

---

# オントロジーの理論、 記述言語、利用環境

奈良先端科学技術大学院大学  
武田英明

# オントロジー構築の基礎

---

- オントロジーをいかにあるべきか
  - オントロジーの理論
    - ◆ Guarinoの形式的オントロジー
- オントロジーをいかに記述し、利用するか
  - オントロジーの記述言語
    - ◆ ARPAの知識共有活動（KIF, KQML , Ontolingua）

# Guarinoの形式的オントロジー

---

- オントロジーをいかに構成すべきかということに関して、論理的および哲学的立場から形式的に議論して、top-level オントロジーの構成原理を定式化

# 基本となる理論

---

- ものの認識とその表現の分別
  - Particulars: 実世界の実体の認識、事象、量等
  - Universals: 実体を語る際のカテゴリー、概念、属性、等
- 部分の理論：何を実体の部分とみなすか
- 全体性の理論：何を全体とみなすか
- 同一性の理論：いかなるとき実体は同一か
- 依存性の理論：実体同士の依存関係

# Particularsのオントロジー

- Substrate: 外延的に同一性基準による実体
- object : 非外延的同一性基準による実体。部分性。
- 空間的・時間的位置の有無で区別
- Concrete objectはさらにstrataによって区別される

## Particular

Substrate

Location

Space

Time

Matter

Object

Concrete object

Continuant

Occurent

Abstract object

Quality

## Strata

Static

(a situation)

Mereological

(an amount of matter)

Physical

Topological

(a piece of matter)

Morphological

(a cubic block)

Functional

(an artifact)

Biological

(a human body)

Intentional

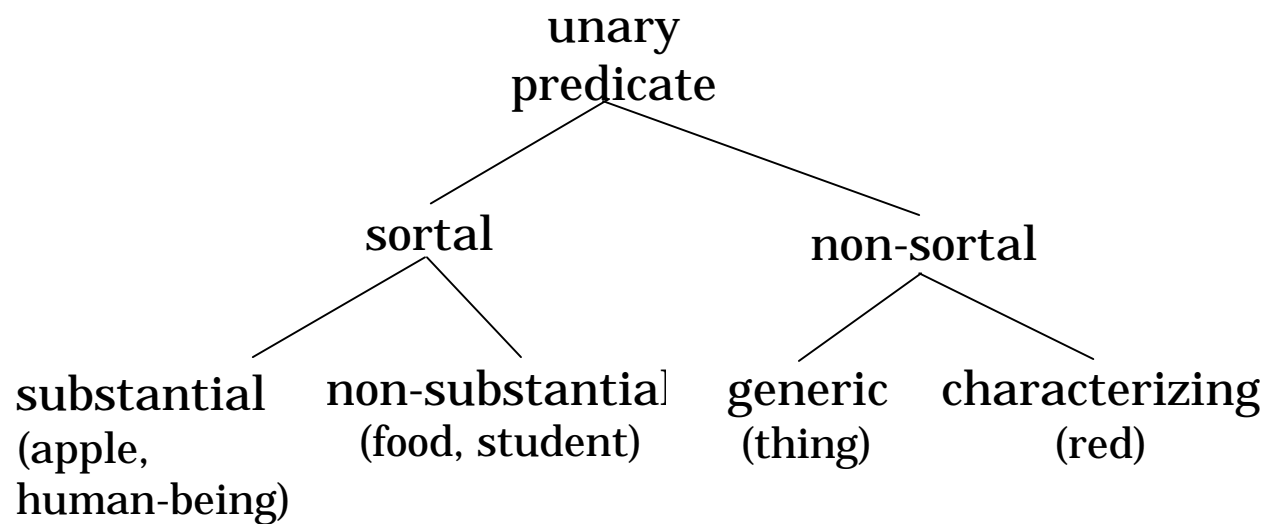
(a person or a robot)

Social

(a company)

# Universalsのオントロジー

- Sortal/non-sortal: sortalとはその個体を峻別する原理を持つもの
- substantial/non-substantial: sortalの区分
- generic predicate/characterizing predicate: の区分

