ONTOMETRY ENGINEERING
AND TOOL

Nopphadol Chalortham, PhD
Faculty of Pharmacy, Chiang Mai University
nopphadolc@gmail.com
Ontology

- A specification of conceptualization.
- A description (like a formal specification of a program) of the concepts and the relationships that can exist for an agent or a community of agents.
Ontology component

- Concept
- Property
- Relationship
- Constraint
- Axiom
Concept

• Concepts (among other things) are in general language-independent
  • Mental or logical representation reality
  • Related to other concept
  • Do not need symbols but hold them for means of communication

• A concept has
  • Intension or meaning
  • Extension i.e. the set of objects that the concept refer to
Relation

- Object relation
  - IS-A relation
  - PART-OF relation
  - Specific relation
    - Equal
    - Sameas
    - Larger-than
    - Not-equal
    - different
  - Defined relation
- Data relation (Attribute-of relation)
  - Boolean
  - String
  - Integer
  - Float
  - Number
Knowledge base

Taxonomy
- Vocabulary
- Structure

Ontology
- Taxonomy
- Relationships
- Constraints
- Axiom

Knowledge
- Ontology
- Instance
Markup languages

- HTML
- XML
- RDF, RDFs
- DAML-ONT, OIL
- OWL
  - OWL lite
  - OWL DL
  - OWL full
Ontology design approaches

- Inspiration approach
  - Based on individual creativities and personal views

- Induction approach
  - Created by observation and analysis of a particular case in that domain

- Deduction technique
  - Applies general principles and adapts them according to a specific case

- Synthesis approach
  - Identifies a set of ontologies and then synthesizes them with other related concepts

- Collaborative approach
  - Concern with a joint effort and using the group members experience and opinions to build the ontology
Ontology development methodology

- Enterprise modeling, *Uschold and King*
- Methontology for the domain of chemicals, *Fernandez-Lopez et al*
- The generic guidelines, *Ontology 101*
- The knowledge meta process, *Staab et al*
- The OntoClean methodology for validation of “the adequacy and logical consistency of taxonomix relationships”
- The methodology for crating business ontology supporting semantics interoperability.
Ontology development

1. Determine scope
2. Define classes
3. Enumerate terms
4. Define relations
5. Consider reuse
6. Create instances

Individual (instance)   Class (concept)   line = part-of relation   line = is-a relation

tablet production

SQC
EQC
manufacturing problem
method
compound
compound
compound

process
direct
compression
dry granulation

granulation
wet granulation

unit operation
mixing
drying
compression

drug X tablet production

SQC1
EQC1
manufacturing problem1
wet
granulation1
drug X
lactose
croscarmellose sodium
PVP

dry mixing1
wet mixing1
communition1
drying1
compression1

SQC1
EQC1
manufacturing problem1
drug X lactose
croscarmellose sodium
PVP

dry mixing1
wet mixing1
communition1
drying1
compression1

line = part-of relation   line = is-a relation
FUNDAMENTAL RULES

• There is no one correct way to model a domain—there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.

• Ontology development is necessarily an iterative process.

• Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain.
<table>
<thead>
<tr>
<th>Hozo</th>
<th>Protege</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is-A relation</td>
<td>Class Tab</td>
</tr>
<tr>
<td>Part-of relation</td>
<td>Object Properties Tab</td>
</tr>
<tr>
<td>Attribute-of relation</td>
<td>Data Properties Tab</td>
</tr>
<tr>
<td>Instance editor</td>
<td>Individual Tab</td>
</tr>
<tr>
<td>Separated file</td>
<td>Single file</td>
</tr>
<tr>
<td>Good graphic user</td>
<td>Many plug-in for protégé</td>
</tr>
<tr>
<td>interface</td>
<td>• Reasoner, JESS, SPARQL, SWRL</td>
</tr>
</tbody>
</table>

**Protege**

- Class Tab
- Object Properties Tab
- Data Properties Tab
- Individual Tab
- Single file
- Many plug-in for protégé
  - Reasoner, JESS, SPARQL, SWRL
Ontology evaluations

- The ‘Gold Standard’ evaluation
- Data driven evaluation
- Evaluation by humans
- Application-based evaluation
Ontologies and Database: What’s difference?

• What is it for?
• What does it look like?
• How do you build one?
• How is it implemented and used?
• Where are the semantics?

