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 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

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

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

- *Hiding/blocking*. `Student` → `Person` but `Student` has no `Weight`.

  `Employee` → `Person` but `Employee` has no `HomePhone`.

- *Multiple inheritance: conflicting attributes + ambiguous inheritance*. Attributes/methods may be declared in different superclasses.

  `WorkingStudent` → `Employee` and `WorkingStudent` → `Student` but room of John when employee ≠ room of John when student.

  `Quacker` → `Person` and `Republican` → `Person` but Nixon as quacker is pacifist while as republican is not.

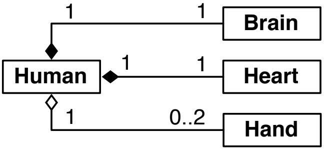
3 Difficulties with isa and inheritance /2

- *Counting.* Suppose `Customer` $\rightarrow$ `Person` and `Customer` has the additional (w.r.t. `Person`) attribute `CustomerCode`.
  In addition, suppose that the same person, say John, is a customer of different companies.
  - We cannot *count* persons to count customers, what do we count then?
  - Customer codes seem to *identify* customers, i.e. customers have an unique code while a person that is a customer of different companies has different codes. Then which objects (of the domain) customer codes identify?

4 General questions

- Are the previous difficulties symptomatic of isa overloading/misusing?
- Is it possible to find an alternative (with respect to Isa and taxonomies) mechanism to *structure* classes (object 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 and change

- Statues are constituted by amounts of matter.

- `Statue` $\rightarrow$ `AmountOfMatter`, i.e. are statues amounts of matter?
  - Problem. Statues can *change* their material supports across time (note that this is not migration).

- `AmoutOfMatter` $\overset{1 \quad 1}{\longrightarrow}\blacklozenge$ `Statue`, i.e. are amounts of matter part of statues?
  - Problem. *Extensionality* of parthood:

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

7 Roles again /1

- Are the following entities individuals objects?
  1. ‘The president of Italy’
  2. ‘The director of the Berlin Philharmonic”
  3. ‘The Amazon customer #125678’
- *General vs. specific dependence*: persons that ‘represent’ (1) and (2) can change trough time while (3) is always related to the same person.

8 Roles again /2

- Customer → Person and President → Person?
  - Migration problems + presidents can be represented by different persons at different times.
  - Are the instances of Person just customers, students, etc., i.e. is Person an abstraction from Customer, Person, etc.?
- Person $\xrightarrow{1\quad *}\!\blacklozenge$ Customer and Person $\xrightarrow{*\quad *}\!\blacklozenge$ President?
  - *Extensionality*: what makes the difference between a person and a customer (similarly for presidents)?

    1. The property of being enrolled in an university: properties, tropes, relators, or … must be introduced in the domain.
    2. "[A]n object is implemented by multiple instances which represent its many faceted nature. Those instances are linked together through aggregation." (Al-Jadir & Leonard, 1999)

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 Claudio;
  - 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, or relators are necessary.
- Persons are not mereologically included in 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.
- In between *pure existential dependence* and *constitution*.
  - In $\Box(\mathsf{E}y \to \mathsf{E}x)$ there are no specific link between x and y.

- The paper proposes a FOL characterization of grounding.

11 Grounding /2

- To exist, customers require both companies and persons (and the relation *customer-of* between them).
- Grounding aims at capturing only the specific existential dependence between customers and persons.

- Intuition:
  - 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*;
  - the *customer* is spatially co-located with John, while the *supplier* is spatially co-located with Alitalia.

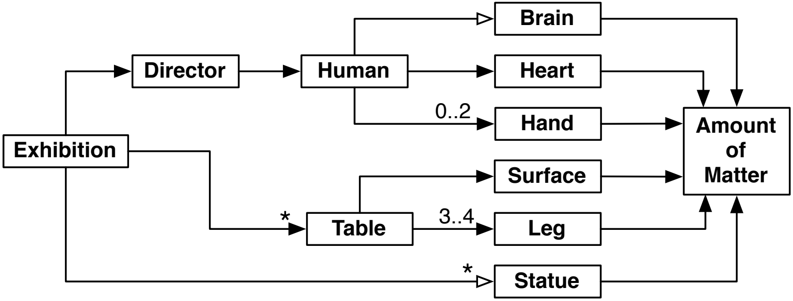
12 Specific vs. generic grounding between classes

- $\mathtt{T}_1$ is *specifically grounded* on $\mathtt{T}_2$ ($\mathtt{T}_1 \rhd \mathtt{T}_2$), if every $\mathtt{T}_1$-object is grounded on a *single* $\mathtt{T}_2$-object during its whole life; e.g. `Customer` ▷ `Person`.
  - Often motivated by emergent properties;
  - `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` ▶ `AmountOfMatter`.
  - Often motivated by different persistence conditions.
  - Grounding does not necessarily require reduction.
- These definitions can be extended to take into account cardinality constraints.

13 Inheritance through grounding

- Isa and grounding are different from the inheritance mechanism.
- If inheritance helps in "factoring out shared specifications", then attributes can be inherited not only through isa but also through grounding (that actually establishes a strict existential dependence between the instances of grounded types).
- Grounded types are rigid and *disjoint* therefore the inheritance of attributes through grounding can be completely controlled.

14 An example of levels



- Isa can be added, but it is internal to a single level.

15 Conclusions and further work

- I just sketched a different perspective on organizing object types that could be quite helpful in solving some classical problems of inheritance through isa.
- From my point of view, the idea of grounding (and existential dependence) is quite powerful, and it provides a new perspective on *abstraction* that I did not explore:
  - are parts abstracted from (and therefore dependent on) wholes, i.e. *whole to parts* instead of *parts to whole*?

    E.g., *brains depend on humans* instead of *humans depend on brains*, brains are *carved out* from humans by an abstraction process.
  - What about *perspectives on a given object*?