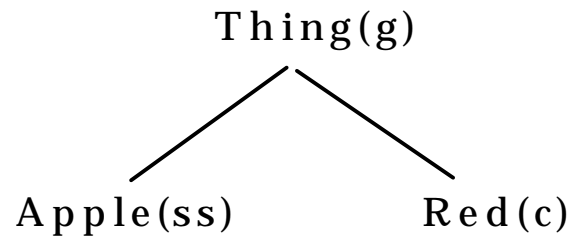


# Universalsのオントロジー

- 自然であるnatural:  $C_i = \lambda x Px$
- オントロジー的に固定されている(ontologically rigid)  
 $C_i = \lambda x (Px \supset Px)$
- 分割可能(divisive)  
 $C_i = \lambda x (Px \supset \forall y. (y < x \supset Py))$
- 一般的述語 : rigidかつdivisive
- 実質的にsortal: rigidかつ非divisive
- 非実質的にsortal: 非rigidかつnaturalな述語で、ある実質的にsortalに包含される。  $C_i = \lambda x (Px \supset Sx)$
- 特徴づけ述語 : 非rigidかつnaturalな述語で、ある実質的にsortalに包含されない

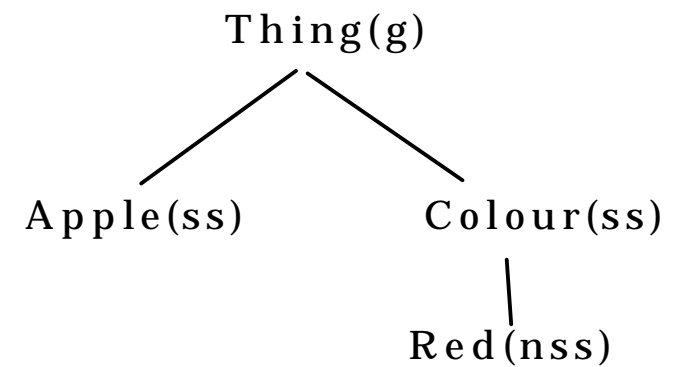
# Universalsのオントロジー

(1)



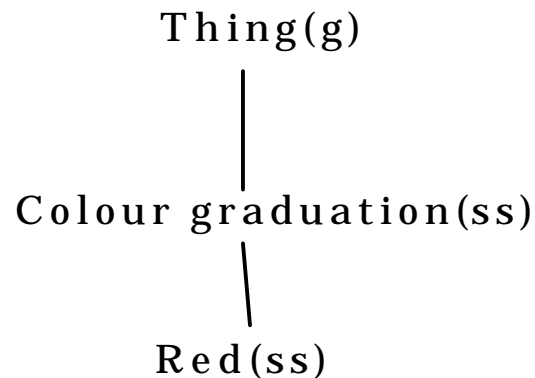
This apple is red.

(2)



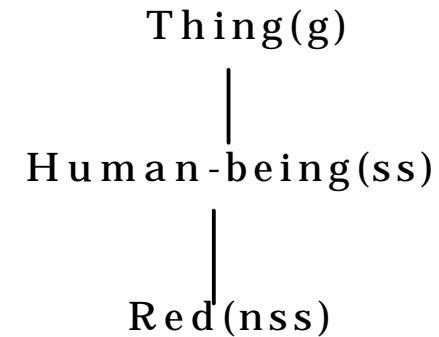
The colour of this apple is red.

(3)



Crimson is a red

(4)



John is red.

