Accessing Existing Distributed Science Archives as RDF Models

Alasdair J G Gray[1]
Norman Gray[2]
Iadh Ounis[1]

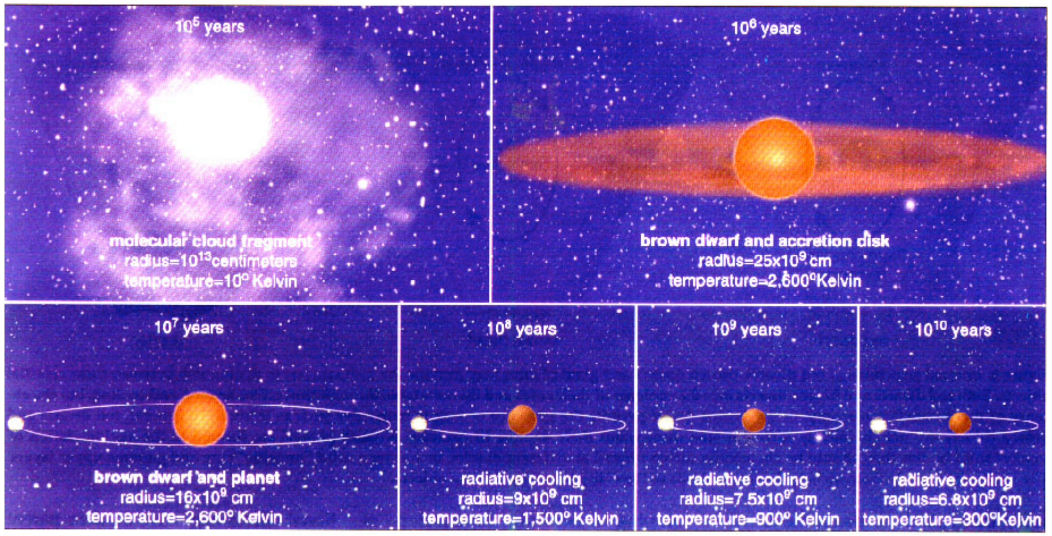
[1]Computing Science, University of Glasgow
[2]Physics and Astronomy, University of Leicester

Outline

- Motivating science problems
- A data integration approach
- RDF and SPARQL
- Extracting scientific data
- Performance results
- Conclusions

Searching for Brown Dwarfs

- Data sets:
  - Near Infrared, 2MASS/UK Infrared Deep Sky Survey
  - Optical, APMCAT/Sloan Digital Sky Survey
- Complex colour/motion selection criteria
- Similar problems
  - Halo White Dwarfs



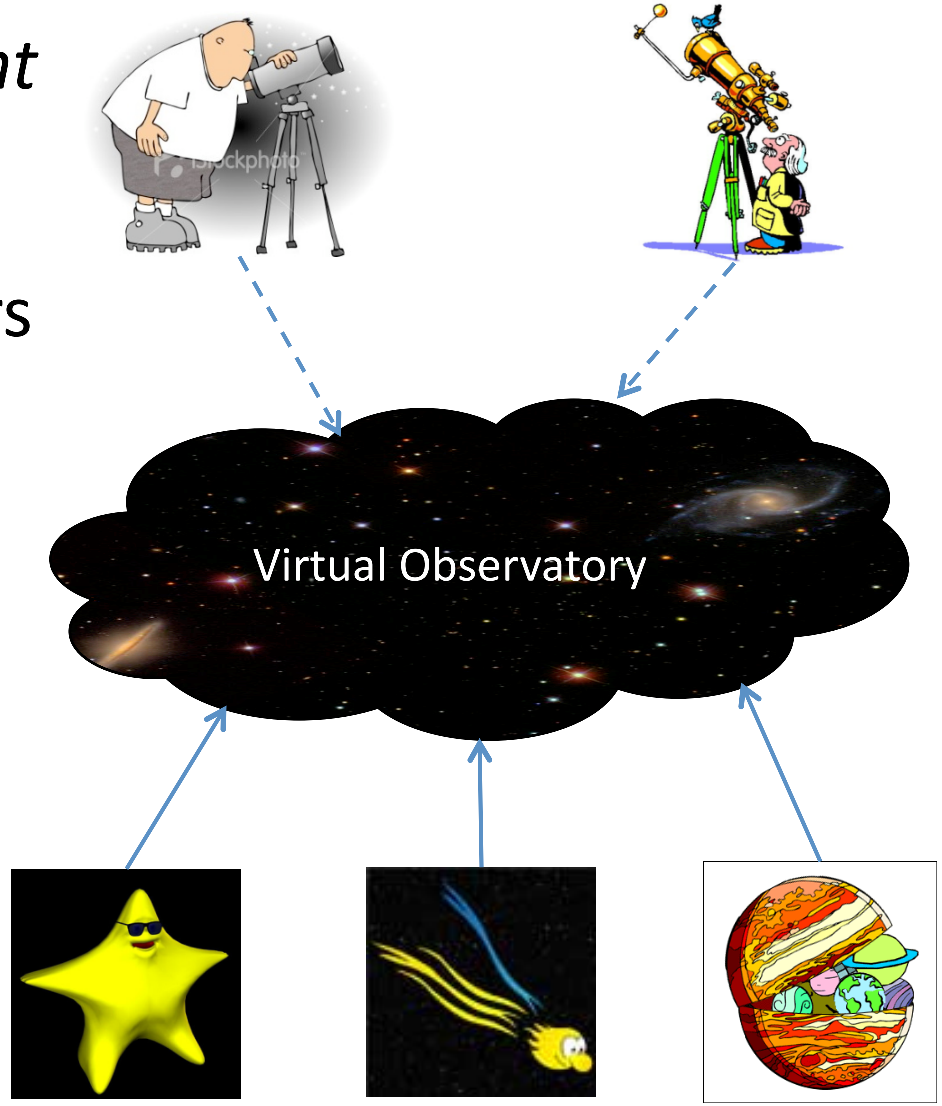
Deep Field Surveys

- Observations in multiple wavelengths
  - Radio to X-Ray
- Searching for new objects
  - Galaxies, stars, etc
- Requires correlations across many catalogues
  - ISO
  - Hubble
  - SCUBA
  - etc

