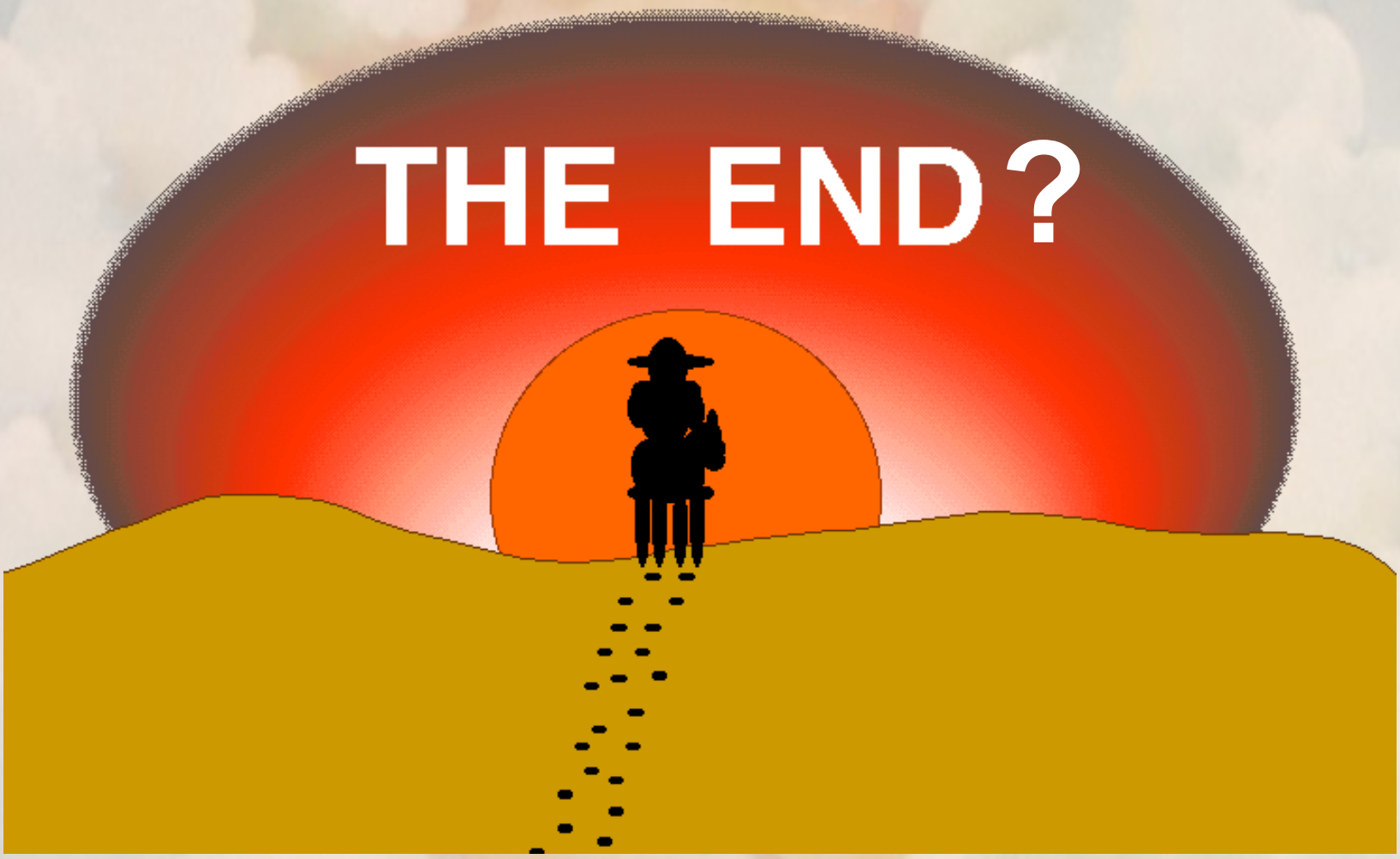
- A (near maximal) fragment of OWL 2 such that
  - Data complexity of conjunctive query answering in **AC^0**
- Based on **DL-Lite** family of description logics
- Can exploit **query rewriting** based reasoning technique
  - Computationally optimal
  - Data storage and query evaluation can be delegated to standard RDBMS
  - Can be extended to more expressive languages (beyond AC^0) by using “hybrid” techniques or by delegating query answering to a Datalog engine

So What About GALEN?

