Levels for Conceptual Modeling

Claudio Masolo

Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy
masolo@loa-cnr.it

Abstract. Usually object types are organized in taxonomies by means of a *specialization* relation (also called *subtyping* or *isa*) 'implemented' by means of *inheritance*. This paper proposes a (non-incompatible) alternative to taxonomies that relies on three primitives: *grounding*, a specific kind of factual existential dependence, extensional atemporal *parthood*, and *existence* at a time. On the basis of these relations, *specific*, *generic*, and *compositional* grounding relations between object types are introduced. By clearly separating the objects from the substrata on which they are grounded, these grounding relations allow to stratify object types in *levels* and to manage inheritance in a flexible way. In particular, this approach helps to avoid isa overloading and to overcome some classical difficulties related to inheritance, e.g. attribute overriding, attribute hiding, or dynamic and multiple classification and specialization, that are relevant aspects especially in modeling *roles*.

Keywords: Grounding, Dependence, Levels, Taxonomies, Inheritance

Classification schemes – taxonomies based on subtyping (isa) among object types – and *inheritance* are central notions in conceptual modeling (CM) and in object-oriented modeling. By assuming, for instance, that Statue is a subtype of Amount Of Matter[1], Statue inherits all the attributes and associations[2] of Amount Of Matter. However, new attributes can be introduced. For instance, Statue, but not Amount Of Matter, could have the attribute Style. Similarly *roles*[3] like Student, Customer, or President could be modeled as subtypes of Person. Student, but not Person, has a Matriculation. Customer, but not Person, has a Code, etc. This powerful mechanism of inheritance faces some well known problems. Statue and Amount Of Matter could have different values for the 'same' attribute, e.g. Price: a statue could be more expensive than a brute piece of bronze. Customer, differently from Person, could not have Weight or Place Of Birth. Attribute *overriding* and *hiding* try to manage these problems. Furthermore, roles can be played by objects with 'incompatible' attributes. For instance, both companies and persons can be customers, but Customer is neither

[1] Amounts of matter are concrete entities, pieces of matter, specific sums of molecules, e.g. the piece of gold that now constitutes my ring, not 'the gold' or 'gold'.

[2] In this paper I will focus only on data modeling and not on behavior modeling.

[3] The term *role* indicates here a specific kind of object types (properties). Roles intended as 'parts in relationships' are close to *relational roles* (see [12]).

a subtype nor a supertype of both Company and Person. "[W]e have the paradoxical situation that, from the extensional point of view, roles are supertypes statically, while dynamically they are subtypes" ([17], p.90). While keeping the same domain, this problem can be managed by adding new objects types, e.g. Private Customer (subtype of both Person and Customer) and Corporate Customer (subtype of both Company and Customer) [7], or by introducing dynamic and multiple classification and specialization (see [17] for a review). Alternatively, more *permissive* or *multiplicative* approaches extend the domain with new entities. Steimann [17] separates *natural types* (e.g. Person) from *role types* (e.g. Customer). Roles are *adjunct instances* linked by a *played-by* relation to their players (the persons or companies in the case of customers). The object and its roles form an aggregate and "the dynamic picking up of a role corresponds to the creation of a new instance of the corresponding role type and its integration in a compound, and dropping a role means releasing the role instance from the unit and destroy it" ([17], p.91). In object-oriented database management systems, by distinguishing *specialization*, an abstract concept, from *inheritance*, a mechanism that implements specialization, [1] systematically multiplies the instances in the presence of a subtype relation. If P is a subtype of Q, then the creation of an object p of type P produces the creation of an object q of type Q plus a link between them that allows p to inherit attributes from q. An object then is implemented "by multiple instances which represent its many faceted nature. Those instances are linked together through aggregation links in a specialization relation" ([1], p.561). The attributes are locally defined and stored but additional ones can be inherited via the links between the instances. From a more foundational perspective, multiplicative approaches have been investigated to solve the *counting problem* [9]. For instance, to count the Alitalia passengers (during 2010), one cannot just count the persons that flew Alitalia (during 2010). By adding *qua-entities* [12], (sum of) *relational tropes* [7], or *role-holders* [13] – entities that *inhere in* (a sort of existential specific dependence), but are different from, the players (see Section 1 for more details) – the counting problem is solved. In philosophy, multiplicativism is often considered also in the case of statues, organisms, tables, etc. (see [14] for a review and [3] for a recent defense). Interestingly, qua-entities have been originally introduced in this contest [6]. As in the case of roles, statues and amounts of matter have different properties (in particular causal properties) and different persistence conditions. The amount of matter that constitutes a specific statue can change through time. Or, an amount of matter can constitute some statue only during a part of its life, when it is statue-shaped. Therefore, some authors assume that statues are *constituted by* (a sort of existential dependence), but different from, amounts of matter.