The Problem

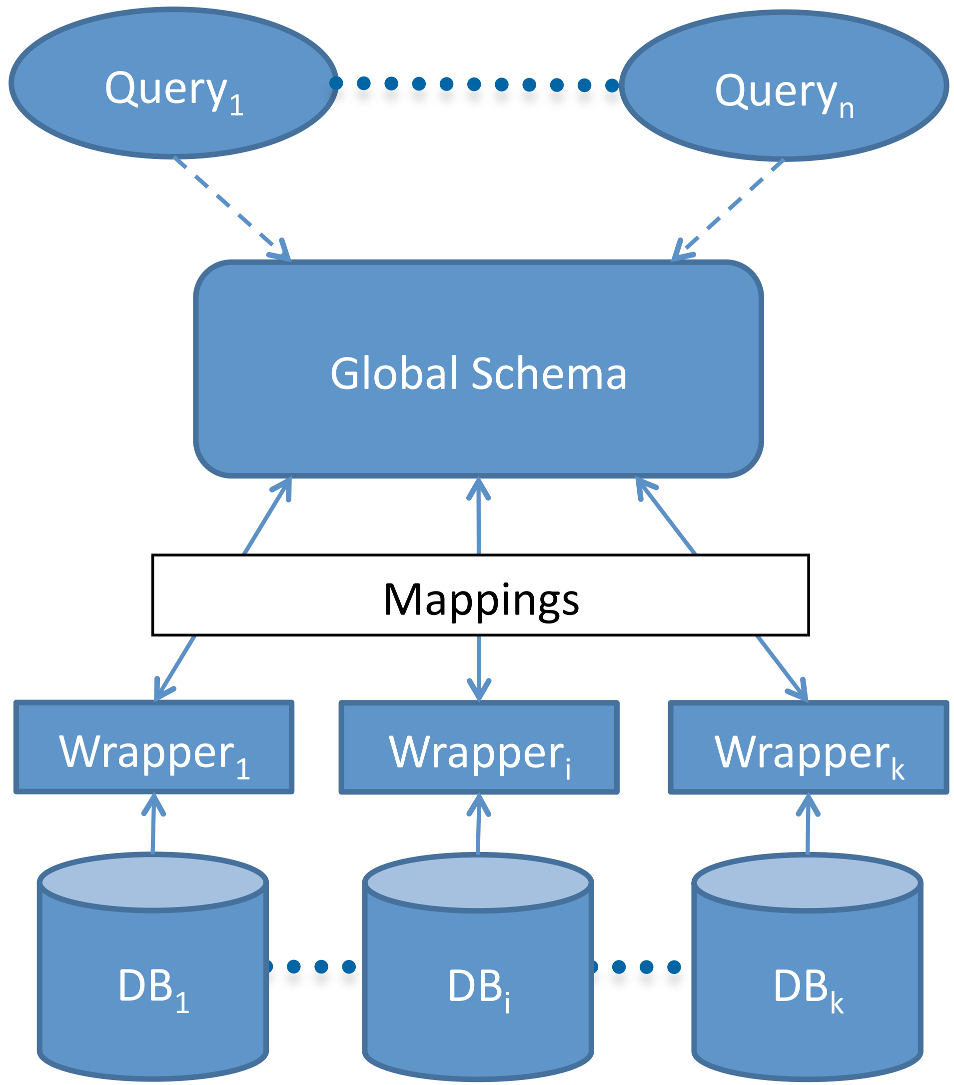
Locate and combine relevant data

- Heterogeneous publishers
  - Archive centres
  - Research labs
- Heterogeneous data
  - Relational
  - XML
  - Files



