

How to Build an Ontology: Examples and Guidelines

Riichiro Mizoguchi
miz@ei.sanken.osaka-u.ac.jp
ISIR, Osaka University

What is NOT an ontology?

- ◆ **A set of vocabulary**
 - ◆ It needs *is-a* hierarchy of them at least
 - ◆ Not terms but **concepts**
- ◆ **A taxonomy**
 - ◆ It is more than a simple classification of concepts
 - ◆ It should consist of concepts in terms of which you can **explain** the target object(world)

An example Ontology of vehicle

A simple taxonomy

Vehicle

- ground vehicle
 - motor car
 - 4 or more wheel car
 - car
 - truck
 - motor bike
 - train
 - ship
 - air craft

An ontology of vehicle

• Vehicle world

- type
 - ground vehicle
 - ship
 - air craft
- function
 - to carry persons
 - to carry freights
- attribute
 - power
 - size
- component
 - engine
 - body
- traffic system
- ...

Key topics in my talk

- Guidelines
- Distinction between **role** and basic concept
- Correct use of **is-a** and **part-of** relations
- Avoiding **multiple inheritance**
- Attribute vs. Property
- Upper ontology
- Ontology of **representation**, etc.

Three-layer model

Top-layer: the whole building process compliant with the conventional **software development process**

Middle-layer: Generic constraints and guidelines which specify major steps

Bottom-layer: The most fine-grain guidelines such as those for **class identification process, etc.**

Fig. 2 Three-layer model of ontology building methodology.

Guidelines at the Middle-layer

- Concepts rather than terms**
Distinction between a term and a concept is critical, since people will be easily trapped by the endless terminological discussion departing from the underlying conceptual structure of the target domain. Ontology is totally independent of terminological problems.
- Mixed and flexible strategies** of Top-down, Bottom-up and Middle-out is strongly suggested. Never stick to only one of the strategies.
- Top-level category should be identified in the early phase of the development process to govern the rest of the steps.
- When you deal with a concept, identify its main components; that is, "**part-of**" relation as well as its main attributes. You can thus find and extend candidates of concepts to be included in the ontology.
- Axiom writing should be done after finishing **is-a** hierarchy building and informal term definition.
- Note that you cannot define any concept completely in theory.** Therefore, do not stick to the definition of each term too much. At best, you only can give necessary conditions of them. Term definition in the early phase can be rough. Detailed definition of a term should be done after you grasp the whole structure of the ontology, that is, after building **is-a** hierarchy.
- Never try to seriously define a term one by one. Definition of a concept needs sufficient contextual information which is usually not available in the early phase. Terms are related to each other and could have several meanings which should be clarified by the context given.
- Arrange and resolve the **terminological issues(how to name a concept) at the last step.**
- When you find the necessity to define more than one meaning for one term, then you are facing the terminological problem. **Each term should correspond to exactly one meaning in an ontology**, since you are not building a dictionary, but a well-organized conceptual structure. Each term is only a label of the concept. You of course can build a dictionary after building an ontology.
- Put a higher priority on is-a hierarchy construction than term definition.** A carefully designed **is-a** hierarchy gives you a correct context to define a term.
- When you get stuck with a term definition, follow either one of the following:
 - Multiple meanings? Then concentrate on meaning one by one.
 - Multiple Viewpoints? Make the viewpoint explicit and then try it again
 - Check if you are discussing terminology.
 - Use **is-a** hierarchy to give enough context.

