Representing Approximate Concepts Using Metaclass Frames in Protege

Hong-Gee Kim

Dankook University, Korea

hgkim@dku.edu

Kyung-Mo Park

Kyunghee University, Korea

Agenda

❐ Approximation & Approximate Concepts

❐ Meta Modeling in Protégé-2000

❐ Reification of Parts

❐ Metaclass Normalization

❐ Qua-Type Creation

❐ Sense Articulation in Polysemous Concepts

❐ Conclusion

Meaning of Approximation

❐ An entity is *approximate* if it can be further refined by learning more things about it (McCarthy, 2000).

❐ Approximate concepts are cognitively intuitive thus essential for common sense knowledge representation.

❐ Reasoning with approximate concepts is computationally efficient by ignoring unnecessary details.

❐ Articulation is the inverse process of approximation.

❐ Different levels of conceptualization are possible for the same object.

Kinds of Approximate Objects

❐ *Static* aspect vs. *Dynamic* aspect of representing knowledge
  ➢ **Concepts and approximate objects are static notions**
  ➢ **Conceptualization and approximation are dynamic notions**

❐ Fuzzy concept
  ➢ **Vague features are represented**

❐ Schema, Scripts, Frames
  ➢ **They can conceptualize similar types of individuals effectively**

❐ Metaclass
  ➢ **Meta-level approximation**

Meta-Level Approach to Conceptualization

❒ Different levels of ontology modeling

  ➢ **Instance level**

  ➢ **Class level – a meta level description for an instant object**

  ➢ **Metaclass level – a meta level description both for a class entity and for a metaclass entity**

❒ A metaclass is a context

  ➢ **The same class can be conceptualized differently if it can be reified from different metaclasses**

  ➢ **Articulation is "slicing and dicing" as well as "drill-down"**

Metaclass Modeling

❒ Meta concepts

  ➢ **'Meta' stands for an idea of a higher level of generality and abstractness**

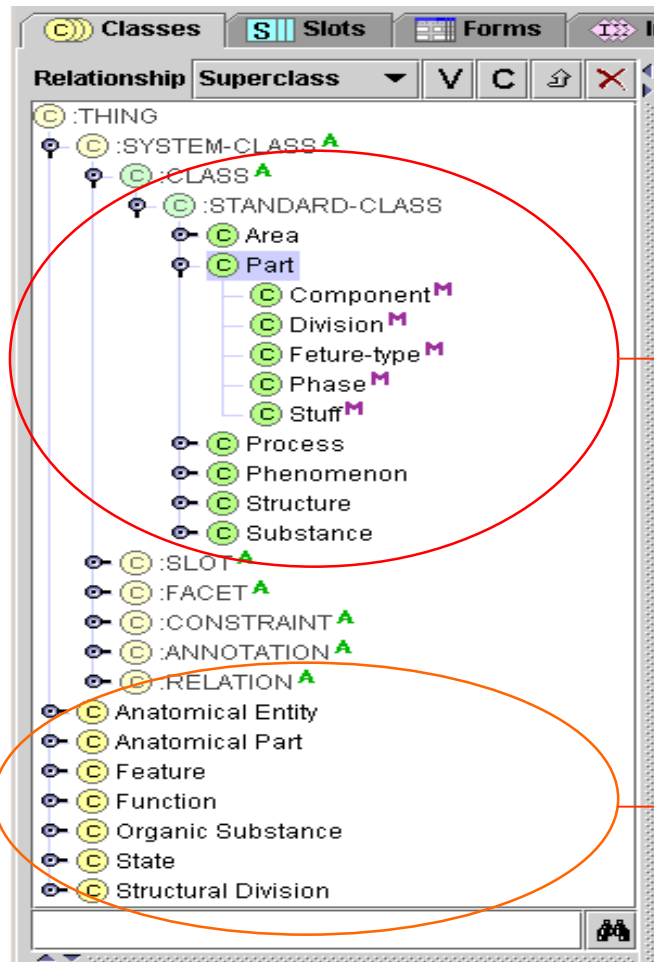 ➢ **Metarule, Metaknowledge, Meadata, Metamodel, Metaclass etc.**

❒ A difficult question

  ➢ **If a metaclass is also a class, then what concepts are represented in metaclasses and what in regular classes?**

Metaclass in Protégé-2000

❒ "A metaclass is a template that is used to define new classes in an ontology"

❒ A metaclass is a class whose instances are themselves classes

❒ The metaclass architecture in Protégé-2000 provides high flexibility since users can define metaclasses explicitly

