Ontologies and Databases

What is an Ontology?

A **model** of (some aspect of) the world

- Introduces **vocabulary** relevant to domain
  - Often includes names for classes and relationships
- Specifies intended **meaning** of vocabulary
  - Typically formalised using a suitable logic
  - E.g., OWL formalised using ***SHOIQ*** description logic
- Consists of two parts
  - Set of **axioms** describing **structure** of the model
  - Set of **facts** describing some particular concrete situation

Axioms

Describe the **structure of the model**, e.g.:

Class: HogwartsStudent
EquivalentTo: Student and attendsSchool value Hogwarts

Class: HogwartsStudent
SubClassOf: hasPet only (Owl or Cat or Toad)

ObjectProperty: hasPet
Inverses: isPetOf

Class: Phoenix
SubClassOf: isPetOf only Wizard

Facts

Describe some **particular concrete situation**, e.g.:

Individual: Hedwig
    Types: Owl

Individual: HarryPotter
    Types: HowgwartsStudent
    Facts: hasPet Hedwig

Individual: Fawkes
    Types: Phoenix
    Facts: isPetOf Dumbledore

Obvious Database Analogy

- Ontology axioms analogous to DB **schema**
  - Schema describes structure of and constraints on data
- Ontology facts analogous to DB **data**
  - Instantiates schema
  - Consistent with schema constraints
- But there are also important differences…

Database -v- Ontology

Database:

- Closed world assumption (**CWA**)
  - Missing information treated as false
- Unique name assumption (**UNA**)
  - Each individual has a single, unique name
- Schema behaves as **constraints** on structure of data
  - Define legal database states

Ontology:

- Open world assumption (**OWA**)
  - Missing information treated as unknown
- **No UNA**
  - Individuals may have more than one name
- Ontology axioms behave like **implications** (inference rules)
  - Entail implicit information

Database -v- Ontology

- E.g., given facts/data:

  Individual: HarryPotter
  Facts: hasFriend RonWeasley
  hasFriend HermioneGranger
  hasPet Hedwig

  Individual: Draco Malfoy

- Query: Is Draco Malfoy a friend of HarryPotter?
  - DB: No
  - Ontology: Don't Know
    - OWA (didn't say Draco was not Harry's friend)

Database -v- Ontology

- E.g., given facts/data:

  Individual: HarryPotter
      Facts: hasFriend RonWeasley
              hasFriend HermioneGranger
              hasPet Hedwig

  Individual: Draco Malfoy

- Query: How many friends does Harry Potter have?
  - DB: 2
  - Ontology: at least 1
    - No UNA (Ron and Hermione may be 2 names for same person)

Database -v- Ontology

- E.g., given facts/data:

  Individual: HarryPotter
  Facts: hasFriend RonWeasley
  hasFriend HermioneGranger
  hasPet Hedwig

  Individual: Draco Malfoy

  ➔ **DifferentIndividuals: RonWeasley HermioneGranger**

- Query: How many friends does Harry Potter have?
  - DB: 2
  - Ontology: at least 2
    - OWA (Harry may have more friends we didn't mention yet)

Database -v- Ontology

- E.g., given facts/data:

  Individual: HarryPotter
  Facts: hasFriend RonWeasley
  hasFriend HermioneGranger
  hasPet Hedwig
  ➔ **Types: hasFriend only RonWeasley or HermioneGranger**

  Individual: Draco Malfoy

  DifferentIndividuals: RonWeasley HermioneGranger

- Query: How many friends does Harry Potter have?
  - DB: 2
  - Ontology: 2!