Guidelines at the Bottom layer

- (1) **Determine an essential property** for each concept and instance. It could be informal. At least, when you come into a trouble, having an essential property of each concept help you a lot.
- (2) Each subclass of a super class is **distinguished by the values of exactly one attribute** of the super class.
- (3) Properly used *is-a* relation should inherit the "Essential" property of its super classes.
- (4) **Clear and consistent differentiation between basic concepts (man, rice, oil, etc.) and role concepts (teacher, food, fuel, etc.).**
- (5) **Avoid the use of multiple inheritance** relation as much as possible. When you want to use multiple inheritance relation, then it is usually the time to make misconceptualization of the world. Suspect if you are dealing with a Role concept, or either one of them is a part of the concept.
- (6) **Do not worry about the vagueness of the boundary between similar concepts.** Most boundaries between concepts are vague. A good example is color. While there cannot be a clear boundary between ANY colors, we all share very clear understanding about color, red, blue, etc.
- (7) Pay a lot of attention on the context issue, I mean, try to build a context-independent ontology by making the context explicit. One of the typical contexts is the task you are involved in.
- (8) **Class partition is not a part-of relation.** Ex. Male is not *part-of* human.
- (9) Deal with "Representation" issue carefully.
A copy of the book of "Hamlet" is not an instance of Hamlet (what Shakespeare wrote). It is an instance of book. There can be infinite number of copies of a book with the same content. Needless to say, what Shakespeare wrote is only one and is independent of if it is written on sheet of papers or on anything else.
- (10) If multiple meanings are identified in a term, then create a new term, say, Term1 and Term2 temporarily to make each term has only one meaning, since we are dealing with a concept rather than a term.
- (11) When you notice you do not have an appropriate term to represent a concept you identify, **do not hesitate to coin a new term.** The new term could be temporary which will be fixed at the last stage.
- (12) Consult a reliable upper ontology when you find the necessity of a general and high level distinction of categories.

Fundamental definitions

- Definition 1: **Intrinsic property**
The intrinsic property of a thing is a property which is essential to the thing and it loses its identity when the property changes.
- Definition 2: **The ontological definition of a class**
 $X = \{x \mid p(x)\}$ can be a class iff p is the intrinsic property of x .
And, then and only then, $\langle x \text{ instance-of } X \rangle$ holds.
- Definition 3: **is-a relation**
is-a relation holds only between classes. $\langle \text{class } A \text{ is-a class } B \rangle$ holds iff the instance set of A is a subset of the instance set of B .

Treatment of the role concept

the context which the concept depends

Users can switch these *three views*

- ◆ **Part View**: the role concept
- ◆ **Basic View**: the basic concept
- ◆ **Full View**: the role holder

Is-a | **Role holder**

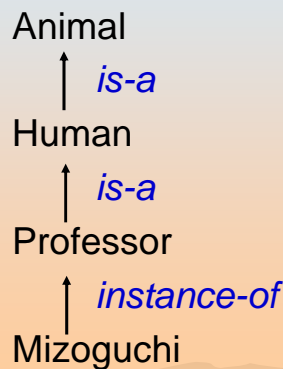
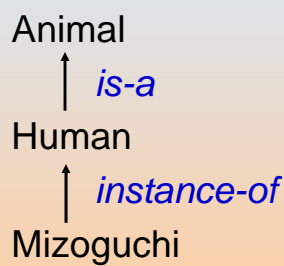
Definition Pane

Full View	Part View	Documentation	Basic View	Instance
kind	Num.	role	class	value
B1	a/o	1	name	String
B1	a/o	1	age	age
B2	a/o	1	height	number
B2	a/o	1	weight	number