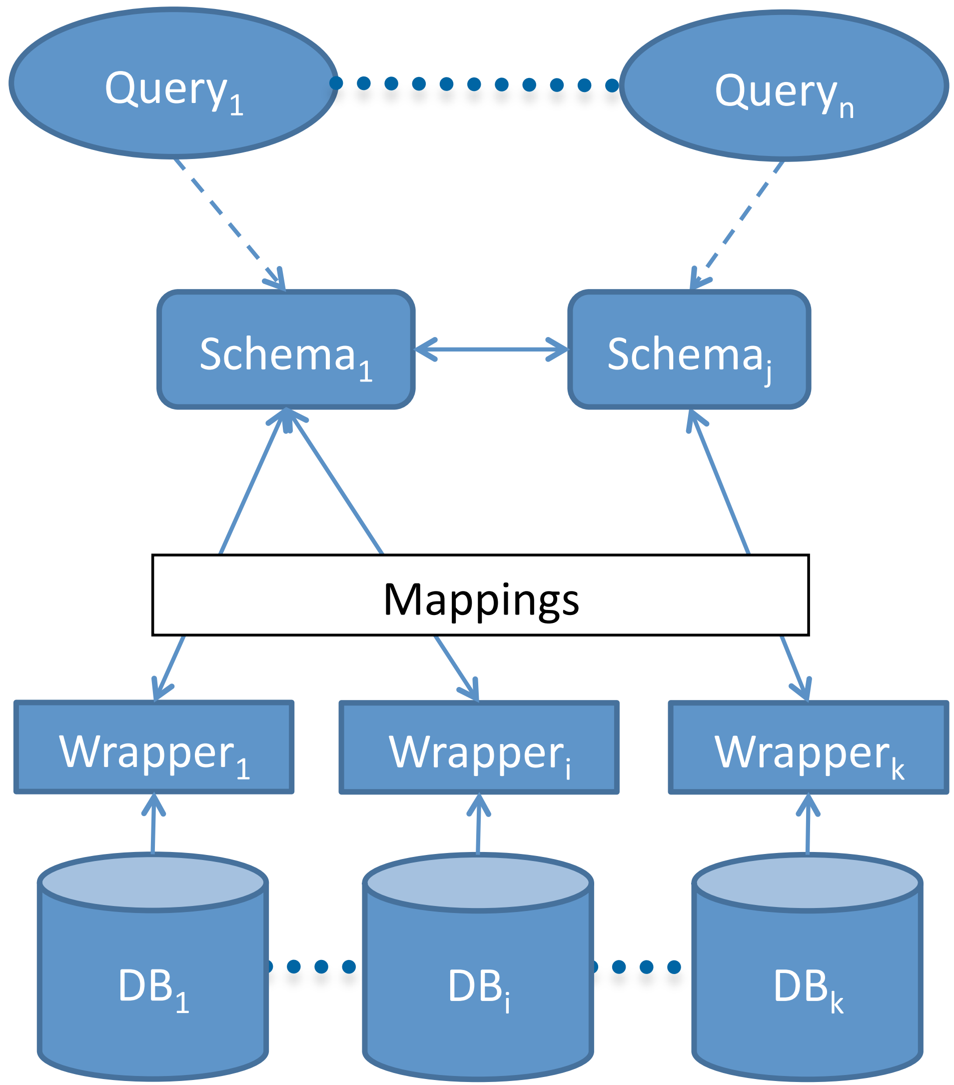
Generic Data Integration Approach

- Heterogeneous sources
  - Autonomous
  - Local schemas
- Homogeneous view
  - Mediated global schema
- Mapping
  - LAV: local-as-view
  - GAV: global-as-view

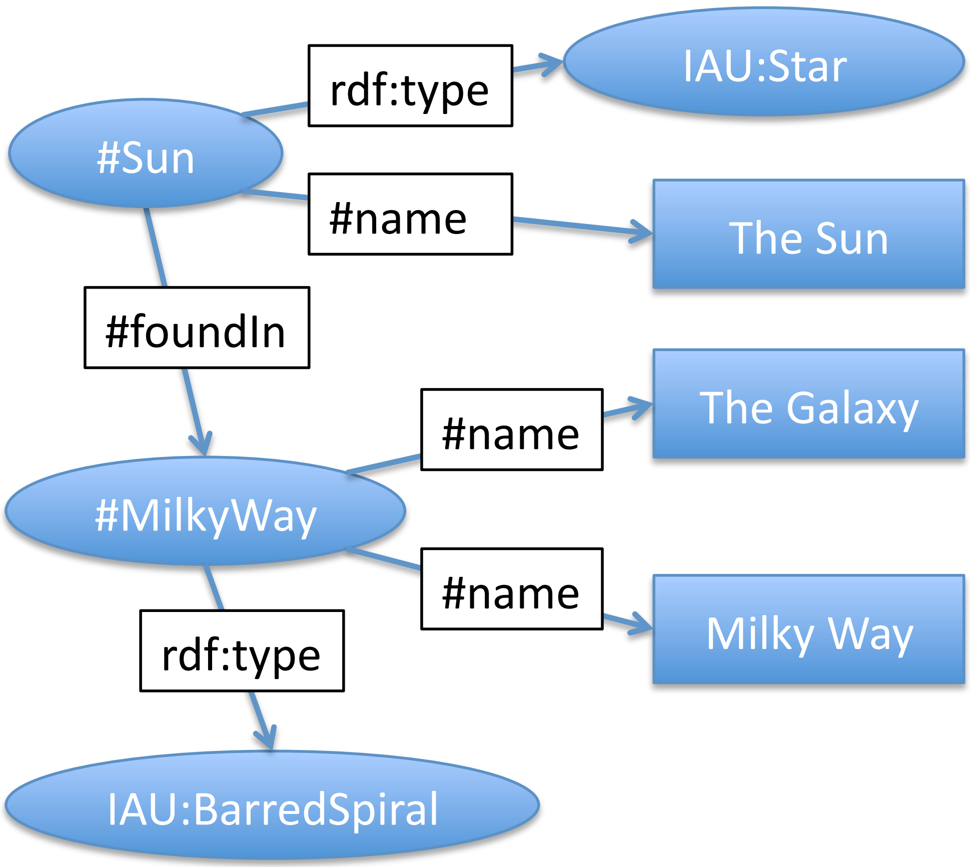


P2P Data Integration Approach

- Heterogeneous sources
  - Autonomous
  - Local schemas
- Heterogeneous views
  - Multiple schemas
- Mapping
  - Between pairs of schema
  - Network of links
- Require common integration data model