ss: substantial sortal

c: characterizing predicate

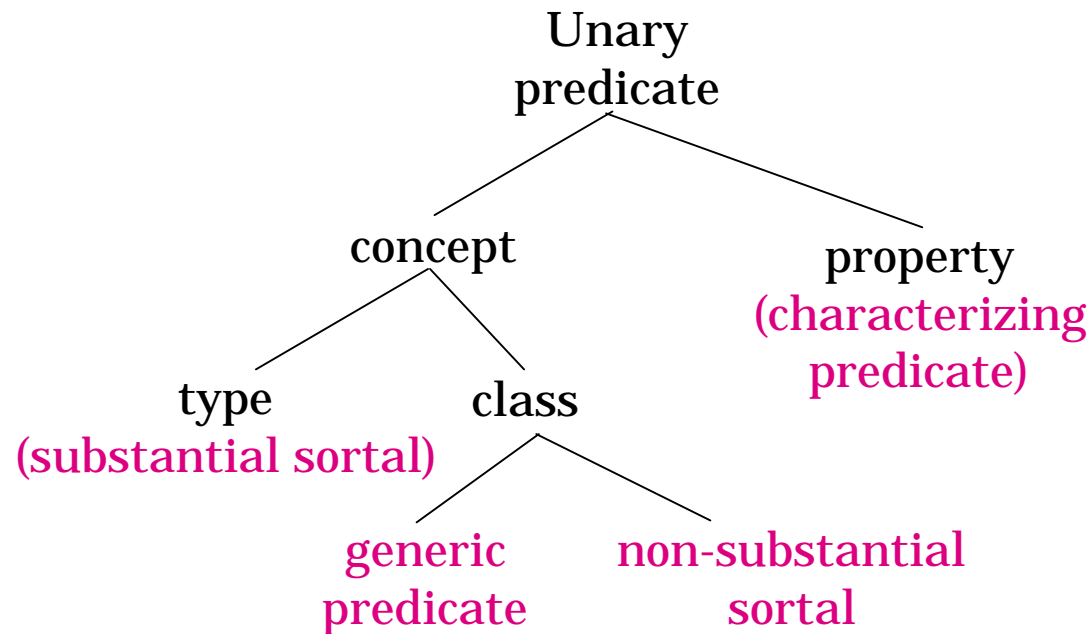
nss: non-substantial sortal

g: generic predicate



# Universalsのオントロジー

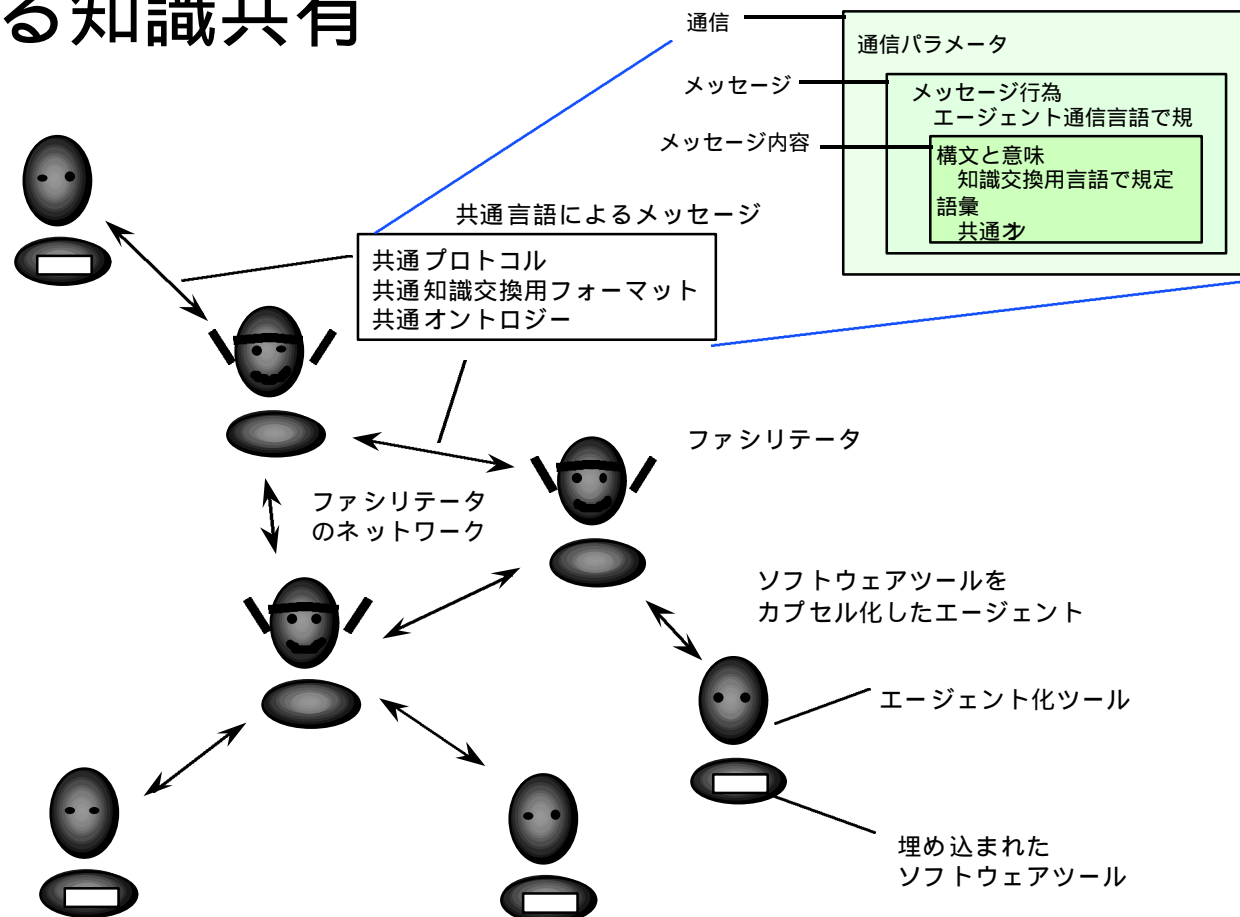
- 知識記述言語の概念のオントロジー



# 知識共有とオントロジー

## ● ARPAの知識共有の枠組み

### ■ エージェントとそのコミュニケーションによる知識共有



# エージェント通信の3つの層

---

- 通信層：
- メッセージ層
  - エージェント通信言語で記述
  - KQML(Knowledge Query and Manipulation)
- メッセージ内容層
  - メッセージ内容のシンタックスとセマンティックス
  - シンタックスは知識交換用言語 KIF(Knowledge Interchange Format)
  - セマンティックスは共通オントロジー記述用言語 Ontolinguaで