❒ More than one class can share the same user-defined metaclass

❒ Protégé-2000 does not allow multiple classification

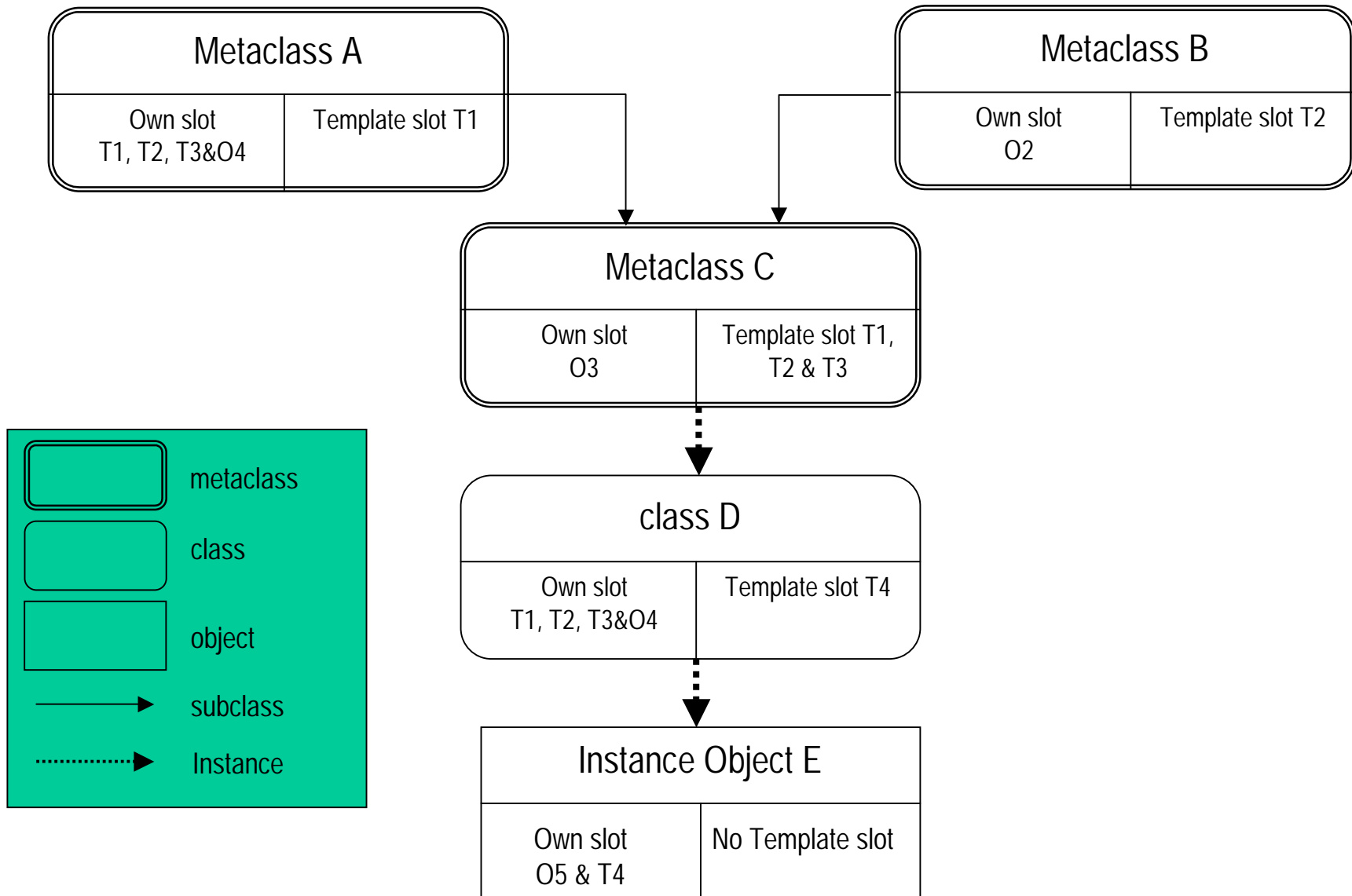➢ **A class is not allowed to be an instance of several metaclasses**

Metaclass vs. class hierarchy in Protégé-2000


Metaclass hierarchy

class hierarchy

Propagation of template and own slots

Metaclass A	
Own slot T1, T2, T3&O4	Template slot T1

Metaclass B	
Own slot O2	Template slot T2

Metaclass C	
Own slot O3	Template slot T1, T2 & T3

class D	
Own slot T1, T2, T3&O4	Template slot T4

Instance Object E	
Own slot O5 & T4	No Template slot

Legend:

- metaclass
- class
- object
- ⟶ subclass
- ┈┈▶ Instance

Meta Modeling Principle

❒ Top-level ontologies are usually independent of a particular domain and put in metaclasses

❒ Top-level ontologies should provide a general but normalized way of describing domain-specific vocabularies

- ➢ **Ontological principles (or meta-properties) specifying own slots of class-level concepts in a domain**
- ➢ **Higher-level meta concepts are possible**
- ➢ **A matter of degree in distinction between top-level and domain ontologies (domain specific top-level vs. domain independent top-level)**