- SOTA (hyper-) tableau reasoners still fail
  - construct huge models
  - exhaust memory or effective non-termination

- BUT, in 2009, new CB reasoner developed by Yevgeny Kazakov
  - used highly optimised implementation of saturation based algorithm for Horn-$\mathcal{SHIQ}$
  - can classify complete GALEN ontology in <10s


THE END


THE END?

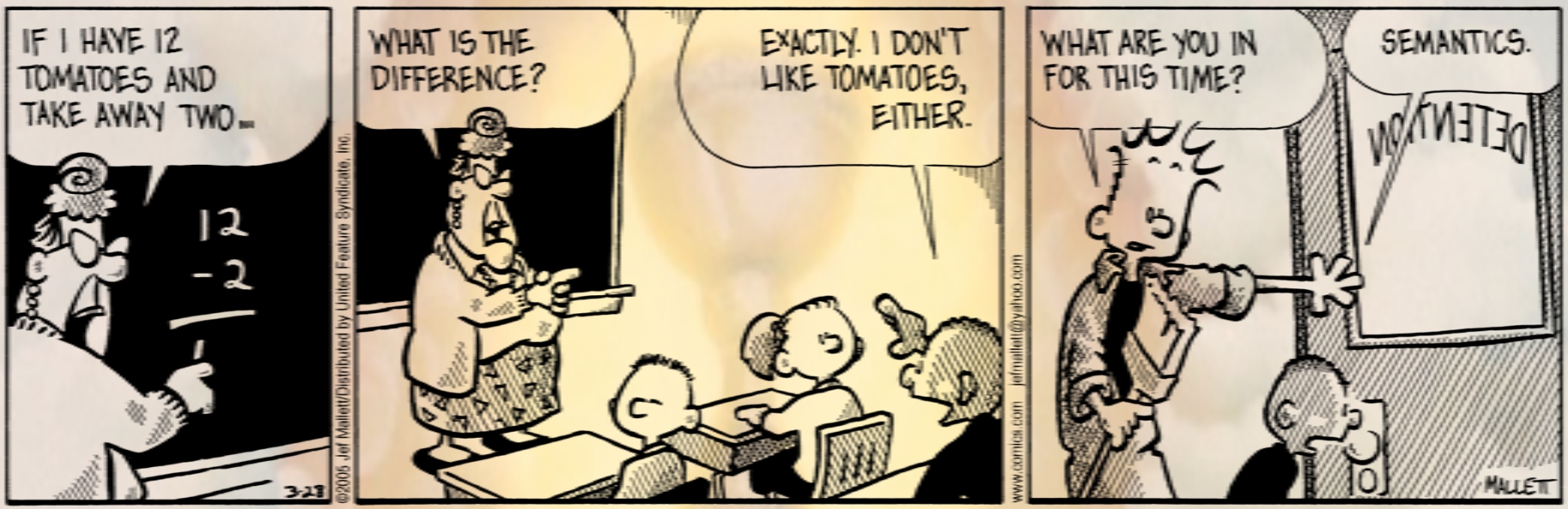
Ongoing Research

- Optimisation
- Query answering
- Second order DLs
- Temporal DLs
- Fuzzy/rough concepts
- Diagnosis and repair
- Modularity, alignment and integration
- Integrity constraints
- ...

Ongoing Standardisation Efforts

- Standardised query language
  - SPARQL standard for RDF
  - Currently being extended for OWL, see *http://www.w3.org/TR/sparql11-entailment/*
- RDF
  - Revision currently being considered, see *http://www.w3.org/2009/12/rdf-ws/*

Thank you for listening



FRAZZ: © Jeff Mallett/Dist. by United Feature Syndicate, Inc.

Any questions?