# KQML (Knowledge Query and Manipulation Language)

---

- 人間どうしの対話を分析した発話行為理論に基づく
- エージェントを仮想知識ベースとしてメッセージ交換を行う
- KQMLのメッセージの例

```
(tell :language KIF
      :ontology motors
      :in-reply-to q1
      :content (fastens frame3 motor5))
```

# Performativeの種類

---

- Basic information performatives  
tell, deny, untell,
- Database performatives  
insert, delete, delete-one, delete-all
- Basic responses  
error, sorry
- Basic query performatives  
evaluate, reply, ask-if, ask-about, ask-one, ask-all, sorry
- Multi-response query performatives
  - ◆ stream-about, stream-all, eos
- Basic effector performatives
  - ◆ achieve, unachieve
- Generator performatives
  - ◆ standby, ready, next, rest, discard, generator
- Capability-definition performatives
  - ◆ advertise
- Notification performatives
  - ◆ subscribe, monitor
- Networking performatives
  - ◆ register, unregister, forward, broadcast, pipe, break, transport-address
- Facilitation performatives
  - ◆ broker-one, broker-all, recommend-one, recommend-all, recruit-one, recruit-all

# KIF (Knowledge Interchange Format)

---

- 1階述語論理を拡張して、項の定義、メタ知識 (知識に関する知識)、集合、非単調理論などを記述できるようにした知識交換言語
- 1階述語論理の表現をLisp風の記法で表現
- 拡張部分
  - 集合・リストの表現setof, やlistof
  - ifやcondによる条件記述子
  - defobject, deffunction, defrelationを使ってオブジェクト、関数、関係を定義

# KIFによる表現の例

述語論理 / 数学の表記	KIF
(1) 項	
$x + y$	(+ ?x ?y)
$\{a, b, c, d\}$	(setof a b c d)
$\{x \mid \text{wheel}(x) \text{ made-in-japan}(x)\}$ ; 「日本製の車輪の集まり」	(setofall ?x (and (wheel ?x) (made-in-japan x)))
$x [\text{wheel}(x) \text{ made-in-japan}(x)]$ ; 「その日本製の車輪」	(the ?x (and (wheel ?x) (made-in-japan x)))
$x [\text{wheel}(x) \text{ made-in-japan}(x)]$ ; 「...は日本製の車輪である」	(kappa ?x (and (wheel ?x) (made-in-japan x)))
$x [\text{founded-year}(\text{vendor}(x))]$ ; 「 $x$ の製造者の設立年」	(lambda ?x (founded-year (maker ?x)))
if number( $x$ ) $x \neq 0$ then $1/x$ ; 「 $x$ が0でない数のとき、 $1/x$ (を与える関数)」	(if (and (number ?x) (not (= ?x 0))) (/ 1 x))
(2) 命題	
$p(a, b)$	(p a b)
$\forall x \exists y [\text{product}(x) \text{ vendor}(y) \text{ makes}(y, x)]$	(forall ?x (exists ?y (=> (product ?x) (and (vendor ?y) (makes ?y ?x))))