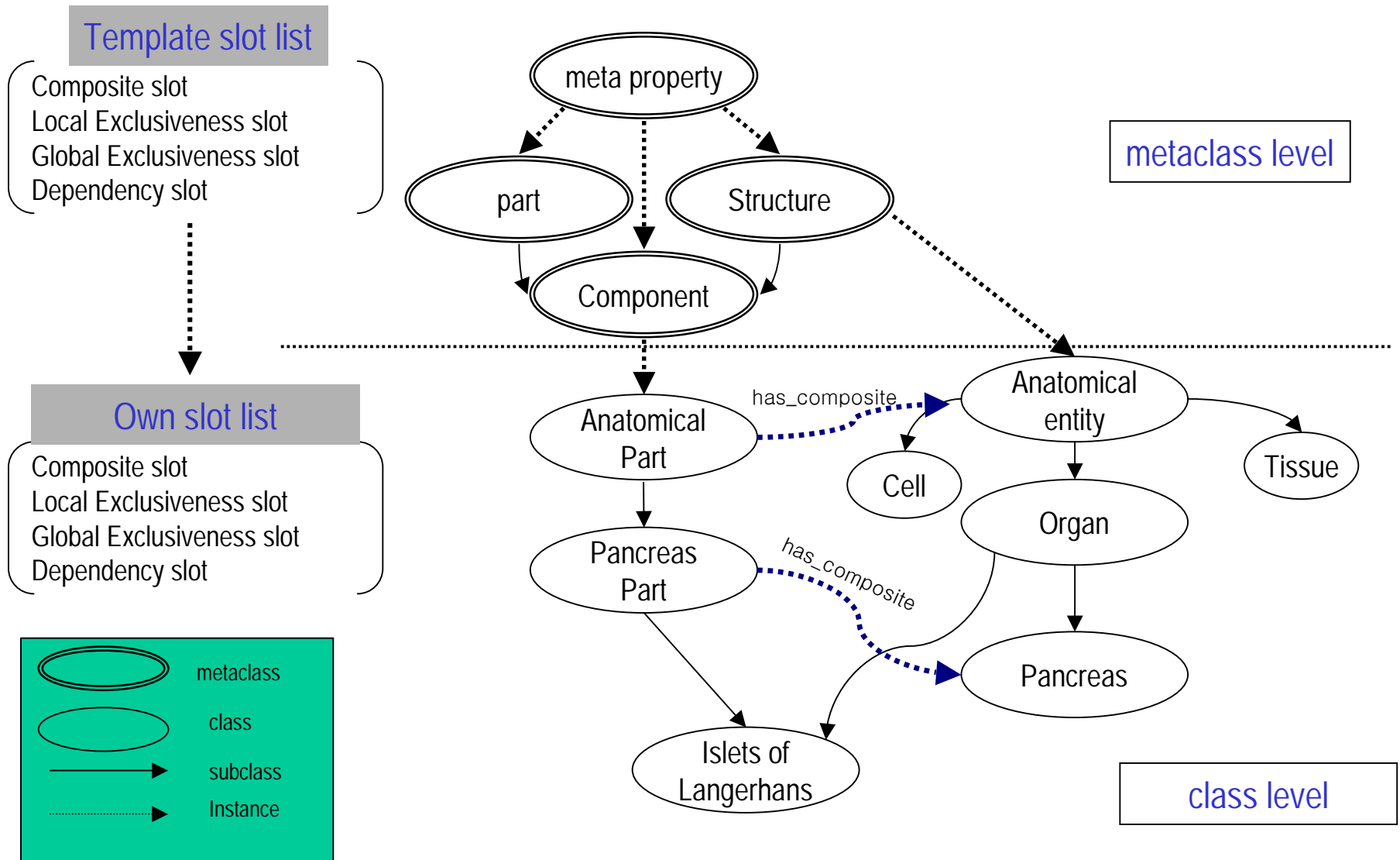❒ Ontology engineers working on domain specific ontologies may ignore all the details of top-top-level ontologies

