Levels for Conceptual Modeling

Claudio Masolo

Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy

masolo@loa-cnr.it

OntoCom @ ER – November 2011

1 Isa and inheritance

- *Abstraction (specification)*: starting from a given domain, objects are grouped in classes (types) according to the properties (attributes) they have in common:
  - more *general* a class is, *less* properties its instances share;
  - more *specific* a class is, *more* properties its instances share.
- Inheritance has no problems if we consider this basic intuition.

2 Difficulties with isa and inheritance /1

- *Hiding/blocking*. `Student` $\Rightarrow$ `Person` but `Student` has no weight.

  `Employee` $\Rightarrow$ `Person` but `Employee` has no home phone number.

- *Overriding*. `Statue` $\Rightarrow$ `AmountOfMatter` but the price of statues could be different from the price of mere amounts of matter.

  `Employee` $\Rightarrow$ `Person` but the phone number of an employee could be different from his/her personal one.

- *Ambiguous inheritance and conflicting attributes*.
  `WorkingStudent` $\Rightarrow$ `Employee` and `WorkingStudent` $\Rightarrow$ `Student` but room of John when employee $\neq$ room of John when student.

  `Quacker` $\Rightarrow$ `Person` and `Republican` $\Rightarrow$ `Person` but Nixon as quacker is pacifist while as republican is not.

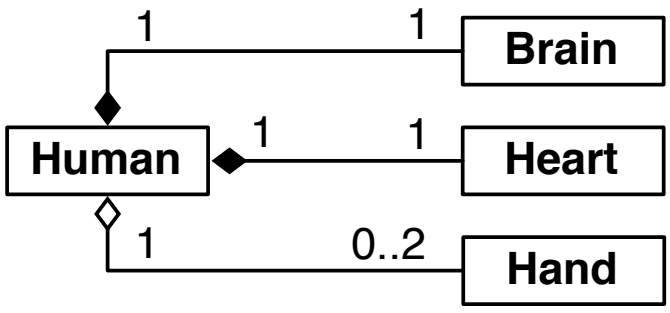
3 Difficulties with isa and inheritance /2

- *Counting.* Suppose `Customer` $\Rightarrow$ `Person` and `Customer` has the additional (w.r.t. `Person`) attribute `CustomerCode`.
  - The same person can be a customer of different persone, therefore we cannot *count* persons to count customers.
  - Can we count customers, i.e. entities *identified* by codes?

4 General questions

- Are the previous difficulties symptomatic of isa overloading/misusing?
- Is it possible to find an alternative mechanism to *structure* types that
  - is general as Isa is,
  - it is compatible with Isa,
  - allows for a controlled inheritance mechanism,
  - does not suffer of the previous difficulties?

5 Parthood (aggregation)



- Each human necessarily has exactly one brain, exactly one heart, and at most two hands (hands are not necessary for humans).
- Some proposals consider a further distinction: humans have necessarily *specific* brains but not *specific* hearts (heart transplantation).
- Some proposals manage attribute inheritance through parthood.
- Less addressed question: is it enough to have a brain and an heart (and maybe two hands, one trunk, etc.) to have an human?

6 Constitution

- Statues are *constituted* by amounts of matter.
- `Statue` $\Rightarrow$ `AmountOfMatter`, i.e. are statues amounts of matter?
  - Problem. Statues can *change* their material support across time.
- `AmoutOfMatter` $\overset{1}{\longrightarrow\!\!\!\!\blacklozenge}{}^{1}$ `Statue`, i.e. are amounts of matter necessary parts of statues?
  - Problem. *Extensionality* of parthood

    $$\mathsf{PP}xy \rightarrow \exists z(\mathsf{P}zy \wedge \neg \mathsf{O}zx)$$

    what makes the difference btw amounts of matter and statues?

    [what makes the difference between four legs plus a top and a table?]

7 Individual roles /1

- Are these objects?
  1. ‘The president of Italy’
  2. ‘The director of the Berlin Philharmonic’

     [‘The Berlin Philharmonic’]
  3. ‘The Amazon customer #125678’
- *General* vs. *specific* dependence: presidents and directors can change their ‘substratum’ while customers relate to one simgle person.

8 Individual roles /2

- `Customer` $\Rightarrow$ `Person` and `President` $\Rightarrow$ `Person`?
  - Migration problems + presidents can be represented by different persons at different times.
  - Is `Person` an abstraction from `Customer`, `Person`, etc., i.e. its instances are customers, students, etc.?
- `Person` $\overset{1}{\longrightarrow}\!\!\!\!\!\!\overset{*}{\blacklozenge}$ `Customer` and `Person` $\overset{1}{\longrightarrow}\!\!\!\!\!\!\overset{*}{\blacklozenge}$ `President`?
  - What makes the difference between persons and customers?
    1. Properties, tropes, relators, etc. to be added to the domain.

       [$\sim$ do tables require some structural constraint btw legs and tops?]
    2. New objects to represent the "many faceted nature" of some kinds of entities.

9 General idea

- Follow a multiplicative approach that puts change at the core of the analysis and generalizes parthood to account for:
  - hearts are *aggregations* of, but different from, pluralities of cells;
  - the Amazon customer #125678 is different from John;
  - today, the president of Italy is only *represented* by Napolitano;
  - statues are *constituted* by, but different from, amounts of matt., paperweights are *constituted* by, but different from, pebbles.

- No properties, roles, relators, or new objects are necessary.
- Persons are not parts customers or presidents.

10 Grounding /1

- Intuitively, x *grounds* y at t if, at t, to exist, y requires x but, vice versa (at t) x does not require y.
- Is asymmetric, transitive, down linear and it does not satisfy neither the strong nor the weak supplementation principles.