Taxonomies are undeniably an important conceptual tool to organize object types according to the set-theoretical inclusion between their extensions. But it is not the only one. This paper proposes a *complementary* structuring mechanism founded on a specific kind of *existential dependence* called *grounding*. This mechanism allows to account for both roles and material objects with a flexible management of inheritance that helps to avoid isa overloading and misuse.

1 Statues, customers, presidents, and tables

Let us assume that statues can change their material support through time while maintaining their shape, i.e. the shape, not the material support, is *essential* for statues. It follows that Statue is not a subtype of Amount Of Matter. One can represent 'being a statue' as a binary predicate with a temporal argument, $\mathsf{Statue}_t x$ stands for "at the time t, the amount of matter x is a statue (is statue-shaped)" (d1)[4]. According to (d1), 'being a statue' becomes a sort of (relational) role played by amounts of matter. Counting seems unproblematic: the statues present at t are the amounts of matter that are statue-shaped at t. However, problems arise by considering a non atomic time, e.g. the whole 2010. A statue could change its material support during 2010, i.e. we could have two amounts of matter that are statue-shaped during 2010 but only one single statue. On the other side, if the same amount of matter, at a given time, is the support of two different statues, then we have one amount of matter but two statues. This sounds wrong because one usually excludes co-location of statues. Different are the cases of artifacts intended as (material) objects with an assigned (by the creator) functionality [4], and roles where, for example, at a given time the same person can be the customer of different companies or a multiple-customer of the the same company (see [12] for more details). The strategy to multiply predicates, e.g. one specialization of Statue for each statue, incurs in the problem of expressing what is the exact property that identifies the amounts of matter that, for instance, 'are' David at different times.

d1 $\mathsf{Statue}_t x \triangleq \mathtt{AmountOfMatter} x \land x \mathsf{HasShape}_t y \land \mathtt{StatueShape} y$

A multiplicative approach helps in managing these problems. In the literature, the nature and the relations among different kinds of entities are discussed.

Four-dimensionalism (see [15]) accepts spatio-temporal-worms. A statue, say david, and the amount of matter m that constitutes david only during a part of its life, are two overlapping but different *worms*: some *temporal slices* of david are not part of m. Problems can arise when david and m coincide (share the same slices) during their whole lives. Some approaches refuse spatio-temporal coincidence. Other approaches support a *modal* distinction founded on slices spreading across *possible worlds*: david and m are different world-spatio-temporal worms because david can exist without coinciding with m (and vice versa).

In a *three dimensionalist* perspective, *multiplicative* positions (see [18]) assume that statues (generically) *existentially depend* on, more precisely they are *consituted* by, amounts of matter without overlapping with them. In particular, Fine [6] analyzes *constitution* on the basis of the notion of *qua-entity*. If an object a, the *basis*, instantiates a property P, the *gloss*, then there exists an additional entity, a-*qua*-P that is a sort of amalgam of a and P.[5] The entity a-*qua*-P, e.g. m-qua-s-shaped (m-qua-having-shape-s) exists at every time at which a instantiates P, it is uniquely determined by a and P, and it can inherit (not necessarily

[4] To avoid reference to shapes one can consider $\mathsf{StatueShaped}_t x$ where x is an object.
[5] Qua-entities seem similar to *states of affairs* as defined in [2].

