Ontologies

R K Joshi
IIT Bombay
Recent Advances in Programming Technologies

The Paradigms

Object orientation

decomposition of domain into interconnected objects

Component orientation

interface separation and instance plugability

Service orientation

descriptions, find and bind

LAN services
Web services

Cell phone/device based service orientation

Event orientation

asynchronous, push and pull styles

Data storage and computing techniques

warehouses, replication, parallel computing, clouds, grids

Interfaces and usability engineering

browser based, desktop based, sms based, email based etc.
The Span of Applications

Initial applications

desktop based

  local database, local transactions, accounting, billing etc

network based

  centralized database, transactions, networked access

web based, enterprise driven

  large scale information and knowledge sharing

    plain content

    semantic content
Range of Applications

Academia
Health
Governance
Banking
Finance
Libraries
Law
Transportation
Logistics
Hotels and Tourism
Travel
Social networking
Event management
Industrial processes
Enterprise functioning
Scientific computing
Software Engineering
Linguistics and semantics
What's common among them

They have commonly occurring entities in their respective domains

The entities are related to each other in some ways

Entities have attributes

There are processes involving state changes

Information finally gets represented and then manipulated through software applications

One should be able to make sense out of information, develop applications whenever needed, and share information among various stakeholders.
Consider the following problem

Is Prof. RKJ available for meeting on Thursday at 9 AM on Nov 15th?

How would your program solve this problem automatically?

Where does Prof. RKJ work?
Is Nov. 15 holiday for the institute?
What are the published slots by the institute?
Which are the slots scheduled around 9 am on that day?
Is Prof. RKJ engaged in any one of those slots?

Manually, you can do it by browsing through the pages
information may be available
but it may not be in a directly programable format.
Consider another problem

Is a room close to PSG available in Coimbatore in range Rs. 1500-5000 per day on Nov 7th?

Which are the hotels in Coimbatore

Their rates

Room availability on a particular day

The location of PSG

The location of the hotels

What does it mean by the term 'close to'? How close?
Another problem

Is there a hollywood movie containing a song by the indian artist Kishore Kumar?

- list of world movies from among them which are hollywood movies?
- list of tracks appearing them
- the singers of the tracks
Some more problems

- What are the side effects of Amoxicillin and are there treatments for those?
  - doctors may want to obtain this information
- Is there an outbreak of Dengue in a particular area?
- What percentage of population in Coimbatore represents uneducated girls?
- What is the procedure to apply for a visa?
- What structure does Raga Kalyani follow?

What is the composition of a given molecule?
The problem of Knowledge and Information Representation

- Software engineers develop applications
- Domain experts use them
- Domain experts handle information
- Software engineers provide ways to create, store, find, manipulate, backup and version information.
- Domain experts know the rules of their domain
- Software engineers create 'soft knowledge' that represents these rules.

The knowledge is then applied to the information
The branch of Ontology

Philosophical Foundations
e.g. Things, Properties, States, Events, Processes, Kinds

Mathematical (Logical) Foundations
e.g. Set/Subset, Type/Subtype, Rules

Domain Ontologies
e.g. Order, delivery, delivery schedule, payment, delivery options, stocks

Information structured following the ontology
e.g. An XML based document, or tagged data, bibtex file
Benefits from ontologies

- Domain Conceptualization in terms of precise semantic Information Systems Representation creating the actual instances and relationships
- Software Applications Building around ontological representation
- Information Sharing and Exchange in an interoperable way
- Knowledge Sharing and Exchange for high level (meta) abstraction sharing – 'share once and use again over instances'
- New Applications benefiting from Knowledge – since knowledge is shared, one is not stuck with proprietary formats. Benefits of interoperability and knowledge sharing makes the system open to new applications
Meaning Triangle

- Thought of Reference
  - Refers To: Referent
  - Symbolizes: Symbol

Referent → Stands for → Symbol

Symbol stands for "House"
Bungean Ontology

- Bunge's Ontology (1977) is considered to be a general systems theory
- Wand and Weber adapted it to Software Systems later (1993, 1995)
- The postulates in Bungean ontology are widely accepted statements about real world phenomena.
- They are every day experiences, facts and observations.
Intrinsic Ontological Categories

Most significant and fundamental categories

- **Thing**
  - They have identity, existence in reality, substantial, they possess properties e.g. Book

- **Property**
  - They characterize objects in reality, objects have properties, they do not have independent existence apart from the objects to which they are associated e.g. White

- **State**
  - Everything is in some or the other state at a given time, captures changing nature of things, property, e.g. available mapped to a value
Representational Categories

- They provide a representation of an intrinsic category

- Schema
  - Representation of a Thing
    - e.g. Book {class no., title, authors}

- Attribute
  - Representation of a Property
    - e.g. Address {house no., street no, pincode}

- State Variable
  - Representation of a State
    - e.g. Boolean book_status
<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Thing</th>
<th>Properties</th>
<th>States</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Book as a Library Item</td>
<td>Title, Author, Price, ISBN, Publisher, Classification No.</td>
<td>onTheRack, issued, claimed, written-off, missing,</td>
<td>Book(Class. No, Title, Author), Book(ISBN, Title, Publisher)</td>
</tr>
<tr>
<td>2.</td>
<td>University Student</td>
<td>Reg. No. Names, Address, Date of Birth, Course Registered, Degree Awarded,</td>
<td>registered, graduated, migrated</td>
<td>Student(Reg.No, Name, Address) Student(Reg. No., Course Registered, Degree awarded)</td>
</tr>
<tr>
<td>3.</td>
<td>Cricket Player</td>
<td>Name, Runs Scored, Centuries Scored, Wickets Taken, 5-wicket Hauls,</td>
<td>playing, injured, rested, retired</td>
<td>Player(Name, Runs Scored, Centuries scored) Player(Name, Wickets Taken, 5-wicket haul)</td>
</tr>
<tr>
<td>4.</td>
<td>Network Printer</td>
<td>Name, Make, Location</td>
<td>on, off, busy, idle</td>
<td>Printer(Name, Make, Location)</td>
</tr>
</tbody>
</table>
Properties