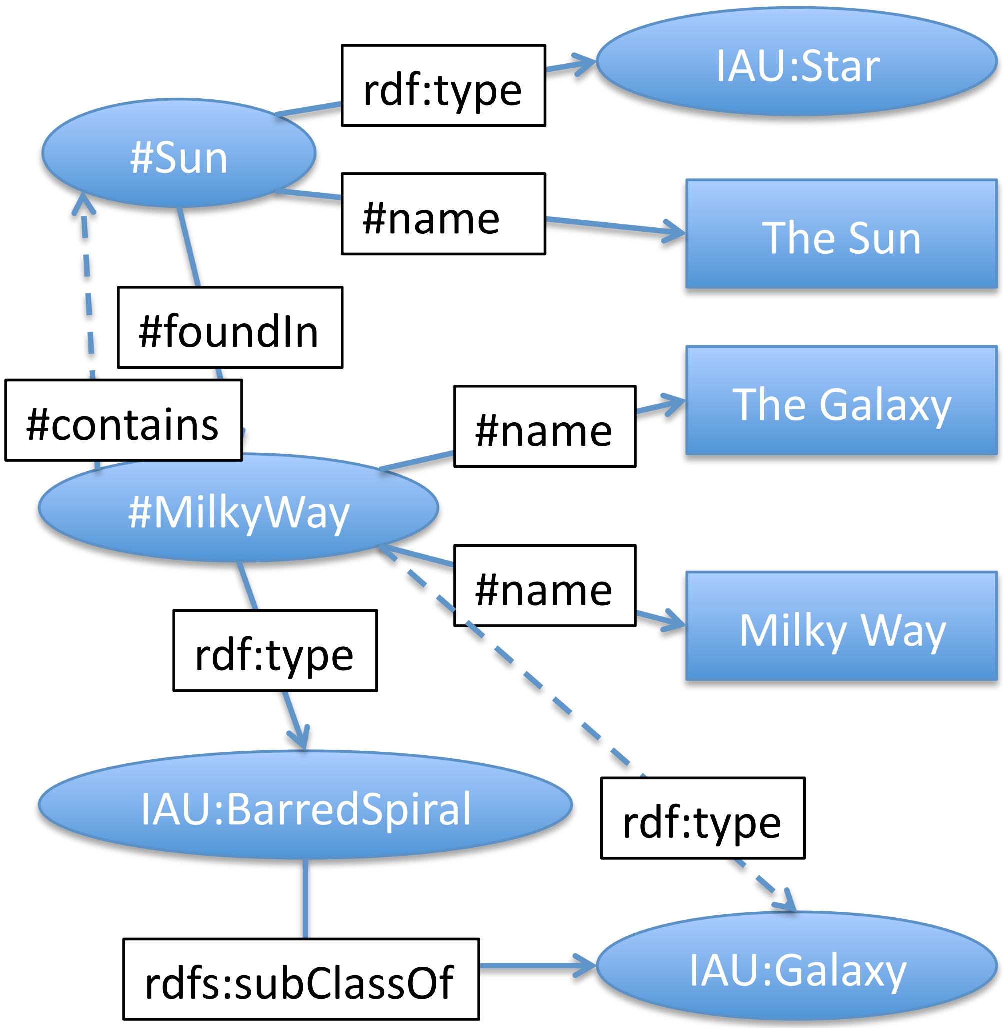


Resource Description Framework (RDF)



- W3C standard
- Designed as a metadata data model
- Make statements about resources
- Contains semantic details
- Ideal for linking distributed data