all) the properties of a. Therefore, by assuming that at a given time t an amount of matter m can have only an unique shape, say s, then only a single m-qua-s-shaped entity exists at t. On the other hand, a s-shaped amount of matter $\mathsf{m}' \neq \mathsf{m}$ generates, say at t', a necessarily different qua-entity m'-qua-s-shaped. If m and m' constitute, respectively at t and t', one single statue, then still we have two qua-entities and just one statue. According to [6], statues are *mereological sums* of qua-entities (m-qua-s-shaped + m'-qua-s-shaped) aggregated by (spatio-temporal) unity criteria.[6]

Because of their relational nature, roles (and artifacts) are more controversial than material objects. While, at a given time t, amounts of matter have an unique shape, the same person, e.g. john, can be simultaneously a customer of different companies, e.g. alitalia and airfrance, i.e. both john-qua-customer-of-alitalia and john-qua-customer-of-airfrance can exist at t.[7] Moreover, differently from statues, customers always depend on the same person. This implies that different customers can share the same support, the same player.[8] Second, '*the* president of Italy' and '*the* president of Alitalia', are 'constituted-by' different persons through time and they can also share the same support at some time (somebody can be both the president of Italy and the president of Fiat). Therefore, in the case of roles, both the nature of the glosses[9] and the unity criteria are quite heterogeneous. Customers have always the same support (player) because they are discriminated on the basis of the glosses, e.g. 'being a customer of Alitalia' vs. 'being a customer of Airfrance' (see *saturated roles* in [12]) while presidents require unity criteria based on laws or social rules because spatio-temporal considerations are no relevant.[10]

The same abstract mechanism works also for *structured* objects. For instance, one can think that (a specific kind of) tables necessarily have four legs and one top even though it is possible to substitute them during their life. In this case tables can be aggregates of qua-entities where the basis is a complex object, e.g.

[6] Differently from classical temporal slices (see the definition in [15]), qua-entities persist through time when the basis instantiates the gloss during a whole interval.

[7] Here 'customer-of' is a relation defined on persons and companies. Qua-entities are then identified by a person and a property like 'being a customer of company A'. DBs often add customer codes that, however, in general, are *keys* to identify *persons* not customers. This is due to the fact that DBs do not refer to persons, they just manage cluster of attributes (e.g. Name, Date Of Birth, etc.) that do not always identify persons. Customer codes could be conceptually necessary when the same person can have different customer roles inside the same company according to, for instance, his/her rights or obligations. In this case, the way qua-entities are identified is different because there is a third argument in 'customer-of'.

[8] In this view customers coincide with single qua-entities, a limit case of mereological sum, that have the 'form' *person-qua-customer-of-A*. This explains why multiplicativist models of roles often consider only qua-entities and not general sums.

[9] Some authors claim that roles are necessary based on *anti-rigid* properties. I will not address here this topic.

[10] It is not clear to me whether unity criteria that involve *diachronic* constraints are part of the glosses.

a sum of four legs and one top, and the gloss is a structural property reducible to some spatial relations holding between the legs and the top. In this case there are two unity criteria. A synchronic one that establishes how the legs and the top must be structured, and a diachronic one that establishes the allowed substitutions of legs and tops through time.

Despite the differences previously discussed, I think that a unified view on (structured and unstructured) material objects and roles is possible. At the end, all these kinds of objects have an *intensional* dimension, to be identified, they rely on intensional rules.

2 Founding modeling primitives on a theory of levels

I consider a temporally qualified and factual primitive called *grounding*: $x \prec_t y$ stands for "x grounds y at t". Following Husserl and Fine, Correia [5] bases his theory of dependence on *grounding*, informally interpreted as "at t, the existence of x makes possible the one of y" or "y owes its existence at t to x". Grounding is factual because the usual *definition* of the *specific* existential dependence of x on y, i.e. $\Box(\mathsf{E}x \rightarrow \mathsf{E}y)$ (where $\mathsf{E}x$ stands for "x exists")[11], reduces dependence to "the necessary truth of a material conditional whose antecedent is about x only and whose consequent is about y only; and given that any such material conditional fails to express any 'real' relation between the two objects, it is hard to see how prefixing it with a necessary operator could change anything in this connection" ([5], p.58). Grounding is temporally qualified because the usual definition of the *generic* existential dependence of an object on an type P, i.e. $\Box(\mathsf{E}x \rightarrow \exists y(\mathsf{E}y \wedge Py))$, does not allow to represent on which object an object depends at a given time.