• Intrinsic properties
  • Associated with a single object e.g. Age and height of a person – the same person

• Mutual property
  • Relational properties between to objects
  • e.g. Worksfor between a company and an employee
    – Binding – sales relationship
    – Non binding – younger than relation
Primitive Relational Categories

- Relate two intrinsic categories
- Possesses: between Thing and Property
  - Book as a thing possesses properties author, title
- Precedes: between Property and Property
  - Being a person precedes being a student
- Event: between two states of a thing, change from state s1 to state s2
Composition Categories

- Complex category from simple categories
- Conjunction
  - Complex properties from simple properties
- Association
  - Complex things from simple things
- Event composition (process) is a complex event
- Part of – a complex thing
Collection Categories

- **Class**
  - Groups similar things together
  - They have a characteristic property
  - CSE students are students enrolled in cse

- **Kind**
  - Organizes a group of things by a set of properties
  - Child labor – group of children who have works for and who are underaged

- **Other categories such as History and State Space**
A W Kiwelekar and R K Joshi, Object oriented metamodel for BWW ontology
Application of Ontological Approach to Architecture Extraction

2013 IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)
<table>
<thead>
<tr>
<th>Fully Qualified Name of a Class</th>
<th>Member Analysis</th>
<th>Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>A class has attributes declaration.</td>
<td>hasAttributes</td>
<td>Y/N</td>
</tr>
<tr>
<td>A class has methods declaration.</td>
<td>hasMethods</td>
<td>Y/N</td>
</tr>
<tr>
<td>All the attributes and methods defined in a class are <em>static</em>.</td>
<td>hasAllMembersStatic</td>
<td>Y/N</td>
</tr>
<tr>
<td>A class has the method <code>main</code> defined in it.</td>
<td>hasMain()</td>
<td>Y/N</td>
</tr>
<tr>
<td>A class has the method <code>run</code> defined in it.</td>
<td>hasRun()</td>
<td>Y/N</td>
</tr>
<tr>
<td>A class has at least one method containing a looping construct.</td>
<td>hasMethodContainingLoop</td>
<td>Y/N</td>
</tr>
<tr>
<td>A class implements a non-null constructor method.</td>
<td>hasConstructor</td>
<td>Y/N</td>
</tr>
<tr>
<td>A class invokes methods from another class.</td>
<td>hasInteractions</td>
<td>Y/N</td>
</tr>
</tbody>
</table>
Rules for Ontological Categorization

\[
\neg \text{hasMain}() \land \neg \text{hasRun}() \land \text{hasAttributes} \land \\
\neg \text{hasMethods} \Rightarrow \text{Property} \tag{Rule 1}
\]

\[
\neg \text{hasMain}() \land \neg \text{hasRun}() \land \text{hasAttributes} \land \\
\text{hasMethods} \land \neg \text{hasAllMembersStatic} \land \\
\text{hasConstructor} \land \text{hasInteractions} \Rightarrow \\
\text{Thing} \tag{Rule 4}
\]

Rules for classifying into Event, Thing, Property, Process
Ontology Abstractions

Upper Ontology

vs

Lower Ontology
Domain Ontology
Measurement Ontology

- Measurement
  - Drawal
  - Voltage
  - TieLineFlow
  - CorridorFlow
  - Frequency
    - Import
    - Export
    - UI-Flow
Location Ontology

- Location
  - Geographical Location
  - Electrical Location
    - Constituent
      - Load
      - Generation
    - Bus
      - located in
    - Tieline
      - connects
      - collection
    - Corridor
      - connects
Overdrawal Event Representation:

XML RDF

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmlns:eo="http://www.iitb.ac.in/eventontology#"
    rdf:RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <eo:Overdrawal rdf:ID="od_1">
    <eo:timestamp>2011-03-08 19:05:00</eo:timestamp>
    <eo:frequency>49.79 Hz</eo:frequency>
    <eo:scheduled>3657 MW</eo:scheduled>
    <eo:actual>4352 MW</eo:actual>
    <eo:constituent.ID rdf:resource="#const_89" />
  </eo:Overdrawal>

</xmlns:eo=http://www.iitb.ac.in/eventontology# rdf:RDF>
```

Undervoltage Event Representation : XML RDF

<eo:UnderVoltage rdf:ID="uv_1">
  <eo:timestamp>2011-03-08 20:22:00</eo:timestamp>
  <eo:frequency>49.64 Hz</eo:frequency>
  <eo:rated>400 KV</eo:rated>
  <eo:actual>377 KV</eo:actual>
  <eo:Bus.ID rdf:resource="#bus_31" />
</eo:UnderVoltage>
Markup Representation Techniques

HTML
  web content tags, generic

XML
  richer tags, human understandable

Further standardization for more semantic content
  RDF, OWL
RDF (Resource Description Framework)

- W3C standard
- Metadata description
- Resource
- Predicates about resources