Reasoning

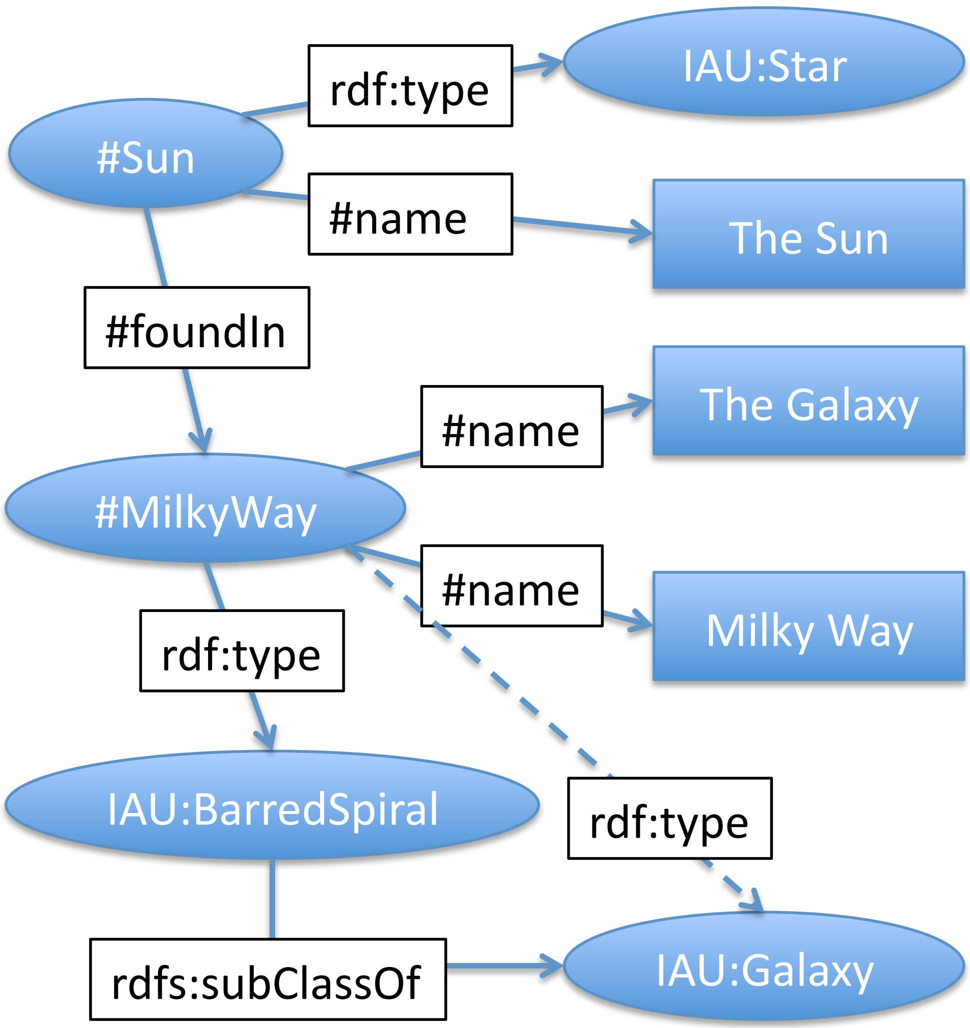


- Infers knowledge from RDF(S) statements
  - Sub classing
- OWL: extends what can be expressed
  - Inverse predicates
  - Equivalence
  - etc

SPARQL

- Declarative query language
  - Select returned data
    - Graph or tuples
    - Attributes to return
  - Describe structure of desired results
  - Filter data
- W3C standard
- Syntactically similar to SQL

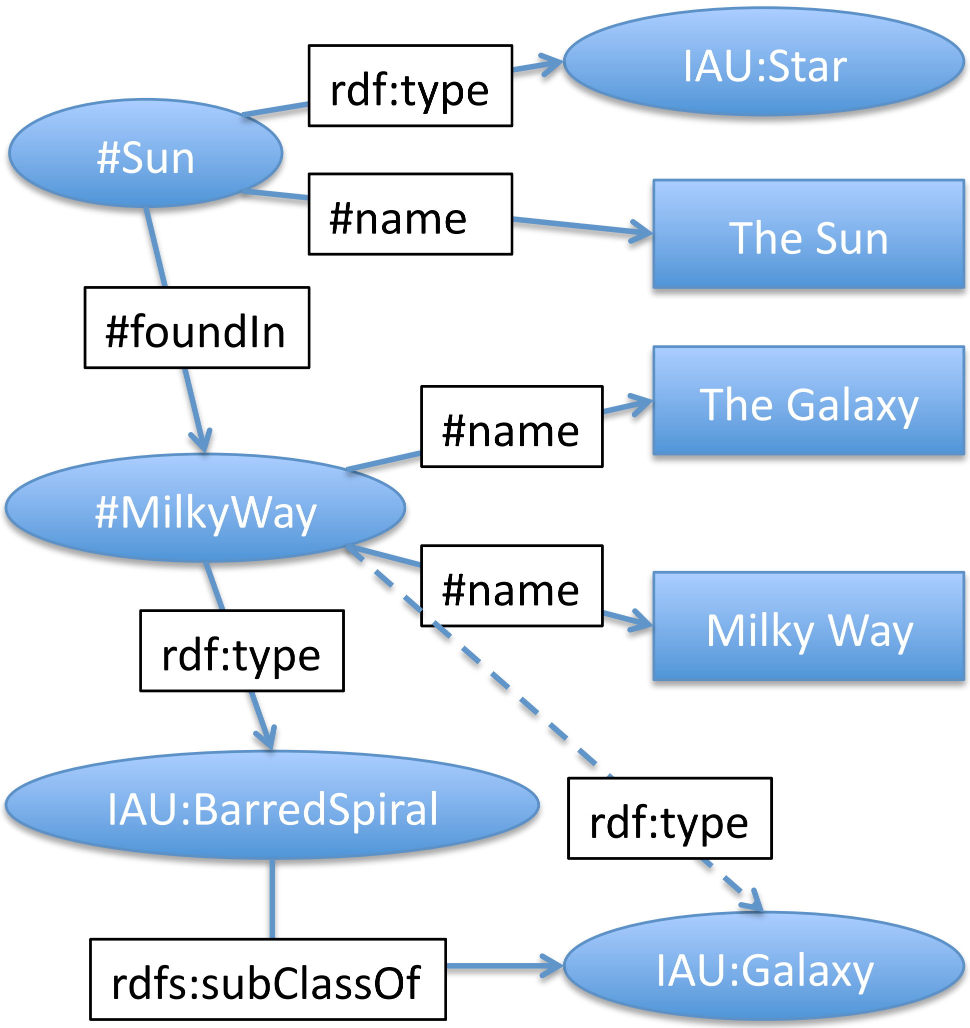
Querying RDF with SPARQL



Find the name of the galaxy which contains a star with the name The Sun

```
SELECT ?galName
WHERE {
?gal a IAU:Galaxy ;
  #name ?galName .
?star a IAU:Star ;
  #name ?starName ;
  #foundIn ?gal .
FILTER REGEX(?
  starName, “The Sun”)
}
```