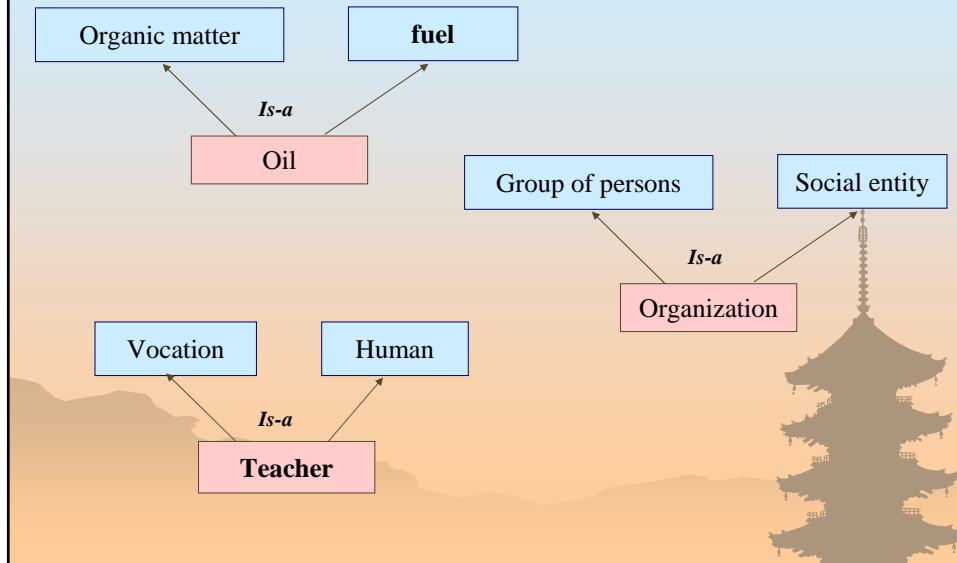
teacher role (role concept) — **teacher** (role holder) — **human** (basic concept (class constraint))

Subject — **age** — **Weight**

Do you know how to use *is-a* and *instance-of* relations?



Incorrect use of multiple inheritance



Not all “A is a B” means *is-a* link

- ◆ Tom is a man => Tom *instance-of* man
- ◆ Marry is a mother => Marry *is-a* woman who plays a mother role
- ◆ Ski is a Mizoguchi’s hobby => Ski *part-of* Mizoguchi’s hobby
- ◆ The culprit is a man => Neither *is-a* nor *part-of*
- ◆ Ocean is water => *part-of*

Typical errors found in *is-a/part-of* identification

- woman **part-of** human
- engine **is-a** car part
- teacher **is-a** human
- apple **is-a** commodity
- organization **is-a** group of persons

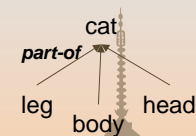
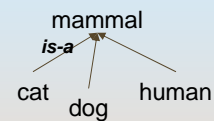
- Normal(routine) operation **part-of** operation
Restoration operation **part-of** operation
- Normal(routine) operation **is-a** operation
Restoration operation **is-a** operation



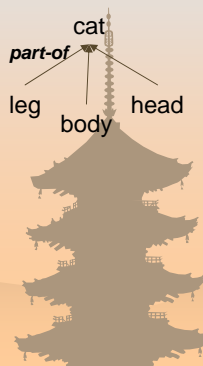
Is-a vs. part-of

- The major difference between **is-a** and **part-of**:
when making an instance of a class:

- **is-a case:**
Only one instance of one of the subclasses are generated
- **part-of case:**
Instances from each of its subclasses are generated



- cat **part-of** mammal? dog **part-of** mammal?
 - If we interpret them as **species**, it is not incorrect.



What is “property”

- In Logic
 - **Unary predicate**: *human(X)*, *red(X)*
- In OWL
 - **Link** in Semantic network (implementation-level concept)
 - **Binary predicate**: *color(X, red)*, *left(X, Y)*, *husband(X, Y)*
 - Confusion between *father(X)* and *father-of(X, Y)*
 - *Role* is not a relation (binary predicate)
- In AI
 - Object(*human*) vs. Property(*red*)
 - Property and Attribute are the same
- In Ontological engineering
 - Object(*human*) vs. Property: *red*
 - Attribute: **color**, Property: **red**
 - **Role is “father” rather than “father-of”**



Attribute representation

color *is-a* attribute
p/o"value": frequency
#red *is-a* color
p/o"value": frequency
#blue *is-a* color
frequency
p/o"value": real number
p/o"unit": #Hz
#Hz *is-a* frequency unit
flower *is-a* plant
a/o"color": #color
flower A *instance-of* flower
a/o"color": #red
post office *is-a* facilities
a/o"color": color

mail box *is-a* post facilities
a/o"color": red
mail box A *instance-of* mail box
a/o"color": n #Hz

length *is-a* attribute
p/o"value": real number
p/o"unit": #length unit
#cm *is-a* length unit
#m *is-a* length unit
human *is-a* mamal
a/o"height": length
John *instance-of* human
a/o"height": 175#cm