Representation of Metaclass Hierarchy on Parts
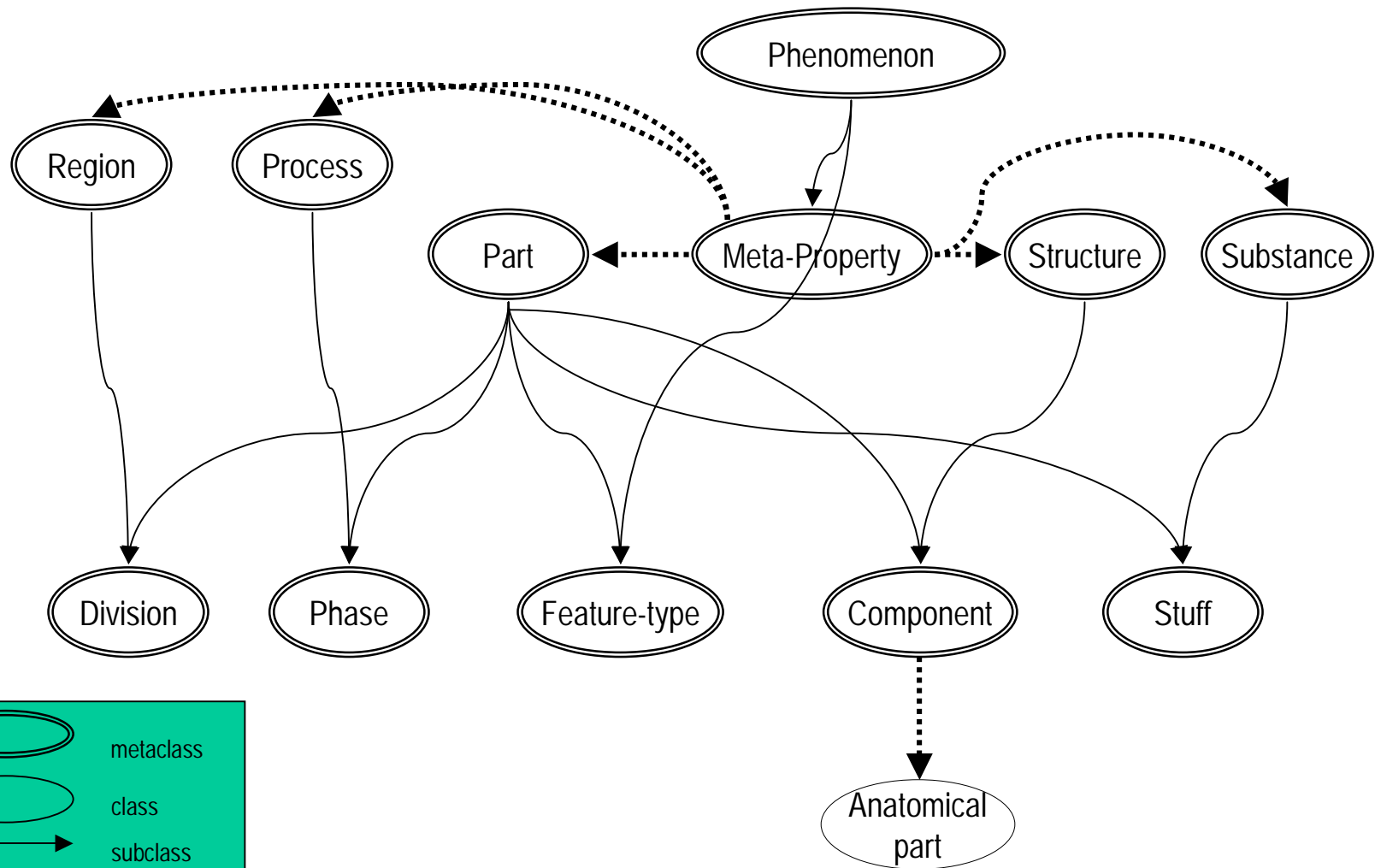
Reification of Parts

❏ A part functions as a placeholder for user intuitions that have not been articulated

❏ Additional attributes for part-whole relations should be included in the part metaclasses

- ➤ **Component exclusiveness (and its opposite, sharing) is about the generalized maximum cardinality on the component side**
- ➤ **Two aspects of exclusiveness: Global vs. Local**
- ➤ **Dependency (and its opposite, independency) is about the generalized minimum cardinality**

❏ Actual Parts vs. Potential Parts

Representation of Anatomical Components

Template slot list

Composite slot
Local Exclusiveness slot
Global Exclusiveness slot
Dependency slot

meta property

part

Structure

Component

metaclass level

Own slot list

Composite slot
Local Exclusiveness slot
Global Exclusiveness slot
Dependency slot

Anatomical Part

has_composite

Anatomical entity

Cell

Tissue

Pancreas Part

has_composite

Organ

Pancreas

metaclass

class

subclass

Instance

Islets of Langerhans

class level

Reification of Parts

❏ An anatomical entity not only can have an independent existence but can be conceptualized in its own → *actual entity & potential part*

  ➢ **But, the conceptualization of components are possible only in the context of the mereological relationship with an actual entity → *possible entity & actual part***

❏ Reasoning along part-whole relations is notoriously tricky

  ➢ **Regular transitivity does not work**

  ➢ **Upward distribution vs. downward distribution**

❏ Reasoning problems may be solved by applying pre-specified metaclasses to different sorts of parts

  ➢ **Not necessarily making too many part-whole relations that are difficult to understand**

Metaclass Normalization in Parts

❒ The mereological attributes should be included in the own slot at the class level not to be propagated into the instance level

❒ Once an entity is conceptualized as a part of another entity, the mereo-topological properties are also gained from the proper part metaclass