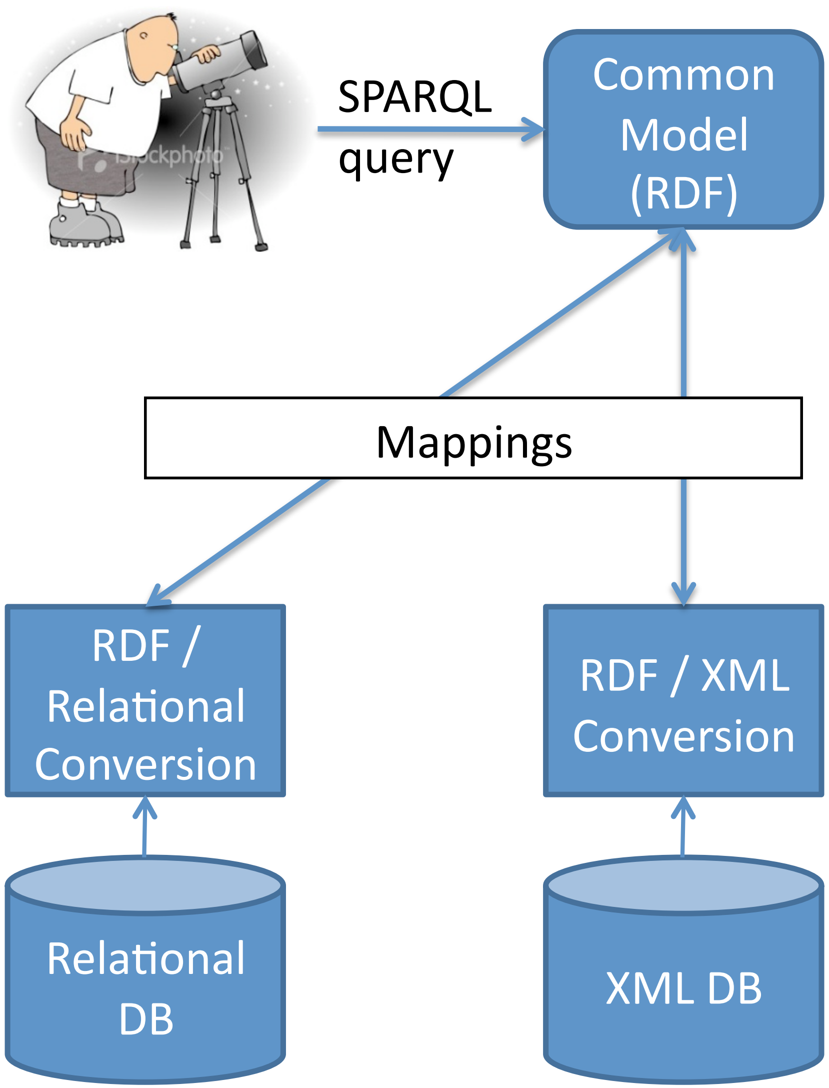
Querying RDF with SPARQL



Find the name of the galaxy which contains a star with the name The Sun

?galName
The Galaxy
Milky Way

Integrating Using RDF



- Data resources
  - Expose schema and data as RDF
  - Need a SPARQL endpoint
- Allows multiple
  - Access models
  - Storage models
- Easy to relate data from multiple sources

Accessing Relational Sources as RDF

Data Dump

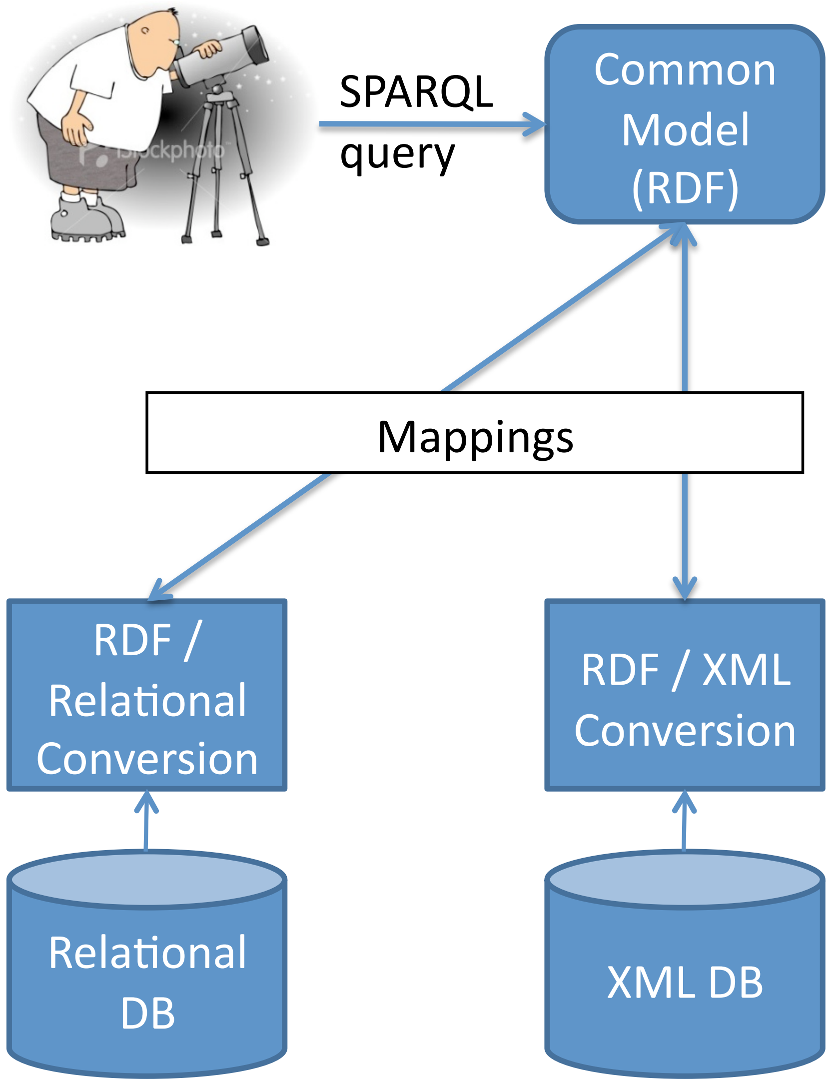
- Data stored as RDF
  - Original relational source is replicated
  - Data can become stale
- Native SPARQL query support
- Existing RDF stores
  - Jena
  - Seasame

