Semantic Integration: A Survey Of Ontology-Based Approaches

Discussion Lead: Devyani McLaren
Presenter: Nalin Munshi

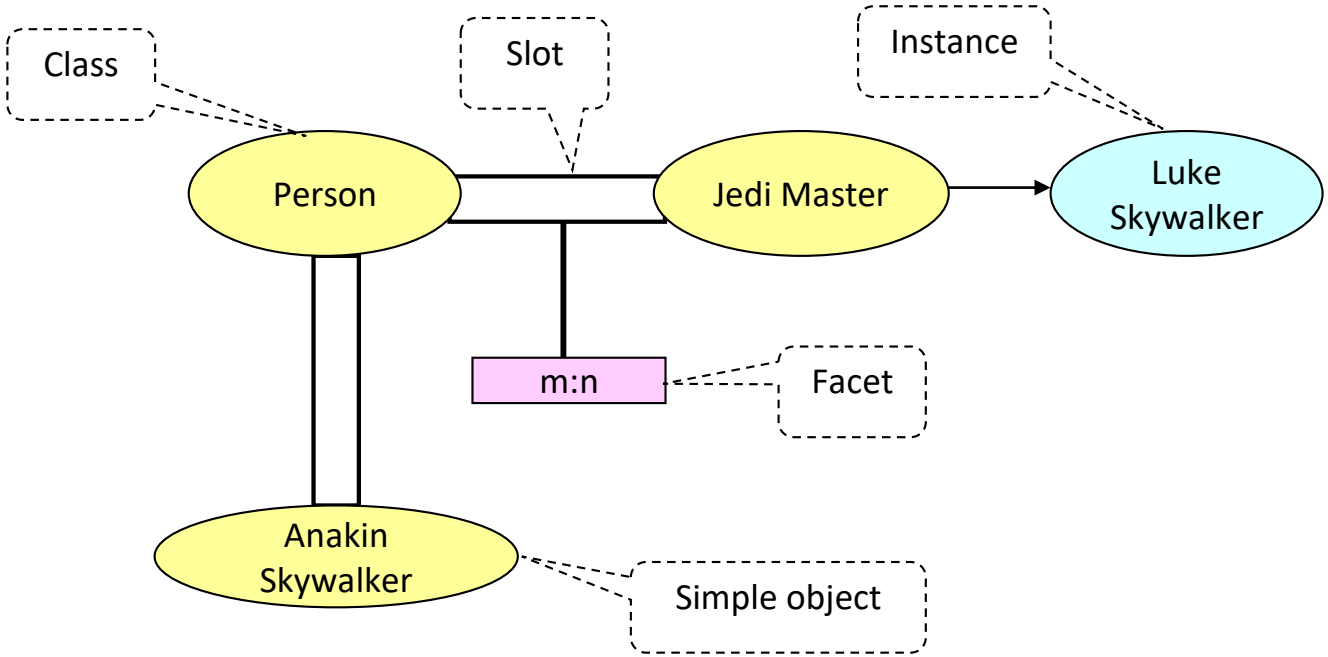
Semantic Integration

- Process of integrating data or information from different sources by mapping their **meaning** and **structure** to a common semantic framework.
- Required in case semantic heterogeneity exists
  - Data Models: relational, hierarchical etc.
  - Data Format: XML, jSON etc.
  - …

Ontology

- An ontology in a database is a way of organizing and defining concepts and their relationships within a specific domain.
- Example, a database for a music streaming service like Scotify (so I don’t get sued!).
- Some **concepts** as part of the ontology can be *artist*, *album*, *song*, *genre* etc.
- Some **relations** among them could be:
  - *Artist* can have multiple *albums*
  - An *album* can have more than one *song*

Ontology example



Solution to Semantic Heterogeneity - Ontology

- Make data standardized and interoperable
- Mapping the data from different sources
- Automated reasoning and inference
- Resolve and surface semantic inconsistencies

Discovering Mappings – 2 Approaches

- Discovering mappings between concepts and entities in different ontologies.
  - Approach #1:
    - General upper ontology agreed upon by developers of different apps
    - The upper ontology extended and used in different apps
    - The extensions should be consistent with the common “grounding”
  - Approach #2:
    - Using heuristics or ML
    - Use structure, definition of concepts etc. to find mappings

Discussion Q #1

“An ontology is some formal description of a domain of discourse, intended for sharing among different applications, and expressed in a language that can be used for reasoning” - Noy 2004

What are some ontologies you’ve encountered in your work/life?

Quick discussion! (~5-7 mins)

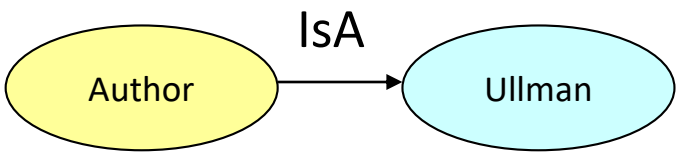
Pair then share

Obi-Wan: Ontology-Based RDF Integration of Heterogeneous Data

Discussion Lead: Devyani McLaren
Presenter: Nalin Munshi

What is RDF?

- RDF stands for Resource Description Framework. It's from the W3C consortium to handle how to store complex data models.
- A way to store things like ontologies or XML. It works for relational data, too.
- The basic form of RDF is a Subject, Object, Predicate triple
- So in the previous ontology example, you might say



Subject: Author Object: Ullman Predicate: IsA

What is the goal of this paper?

- For RDF data sources, integrating data is not as simple as just rewriting and answering the queries using views
- You also have to make it so that you can add about the **reasoning** in the ontology.
- In other words, you have to add *data* and *rules*
- The goal of this paper is to describe how to show how to make these rules work.