❒ Updated conceptualization (or concept articulation) should result in the schema change
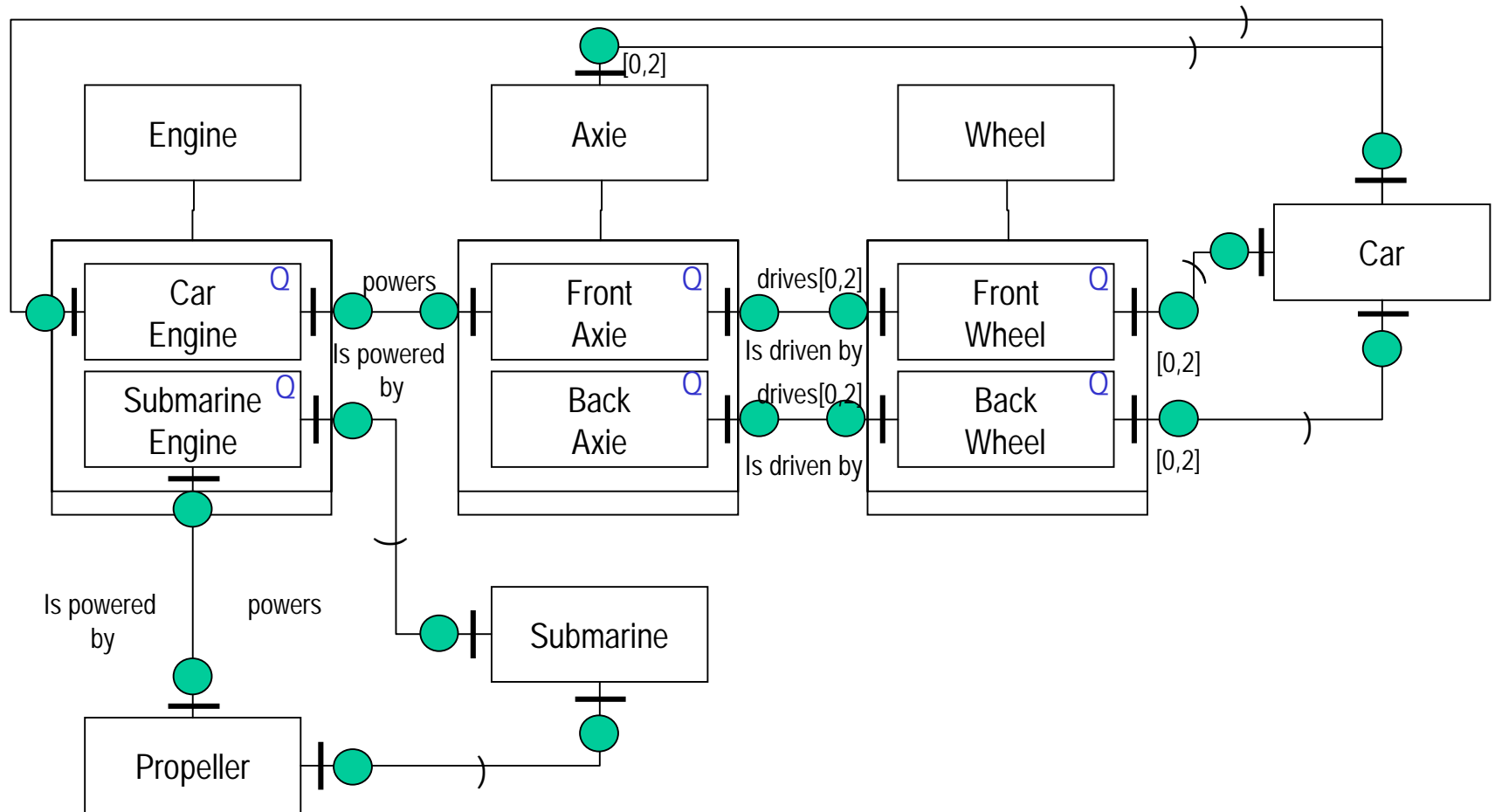 ➢ **The dynamic aspect of approximation and articulation**