On-the-fly Translation

- Data stored as relations
- Native SQL support
  - Highly optimised access methods
- SPARQL queries must be translated
- Existing translation systems
  - D2RQ/D2R Server
  - SquirrelRDF

System Hypothesis

It is viable to perform *on-the-fly* conversions from existing science archives to RDF to facilitate data access from a data model that a scientist is familiar with



Test Data

- SuperCOSMOS Science Archive (SSA)
  - Data extracted from scans of Schmidt plates
  - Stored in a relational database
  - About 4TB of data, detailing 6.4 billion objects
  - Fairly typical of astronomical data archives
- Schema designed using 20 real queries
- Personal version contains
  - Data for a specific region of the sky
  - About 0.1% of the data, ~500MB

Analysis of Test Data

- About 500MB in size
- Organised in 14 Relations
  - Number of attributes: 2 – 152
    - 4 relations with more than 20 attributes
  - Number of rows: 3 – 585,560
  - Two views
    - Complex selection criteria in view

Real Science Queries

Query 5

Find the positions and (B,R,I) magnitudes of all star-like objects within delta mag of 0.2 of the colours of a quasar of redshift 2.5 < z < 3.5

```
SELECT TOP 30 ra,
  dec, sCorMagB,
  sCorMagR2,
  sCorMagI
FROM ReliableStars
WHERE (sCorMagB-
  sCorMagR2 BETWEEN
  0.05 AND 0.80) AND
  (sCorMagR2-
  sCorMagI BETWEEN
  -0.17 AND 0.64)
```

Analysis of Test Queries

Query Feature	Query Numbers
Arithmetic in body	1-5, 7, 9, 12, 13, 15-20
Arithmetic in head	7-9, 12, 13
Ordering	1-8, 10-17, 19, 20
Joins (including self-joins)	12-17, 19
Range functions (e.g. Between, ABS)	2, 3, 5, 8, 12, 13, 15, 17-20
Aggregate functions (including Group By)	7-9, 18
Math functions (e.g. power, log, root)	4, 9, 16
Trigonometry functions	8, 12
Negated sub-query	18, 20
Type casting (e.g. Radians to degrees)	7, 8, 12
Server functions	10, 11

Expressivity of SPARQL

Features

- Select-project-join
- Arithmetic in body
- Conjunction and disjunction
- Ordering
- String matching
- External function calls
  (extension mechanism)

Limitations

- Range shorthands
- Arithmetic in head
- Math functions
- Trigonometry functions
- Sub queries
- Aggregate functions
- Casting

Analysis of Test Queries

Query Feature	Query Numbers	
Arithmetic in body	1-5, 7, 9, 12, 13, 15-20	✓
Arithmetic in head	7-9, 12, 13	✗
Ordering	1-8, 10-17, 19, 20	✓
Joins (including self-joins)	12-17, 19	✓
Range functions (e.g. Between, ABS)	2, 3, 5, 8, 12, 13, 15, 17-20	✓
Aggregate functions (including Group By)	7-9, 18	✗
Math functions (e.g. power, log, root)	4, 9, 16	✗
Trigonometry functions	8, 12	✗
Negated sub-query	18, 20	✗
Type casting (e.g. radians to degrees)	7, 8, 12	✗
Server functions	10, 11	✗

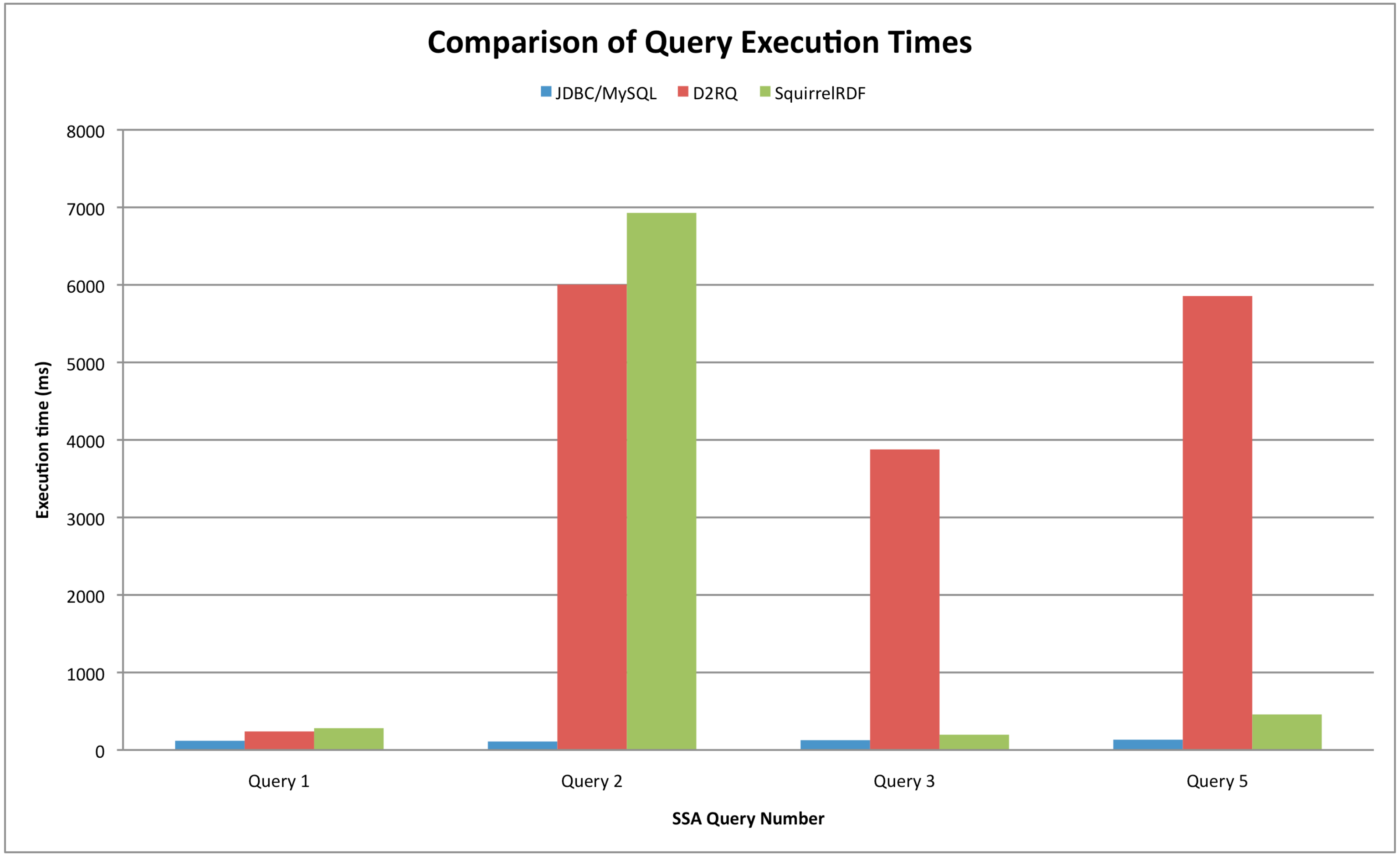
Expressible queries: 1, 2, 3, 5, 6, 14, 15, 17, 19