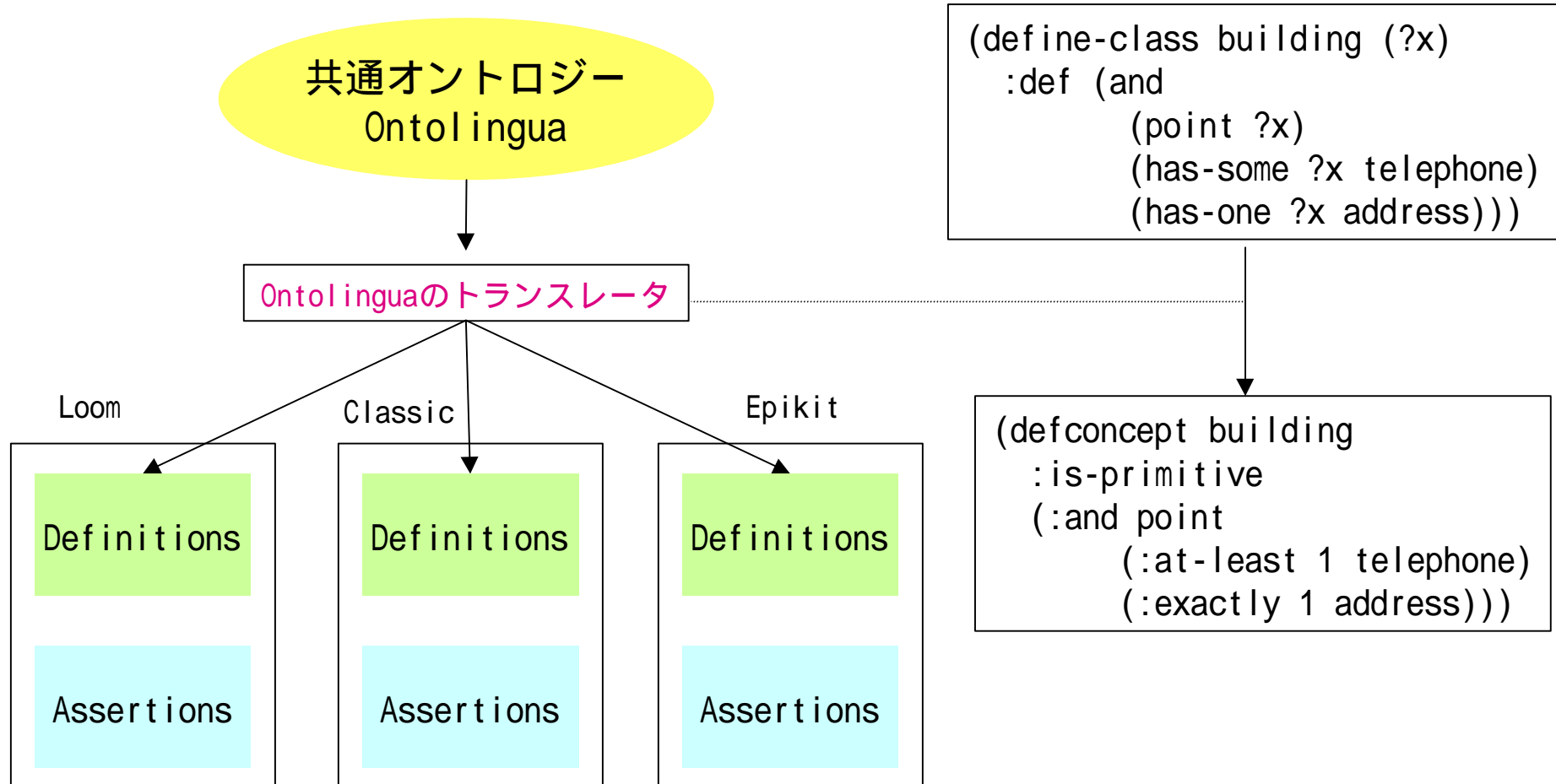
# Ontolingua

---

- Ontolinguaは、対象領域の概念的なモデリングとその表現のしかたに関する合意を明示的に表示するためのオントロジー記述言語
- KIFのなかの定義に関する表現を用いて、対象領域をオブジェクト、関数、関係の集まり（オントロジー）として形式化して定義



# Ontolinguaによる知識の共有へのアプローチ



# Ontolinguaによる記述の例

---

```
(define-class AUTHOR (?author)
: def (and (person ?author)
          (= (value-cardinality ?author AUTHOR.NAME) 1)
          (value-type ?author AUTHOR.NAME
                     (>= (value-cardinality ?author AUTHOR.DOCUMENTS) 1)
                     (<=> (author.name ?author ?name)
                          (person.name ?author ?name))))))
```

「?authorがauthorであるとは、?authorがpersonであり、単一値を取り、値のタイプがbiblio-nameであるAUTHOR.NAME属性と、1個以上の値を取るAUTHOR.DOCUMENTS属性をもち、?authorのauthor.nameとperson.nameが同一値となることを意味する。」

# オントロジーの設計 -- 設計の要件

---

- 明瞭性
  - オントロジーは定義された用語の意図した意味を効果的に伝達しなければならない。定義は客観的でなければならない。そのためには、形式を整えるとともに、できる限り完全な定義を与えることが望ましい。
- 結束性
  - 定義と整合性のある推論を促進する必要。少なくとも、基底となる公理は無矛盾でなければならない。
- 拡張可能性
  - 適用しようとしている領域に関する概念的な基礎を与える必要。利用者がオントロジーを拡張したり、特化することを可能にしなければならない。
- 最小限のコーディングバイアス
  - インプリメンテーションに独立にした知識レベルの表現を使うべき
- 最小限のオントロジーのコミット
  - モデル化しようとしている世界に関する先験的な仮定をできるかぎりなくすこと