Metaclass Change

❏ Protégé-2000 does not allow multiple classification
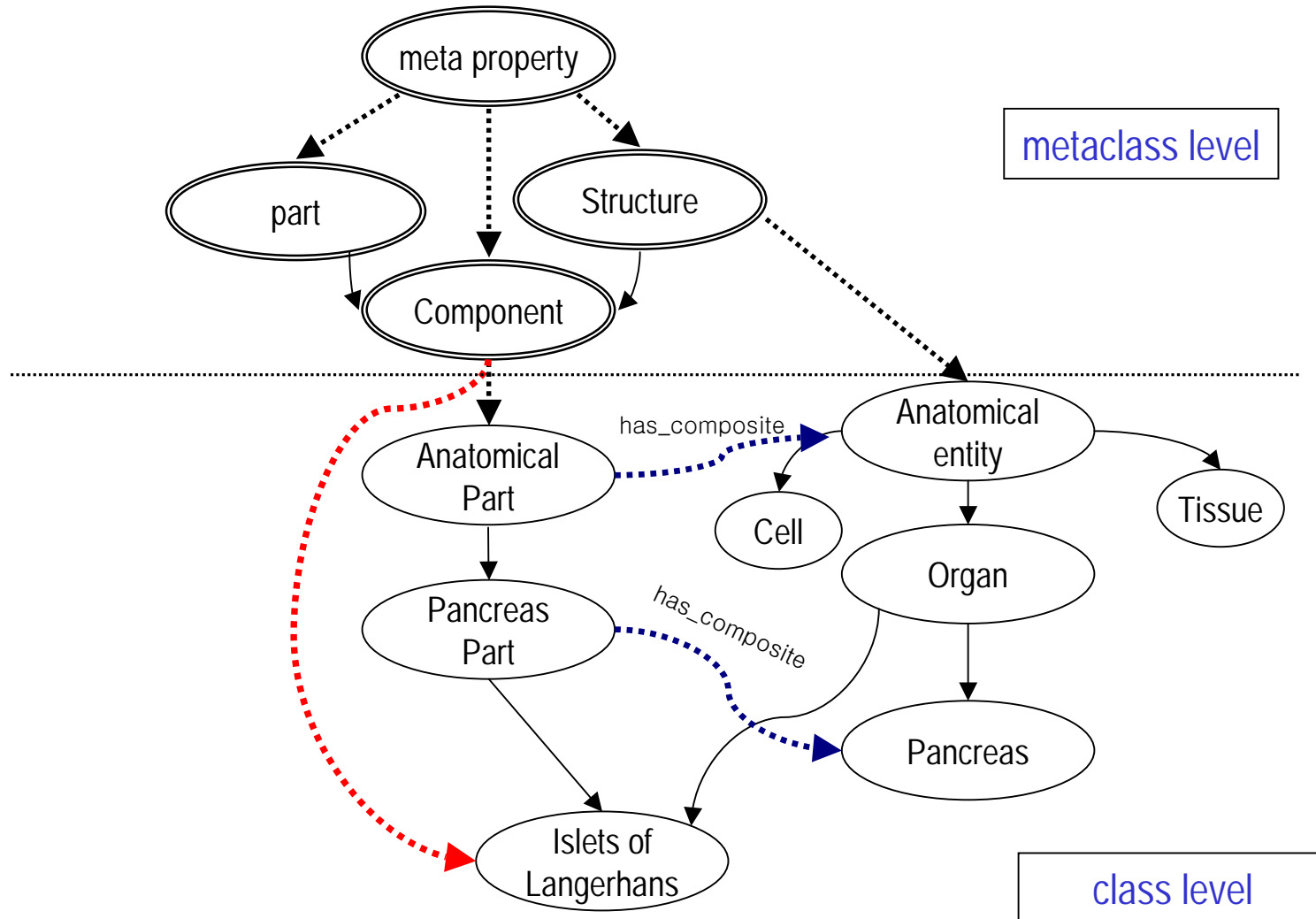
Parts as Qua-Types

❒ Definition of qua-types

➢ **A subtype that has been created solely because of a specialized relationship (Odell, 1998)**

Creation of Qua-Types Articulating Part Concepts

❑ An algorithm to create a qua-type

➢ **If an entity becomes a component of two or more entities & it is globally sharable, then the name of its composite is prefixed to its name & it becomes globally exclusive**
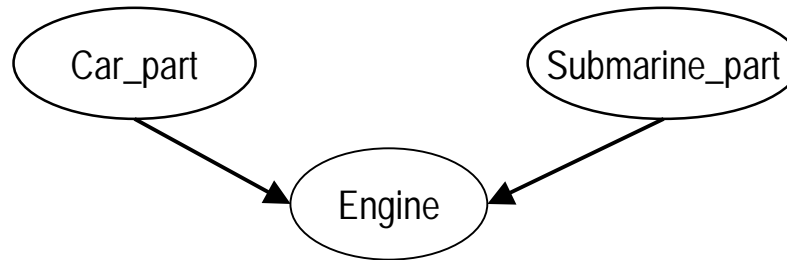
Creation of Qua-Types Articulating Part Concepts