Michael F. Uschold
Senior Ontology Consultant
Semantic Arts - See more at:
Relation

- Object relation
  - IS-A relation
  - PART-OF relation
  - Specific relation
    - Equal
    - Sameas
    - Larger-than
    - Not-equal
    - different
  - Defined relation

- Data relation (Attribute-of relation)
  - Boolean
  - String
  - Integer
  - Float
  - Number
An *Is-a* relation between A and B represents the relation that all the characteristics of A is inherited to B.
Part-of relation

- A *Part-of* relation between A and B represents the relation that B is one of the component of A.
Cardinality

• The value-cardinality of a binary-relation with respect to a given domain instance is the number of range-elements to which the relation maps the domain-element.
• Defines how many values a slot can have.
Attribute-of relation

• An *Attribute-of* relation between object and data type that describe characteristics of object.
Bicycle concept

- bicycle [1]
  - handlebar [1]
  - saddle [1]
  - pedals [1]
  - body [1]
  - wheel [2]

part-of relation

Class (concept)
Bicycle concept (Cont.)

- Bicycle
  - Touring bicycles
  - Utility bicycles
  - Cruiser bicycles
  - Mountain bicycles
  - BMX bicycles
  - Racing bicycles

Is-a relation

Class (concept)
Bicycle concept (Cont.)

- bicycle
  - touring bicycles
  - utility bicycles
  - cruiser bicycles
  - mountain bicycles
  - racing bicycles
  - BMX bicycles

Is-a relation

Class (concept)
Bicycle concept (Cont.)

- **Part-of relation**
  - **Class (concept)**
    - touring bicycles
    - utility bicycles
    - cruiser bicycles
    - mountain bicycles
    - racing bicycles
    - BMX bicycles

- **Is-a relation**
  - bicycle
    - handlebar
    - saddle
    - pedals
    - body
    - wheel

1. touring bicycles
2. utility bicycles
3. cruiser bicycles
4. mountain bicycles
5. racing bicycles
6. BMX bicycles
Part-of and Is-a relations (1)

Skiing \(\text{Is-a}\) Hobby

Restaurant \(p/o\) Hotel

Human \(\text{Is-a}\) Mammal

YeePeng \(\text{Is-a}\) Loy Kratong Festival
Part-of and Is-a relations (2)

Plastic \( p/o \) Glass (cup)

Midfield \( p/o \) Football Team

Linux \( p/o \) Computer

Mushroom \( p/o \) Tomyum
Vehicle (Is-a relation)

- Vehicle
  - Space Vehicle
  - Air Vehicle
  - Ground Vehicle
    - Road Vehicle
    - Rail Vehicle
  - Water Vehicle
    - On water Vehicle
    - Under water Vehicle
    - Engine water Vehicle
    - Non-engine water Vehicle
  - Snow Vehicle
Ground Vehicle (*Part-of*)

- Ground Vehicle
  - Road Vehicle
  - Rail Vehicle
Ground Vehicle 

Ground Vehicle

Road Vehicle

Rail Vehicle

wheel

wheel

wheel

trunk

trunk

p/o 1

p/o 1

p/o 1.. wheel

p/o 1.. wheel

p/o 0.. trunk

p/o 1.. wheel

(\textit{Part-of})
Address

```
part-of relation
Is-a relation
Class (concept)
```
ทศพิธราชธรรม

part-of relation

Is-a relation

Class (concept)
Hand

part-of relation

Is-a relation

Class (concept)
Color

- **Eye**
  - p/o 1 **Color**

- **Color**

- **Eye**
  - a/o 1 **String**

- **Class (concept)**: part-of relation, Is-a relation
Hozo Environment(1)

- Mizoguchi Lab., The Institute of Scientific and Industrial Research, Osaka University
  - Kouji Kozaki
  - Mamoru Ohta
  - Eiichi Sunagawa
  - Yoshinobu Kitamura
  - Mitsuru Ikeda
  - Riichiro Mizoguchi
- http://www.ei.sanken.osaka-u.ac.jp/
- http://www.hozo.jp/hozo_eng/
Hozo Environment(2)

• System requirements
  • Machine PC/AT compatible (DOS/V)
  • CPU Intel Pentium III processor, 800MHz or higher
  • Memory 256MB or more RAM (512MB or more recommended)
  • Hard Disk Space 30MB or more available hard disk space.
  • Monitor Resolution 1024x768 (XGA) or higher
  • OS* Windows 2000 or higher (Windows XP or higher recommended), Mac OS X 10.4 or higher
  • Java JRE/JDK 1.5.0_10 or higher (1.6.0 or higher recommended)
  • Network Not required if using on a local machine. When sharing within networks, clients and servers are connected via Ethernet using TCP/IP protocol.
Hozo Environment (3)

Hozo Ontology Editor

Instance Editor (Model)
File > new File …

Project > new Project …
New Ontology

New Ontology Name: bicycle

Please select a project to create new ontology.

bicycle

- LocalStorage
  - LocalStorage
  - SampleProject1
  - SampleProject2
  - SampleProject3
  - Pizza
  - bicycle

[New] [Close]
Do you add the "Any" concept as root concept?

The "Any" concept is undefined.
Do you add the "Any" concept as root concept?

Yes  No
Right click > add Node...
Right click > add Slot > part-of
THANK YOU FOR YOUR ATTENTION

nopphadolc@gmail.com