Database -v- Ontology

- Insert new facts/data:

  Individual: Dumbledore

  Individual: Fawkes
  Types: Phoenix
  Facts: isPetOf Dumbledore

- Response from DBMS?
  - Update rejected: **constraint violation**
    - Range of hasPet is Human; Dumbledore is not Human (CWA)
- Response from Ontology reasoner?
  - **Infer** that Dumbledore is Human (range restriction)
  - Also infer that Dumbledore is a Wizard (only a Wizard can have a pheonix as a pet)

DB Query Answering

- Schema plays **no role**
  - Data must explicitly satisfy schema constraints
- Query answering amounts to **model checking**
  - I.e., a “look-up” against the data
- Can be very **efficiently implemented**
  - Worst case complexity is low (logspace) w.r.t. size of data

Ontology Query Answering

- Ontology axioms play a powerful and **crucial role**
  - Answer may include implicitly derived facts
  - Can answer conceptual as well as extensional queries
    - E.g., Can a Muggle have a Phoenix for a pet?
- Query answering amounts to **theorem proving**
  - I.e., logical entailment
- May have very **high worst case complexity**
  - E.g., for OWL, NP-hard w.r.t. size of data (upper bound is an open problem)
  - Implementations may still behave well in typical cases

Ontology Based Information Systems

- Analogous to **relational database management systems**
  - – Ontology ¼ schema; instances ¼ data
- Some important (**dis**)**advantages**
  - + (Relatively) easy to maintain and update schema
    - Schema plus data are integrated in a logical theory
  - + Query answers reflect both schema and data
  - + Can deal with incomplete information
  - + Able to answer both intensional and extensional queries
  - – Semantics may be counter-intuitive or even inappropriate
    - Open -v- closed world; axioms -v- constraints
  - – Query answering (logical entailment) much more difficult
    - Can lead to scalability problems

Ontology Based Information Systems

- Similar to **relat**
  - Ontology ¼ sc
- Some importar
  + (Relatively) ea chema
    - Both schem
  + Query answer ta
  + Able to answe sional queries
  - Semantics ma n inappropriate
    - Open -v- cl ts
  - Query answer more difficult
    - Can lead to scalability problems

Very powerful, but not miraculous!

Best of Both Worlds?

- W3C OWL working group is developing **OWL 2**
  - OWL 2 is an update to OWL adding many useful features
    - Increased expressive power, e.g., w.r.t. properties
    - Extended support for datatypes and values
    - Database style keys
    - Rich annotations
- OWL 2 also defines several **profiles**
  - Profile is a language subset with
    - Useful computational properties
    - Useful implementation possibilities

Best of Both Worlds?

EL++ profile

- Maximal language for which reasoning (including query answering) known to be worst-case **polynomial**
- Captures expressive power used by many **large-scale ontologies**
  - Features include existential restrictions, intersection, subClass, equivalentClass, class disjointness, range and domain, transitive properties, …
  - Missing features include value restrictions, Cardinality restrictions (min, max and exact), disjunction and negation

Best of Both Worlds?

DL-Lite profile (not to be confused with OWL Lite!)

- Maximal language for which reasoning (including query answering) is known to be worst case **logspace** (same as DB)
- Captures (most of) expressive power of **ER/UML** schemas
  - Features include limited form of existential restrictions, subClass, equivalentClass, disjointness, range and domain, symmetric properties, …
- Query answering can be implemented using **query rewriting**
  - Resulting SQL query/queries capture all information from axioms
  - Can use query/queries with standard DBMS and relational data

Best of Both Worlds?

OWL-R profile

– Allows for scalable (**polynomial**) reasoning using rule-based technologies
– Includes support for **most OWL features**
    • But standard semantics only apply when they are used in a restricted way
    • Related to DLP and pD*
– Can be implemented on top of **rule extended DBMS**
    • E.g., Oracle's OWL Prime implemented using forward chaining rules in Oracle 11g

Summary

- Ontologies consist of sets of axioms and facts
- Analogous to DB: axioms ¼ schema; facts ¼ data
- Important differences in semantics
  - DB: UNA, CWA and constraints
  - Ontology: OWA and implications
- Ontologies are very powerful, but there are costs
  - Can be scalability problems
- OWL 2 provides choice of several profiles
  - Tractable reasoning (logspace or polynomial)
  - Different features and implementation pathways