□ An algorithm to create a qua-type

➢ **If an entity becomes a component of two or more entities & it is globally sharable, then the name of its composite is prefixed to its name & it becomes globally exclusive**
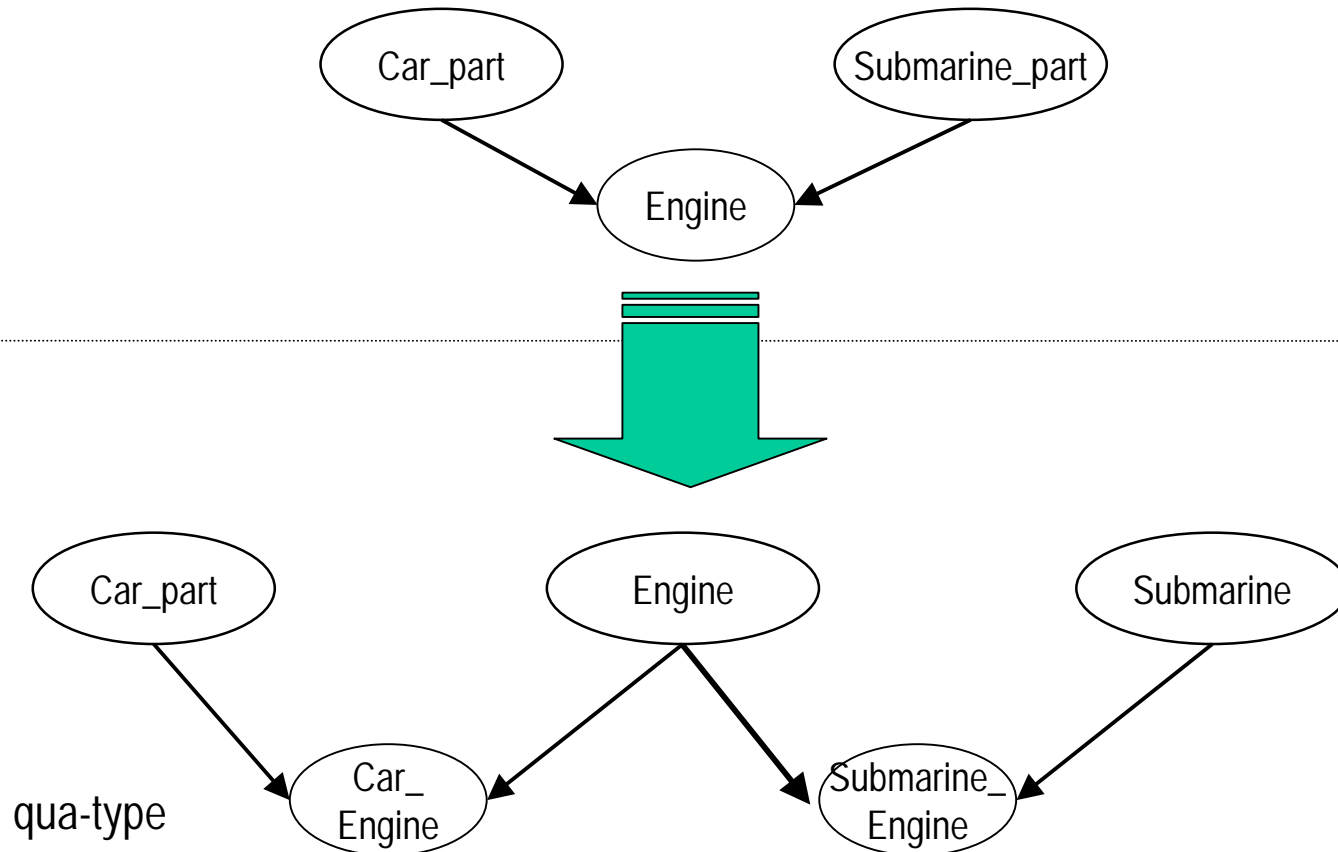
Polysemy in Medical Concepts

❐ Terminological economy
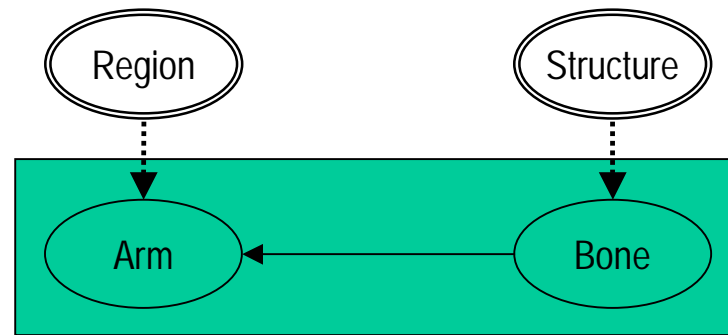
  ➢ **Humans has a tendency to use already known terms**

❐ A problem of semantic mismatches

  ➢ **An important issue for ontology integration**

❐ 'Inflammation'