  [For a FOL characterization see the paper or KR2010]
- It does not necessarily require reduction.
- In between *pure existential dependence* and *constitution*.

11 Grounding /2

- To exist, customers require both companies and persons.
- Grounding aims at capturing only the specific existential dependence between customers and persons.
- Intuitions:
  - the customer is spatially co-located with John not with Alitalia;
  - relations are "*directed*":
    - there is a difference between "John is a *customer of* Amazon" and "Amazon is a *supplier for* John";
    - there is a change in *perspective* from *John seen as a customer of Alitalia* to *Alitalia seen as a supplier for John*.

12 Specific vs. generic grounding between classes

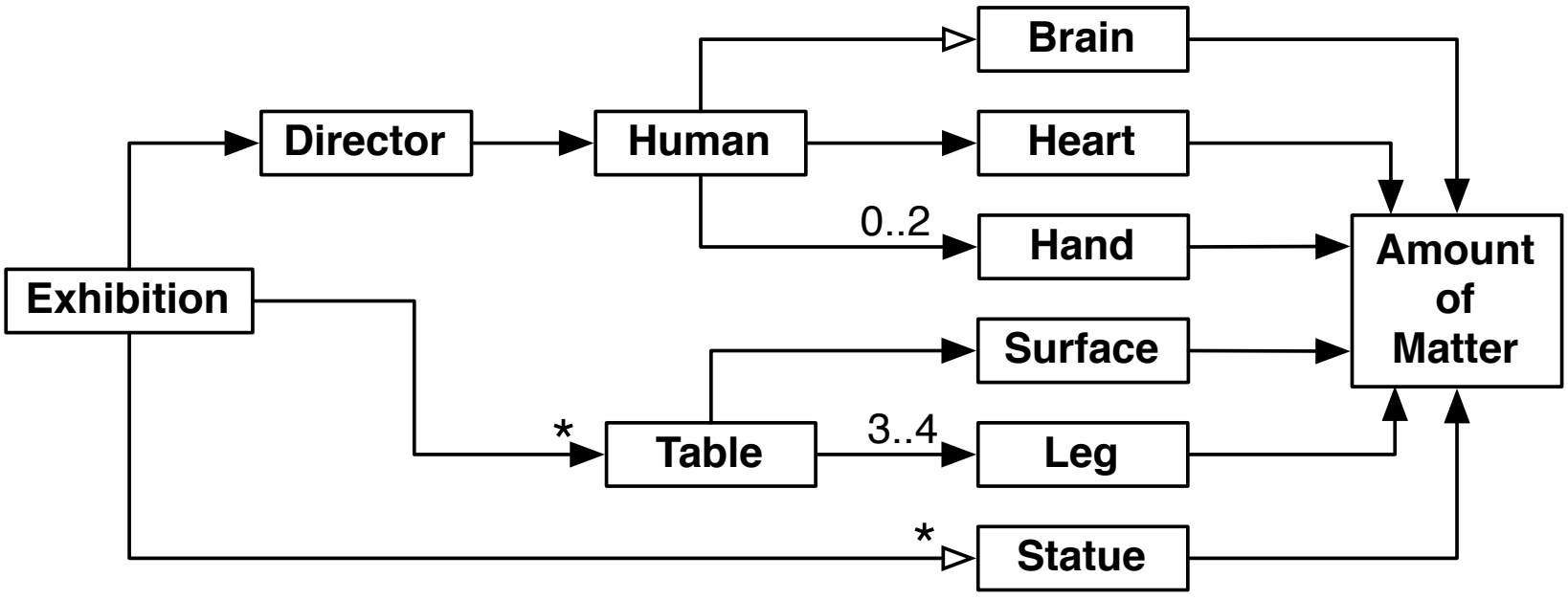
- $\mathtt{T}_1$ is *specifically grounded* on $\mathtt{T}_2$ ($\mathtt{T}_1 \triangleright \mathtt{T}_2$), if every $\mathtt{T}_1$-object is grounded on a *single* $\mathtt{T}_2$-object during its whole life;
  e.g. `Customer` $\triangleright$ `Person`.
  - Often motivated by emergent properties;

    [note: `Customer` is now a *rigid* type]
- $\mathtt{T}_1$ is *generically grounded* on $\mathtt{T}_2$ ($\mathtt{T}_1 \blacktriangleright \mathtt{T}_2$), if every $\mathtt{T}_1$-object is grounded on on some, but not necessarily the same, $\mathtt{T}_2$-object;
  e.g. `Statue` $\blacktriangleright$ `AmountOfMatter`.
  - Often motivated by different persistence conditions.
- These definitions can be extended to take into account cardinality constraints.

13 Inheritance through grounding

- Often it is taken for granted that:
  1. the intension of a type reduces to the set of its properties;
  2. if the *intension* of $\mathtt{T}_1$ includes the *intension* of $\mathtt{T}_2$ then the *extension* of $\mathtt{T}_1$ is included in the *extension* of $\mathtt{T}_2$.
- But grounded types are *disjoint* therefore, from (1)-(2), grounding, in general, does not allow for inheritance of (all) attributes.
- By relaxing (2), seeing inheritance as a mechanism that helps in "factoring out shared specifications", then
  - inheritance not only through isa but also through grounding;
  - the inheritance through grounding can be completely controlled.

14 An example of levels



15 Concluding remark

- Grounding allows also for a new perspective on "*abstraction*" that I did not explore in this work.
  - Are parts abstracted from (and therefore dependent on) wholes, i.e. *whole to parts* vs. *parts to whole*?

    E.g., *brains depend on humans* vs. *humans depend on brains*,
    brains are *carved out* from humans by an abstraction process.
  - More generally, what about *perspectives on a given object*?