Experimental Setup

- Machine
  - Intel Core 2 Duo 2.4GHz
  - 2GB RAM
  - Windows XP
  - Java 1.5
- Software
  - MySQL 5.0.51a
  - D2RQ 0.5.1
  - SquirrelRDF 0.1

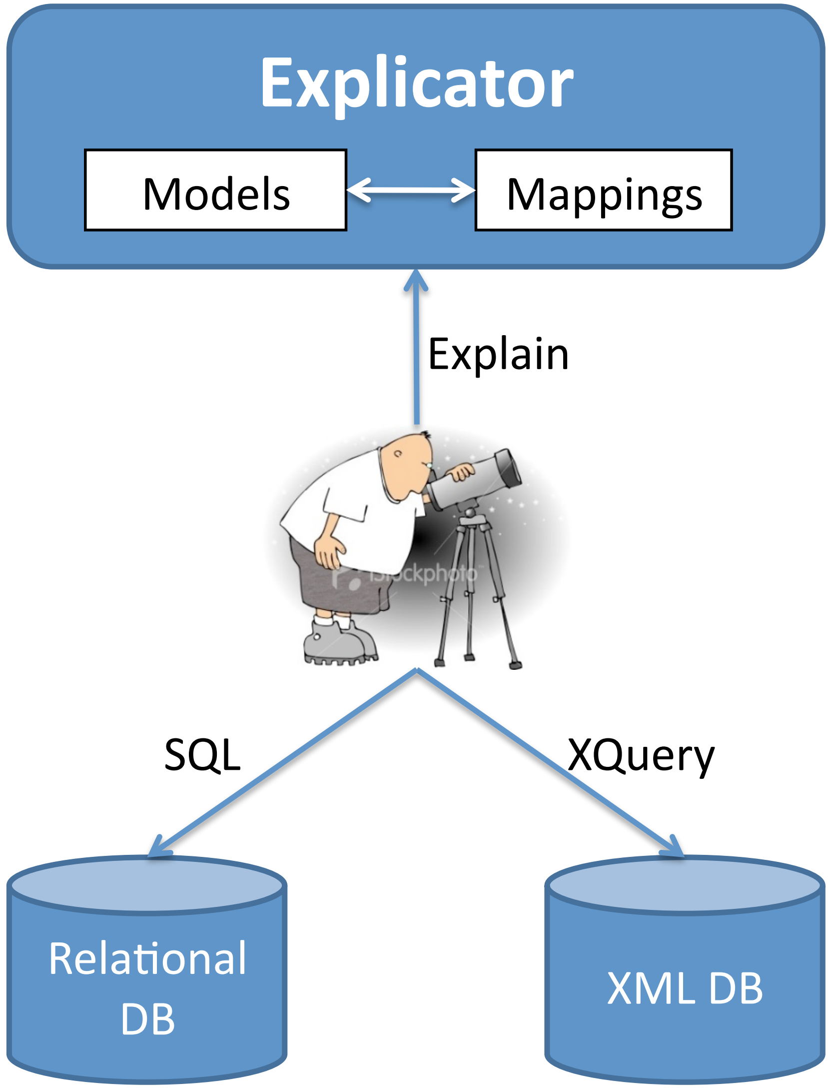
- Only 4 queries completed within 2 hours

Performance Results



A New Approach

- Exploit query engines and data structure of underlying data sources
- Aid user query generation by *explaining* source data model in terms of known data model
- Data extracted in native model



Conclusions

RDF	Relational
Ragged data	Structured data
Small to medium data volumes	Large data volumes
Reasoning over the data	Extracting specific data

- SPARQL: Not expressive enough for science
- Query Converters: Poor performance
- Proposed new approach
  - – RDF to understand data models
  - – Native query engines for data extraction