  ➢ **A physiological** *function*
  ➢ **A symptomatic** *condition*
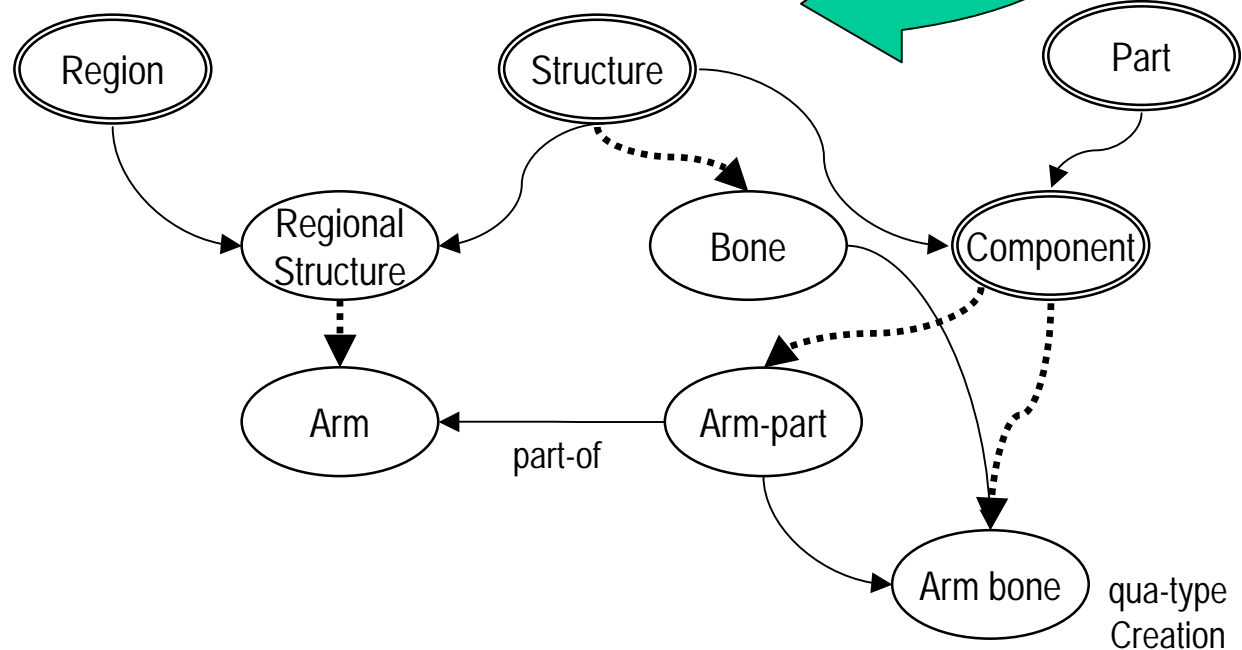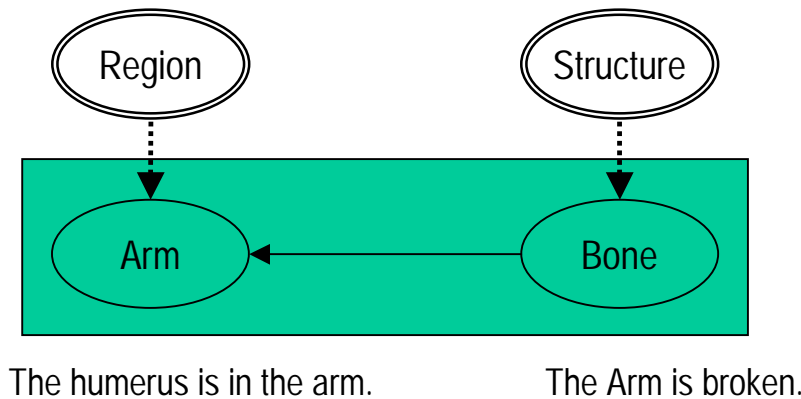 ➢ **The** *area* **of an organ that bears an inflammation process**

Sense Articulation in the Metaclass Architecture

The humerus is in the arm.   The Arm is broken.

Sense Articulation in the Metaclass Architecture

The humerus is in the arm.   The Arm is broken.

Conclusion

❒ A metaclass in Protégé-2000 is an approximate entity

❒ The metaclass architecture provides a flexible way to design concepts

❒ A meta-level modeling principle should be studied to make full use of Protégé metaclasses

❒ Additional reasoning mechanisms can be implemented to solve some difficult problems such as partitive reasoning, polysemy representation, and concept articulation.

References

▪ Artale, A., Franconi, E., Guarino, N., & Pazzi, L, Part-whole relations in object-centered systems: an overview, *Data & Knowledge Engineering* 20: 347-383, 1996.

▪ McCarthy, J. Approximate objects and approximate theories, In Cohn, A., Giunchiglia, F., & Selman, B., eds., *KR2000: Principles of Knowledge Representation and Reasoning, Proceedings of the Seventh International Conference*, 519-526, 2000.

▪ Noy, N.F., Fergerson, R., & Musen, M., The knowledge model of Protégé-2000: combining interoperability and flexibility, *12th International Conference on Knowledge Engineering and Knowledge Management*, 2000.

▪ Odell, J., *Advanced Object-Oriented Analysis and Design Using UML*, Cambridge Univ. Press, 1998.

▪ Schulz, S. & Hahn, U., A knowledge representation view on biomedical structure and function, *Proceedings of the AMIA 2002 Annual Symposium*, 2002.

▪ Sowa, J., *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publisher, 2000.