Ontology of time

- We employ the Newtonian time and model it as A sequence(line) which accepts the concept of "present time/now".
- We distinguish *time length, time interval and time point*.
- We consider time interval is the key concept in time ontology rather than time point which is considered as just an identifier of time point.
- We assume time is measured by an external device.
- We define "**Present time**" in the context of "tensed time interval" as the role-holder of the time point playing the "**Now role**".
- We assume axioms for interval which is composed of points.
- **Time is a measurement** rather than representation which has parts. Just like 1km= 1000m, 1 day=24 hours has to be written as an axiom rather than a class whose parts are "n" hour.
- There are **two units of time**: one is for time length and the other is for identity of time point. The same letter is used for it.
- We cannot make an instance of time independently. Like the attribute case, it has to be made associated with an instance of event and stored in the time slot of the event.

How to define "present time"

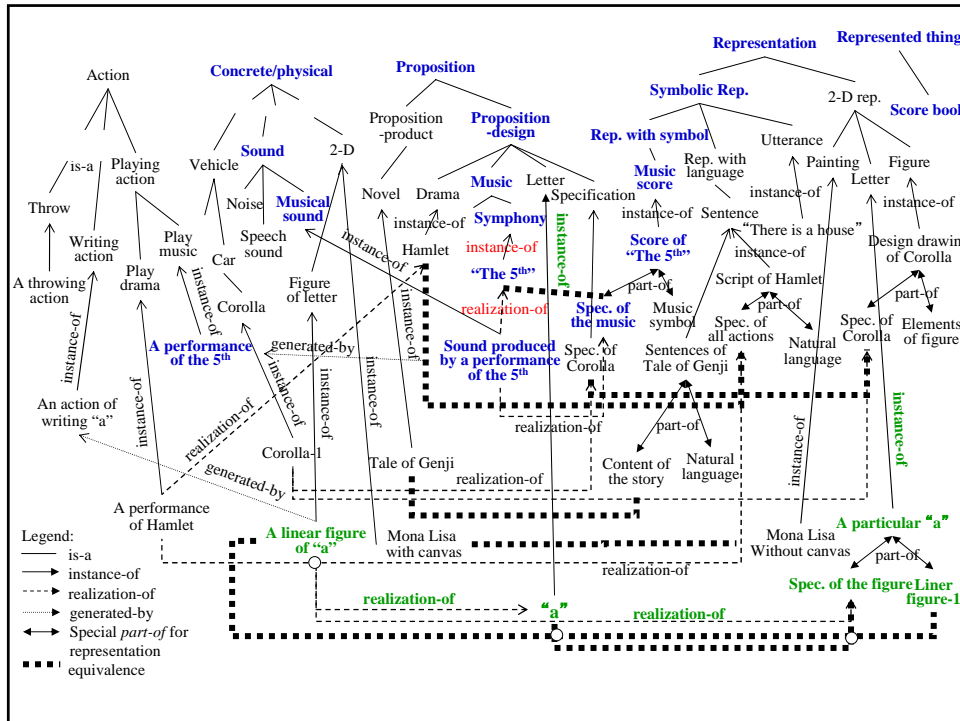
- A sequence(line)
- "Present time" is the role-holder of the time point playing the Now role. Its another role is to be a **reference point** used in the lower classes. For example, it is used as the reference point(the boundary) between past and future in the lower classes. It thus defined indirectly.