OBI-WAN

- Novel mediator following the Ontology-Based Data access (OBDA) paradigm.
- Integrates data sources of many data models under an interface based on RDF graphs and ontologies.
- RDF queries not only over the data but also over the integration ontologies.
- SPARQL is the query language for RDF

Discussion Q#2

In what domains/topics are ontologies suitable or not suitable? (Ehsan)

- In the age of deep learning, do we still need ontologies?

Pairs

OBI-WAN in context of mappings

- Local As View (LAV)
- Global As View (GAV)
- Global-Local As View (GLAV)
- OBI-WAN is the first mediator system capable of integrating multiple data sources of heterogeneous data models through **GLAV** mappings.

Answering queries over Local As View

Query:

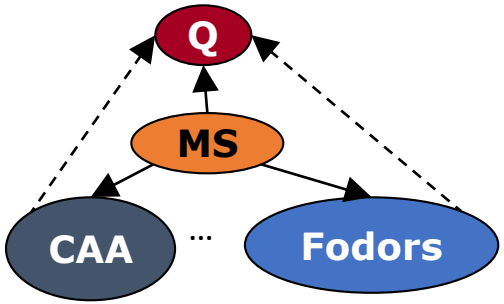
Dest(code) :- Airport(code, city), Feature(city, “Beach”)

Sources/Views:

CAA-Air(code, city) :- Airport(code, city)

Fodors(city, POI) :- Feature(city, POI)

Sun-Surf(city) :- Feature(city, “Beach”)



Rewriting:

Dest(code):-CAA-Air(code, city), Fodors(city, “Beach”) ∪
Dest(code):-CAA-Air(code, city), Sun-Surf(city)

Maximally Contained Rewriting: all answers to Query are a subset of those of Rewriting, and Rewriting contains all possible answers given local sources

Answering queries over Global As View is easy

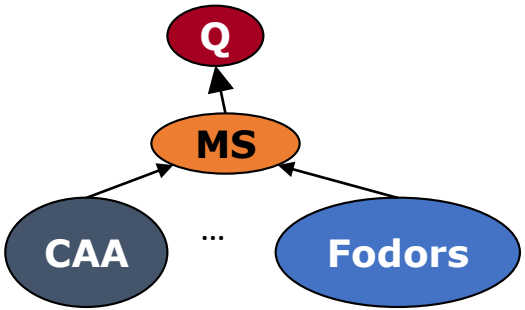
Query:

Dest(code) :- Airport(code, city), Feature(city, “Beach”)

Sources/Views:

Airport(code, city):- CAA-Air(code, city)

Feature(city, POI):- Fodors(city, POI)



Rewriting in Global as View: just expand the query. No need for complicated rewritings.

Dest(code):-CAA-Air(code, city), Fodors(city, “Beach”)

Third Architecture: Global Local As View

Mediated Schema:

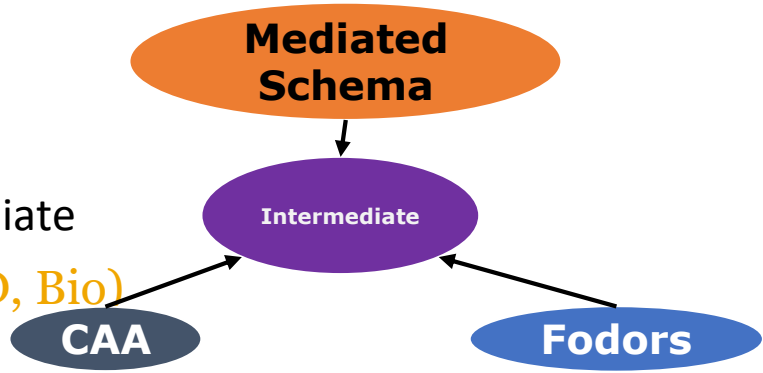
Guide(Name, ID, Bio)

Global View – View(Mediated Schema) -> Intermediate

V-Fodors-Guide(Name, Bio) :- Guide(Name, ID, Bio)

Local View – View(Intermediate) -> Fodors

V-Fodors-Guide(Name, Bio) :- Fodors-Guide(Name,Bio)



Discussion Q#3

LAV, GAV, GLAV

What are the advantages/disadvantages of each type of mapping? Why did they us GLAV for Obi-Wan?

Number off, 1,2,3,4

Local as view (LAV): global view created from local views

Global as view (GAV): global view defined first

Global local as view (GLAV): intermediate views used to build up local views to global views

Advantages vs. Disadvantages

Summary from discussion Q#3 (previous slide)

Property	LAV	GAV	GLAV
Query Types	Simple	Complex	Complex
Implementation	Easy	Hard	Medium
Query Optimization	Easy	Hard	Easy
Schema Change	Hard	Easy	Medium

RDF Integration System (RIS)

- Integrating data from heterogeneous sources (data model + query language) into an RDF graph
- The graph consists of RDFS ontology, and of data derived from the sources by means of GLAV mappings.

How to answer RIS queries?

- They have three different options
  - No reasoning at query time – don’t create new rules, just create new tuples
  - All reasoning at query time – create all the new rules AND the new tuples
  - Some reasoning at query time – create the new tuples, and some of the rules

No Reasoning at Query Time

- While querying over the RIS, nothing is inferred regarding the underlying data
- When simple queries are required
- Computationally efficient

All Reasoning at Query Time

- While querying over the RIS, all rules are inferred from the underlying data
- Works with complex queries
- Computationally taxing

Some Reasoning at Query Time

- While querying over the RIS, some rules are inferred from the underlying data
- Works with complex queries
- Computationally taxing

Discussion Q#4

Data?

All reasoning at query time

G1

Complex queries

Data?

Some reasoning at query time

G2

Moderately complex queries

Data?

No reasoning at query time

G3

Simple queries

What forms of data sources are best for each answering strategy?