<ul style="list-style-type: none"> -substrate -entity <ul style="list-style-type: none"> -physical thing -object (Continuant) <ul style="list-style-type: none"> -topological whole -morphological thing -functional <ul style="list-style-type: none"> -biological -artificial -agent <ul style="list-style-type: none"> -single <ul style="list-style-type: none"> -animal <ul style="list-style-type: none"> -human -robot -complex <ul style="list-style-type: none"> -organization -corporation -society -process (Occurrent) <ul style="list-style-type: none"> -active <ul style="list-style-type: none"> -Action <ul style="list-style-type: none"> -physical action -achievement act (sit) -continuing act (sing) -meta-act (begin) -cognitive act -phenomena <ul style="list-style-type: none"> -burning, rain fall -stative <ul style="list-style-type: none"> -state <ul style="list-style-type: none"> -internal state (hunger) -external state (singing) -state of affairs -abstract thing <ul style="list-style-type: none"> -truth -set -value -function 	<ul style="list-style-type: none"> -number <ul style="list-style-type: none"> -real -proportional -ordinal -categorical -quality <ul style="list-style-type: none"> -intrinsic attribute <ul style="list-style-type: none"> -mass -length -color, etc. -accidental attribute <ul style="list-style-type: none"> -possessions -property <ul style="list-style-type: none"> -red -unit <ul style="list-style-type: none"> -length unit <ul style="list-style-type: none"> -cm, m, km, etc. -weight unit, etc. -quantity -role <ul style="list-style-type: none"> -intrinsic role -part-whole role <ul style="list-style-type: none"> -husband -relation-role <ul style="list-style-type: none"> -husband -friend -action role <ul style="list-style-type: none"> -agent -object -means role -vocational role <ul style="list-style-type: none"> -nurse, etc. -others <ul style="list-style-type: none"> -material <ul style="list-style-type: none"> -input, etc. -pseudo role <ul style="list-style-type: none"> -Agent <ul style="list-style-type: none"> -rotating object -flying object, etc. 	<ul style="list-style-type: none"> -relation <ul style="list-style-type: none"> -topological relation <ul style="list-style-type: none"> -structure <ul style="list-style-type: none"> -right and left -basic relation <ul style="list-style-type: none"> -is-a -part-of -temporal relation, etc. -proposition <ul style="list-style-type: none"> -belief -knowledge -event -fact <ul style="list-style-type: none"> -product proposition <ul style="list-style-type: none"> -novel -poem -etc. -design proposition <ul style="list-style-type: none"> -procedure -music -drama -specification -representation <ul style="list-style-type: none"> p/o "form": rep. form p/o "content": proposition -representation by symbol <ul style="list-style-type: none"> -musical score -sentence -representation by image -representation form <ul style="list-style-type: none"> -language <ul style="list-style-type: none"> -spoken language <ul style="list-style-type: none"> -utterance -written language <ul style="list-style-type: none"> -hand-written -still image <ul style="list-style-type: none"> -photo -hand-written image -figure, etc. -motion image
---	---	--

Ontology of Representation

- Not Semiotics, but an Engineering model of representation/content-bearing thing.
- What is an instance of Procedure?

<ul style="list-style-type: none"> -Representation <ul style="list-style-type: none"> p/o "form": Representational form p/o "content": Proposition -Represented thing <ul style="list-style-type: none"> p/o "representation": Representation p/o "medium": Representational medium -Representational medium <ul style="list-style-type: none"> -Physical thing <ul style="list-style-type: none"> -paper -canvas -Electronic thing <ul style="list-style-type: none"> -CD-ROM -etc. 	<ul style="list-style-type: none"> -Representational form <ul style="list-style-type: none"> -Symbol sequence <ul style="list-style-type: none"> -Natural language -Artificial language -Speech -Still image <ul style="list-style-type: none"> -Photo -Hand-written image -Figures, tables, etc. -Motion image -Proposition <ul style="list-style-type: none"> -Design Proposition <ul style="list-style-type: none"> -Procedure -Plan -Music -Drama -Symbol -Specification -Product Proposition <ul style="list-style-type: none"> -Novel -Poem -Painting -Calligraphy
--	---



Concluding remarks

- There is no easy way to build a good ontology
- You need to master fundamental guidelines justified by ontological theories.
- Hozo might be a usable tool.
<http://www.hozo.jp/>