Even though I completely agree on these remarks, I consider here a notion of grounding that is stricter than the one of Correia, a notion somewhere in between pure existential dependence and constitution. Let us come back to qua-entities. Fine considers *a-qua-P* as an amalgam of a and P. From a purely existential perspective, *a-qua-P* depends on both a and P. If P is a relational property, e.g. 'being the customer of alitalia', then john-qua-customer-of-alitalia existentially depends not only on john but also on alitalia. Intuitively, my grounding aims at capturing the specific existential dependence between john and john-qua-customer-of-alitalia by excluding the one between alitalia and john-qua-customer-of-alitalia. To add intuitions. Let us suppose that 'supplier-for' is the inverse of 'customer-of', i.e. john is a customer of alitalia if and only if alitalia is a supplier for john. Ontologically, there are reasons to identify 'customer-of' and 'supplier-for'. However also in this case, john-qua-customer-of-alitalia is intuitively different from alitalia-qua-supplier-for-john because we are changing the 'perspective', we are changing the basis (and therefore the gloss). In particular, even though the first qua-entity existentially depends on alitalia, it is strictly linked to (*directed* to and *thicker* than) john. Approaches based on constitution, often advocate spatial co-location. The constituted entity is co-located with the constituent entity.

[11] This definition has been modified to answer some criticisms. See [16] and [5].

In the case of qua-entities, john-qua-customer-of-$\mathsf{alitalia}$ is intuitively co-located with john not with $\mathsf{alitalia}$. However, my grounding is defined on objects that are not necessarily in space. In addition, constitution (and supervenience [10]) often excludes relational properties from the ones that can 'generate' new (kinds of) entities. Aiming at managing roles, this constraints is too strong for my goal.

Formally, I simplify the theory in [11] by avoiding the temporal qualification of *parthood* and by discarding the primitive of *being at the same level as*.

Grounding is asymmetric, transitive, down linear (a1), and implies existence (a2), where the primitve $\mathsf{E}_t x$ stands for "the object x exists, is present, at time t", or, more neutrally, "x is temporally extended through the time t".[12] *Direct grounding* (d2) captures one single grounding step.

Parthood, $x\mathsf{P}y$ stands for "x is *part* of y", is a a purely *formal* notion on the basis of which *overlap* (O) is defined as usual [16]. More precisely, I consider a *classical extensional mereology*: parthood is reflexive, antisymmetric, transitive, implies existence, and satisfies the strong supplementation principle (a3) guaranteeing that two objects with the same parts are identical [16]. *Mereological sums*, $s\mathsf{SM}\{a_1, \ldots, a_n\}$ stands for "s is the mereological sum of $a_1, \ldots, a_n$" (d3), refer to 'multitudes' of objects without a strong ontological commitment. For instance, four legs and one top exist if and only if their mereological sum exists, but if they are disassembled no table exists.[13] Grounding is not a specific kind of parthood. Differently from (improper) parthood, grounding is irreflexive (directly from asymmetry). Differently from proper parthood, grounding does not satisfy the strong (and the weak) supplementation principle. For example, the fact that an amount of matter $\mathtt{m}$ grounds a statue does not require the statue to be grounded on additional entities disjoint from $\mathtt{m}$, i.e. $\mathtt{m}$ could be the only grounding of the statue. More strongly, I assume that grounding and parthood are incompatible: $x \prec_t y \to \neg x\mathsf{P}y$. Note however that a grounding object is not necessarily atomic, i.e. it can have proper parts.

a1	$y \prec_t x \wedge z \prec_t x \to y \prec_t z \vee y = z \vee z \prec_t y$	(*down linearity*)
a2	$x \prec_t y \to \mathsf{E}_t x \wedge \mathsf{E}_t y$	
a3	$\neg x\mathsf{P}y \to \exists z(z\mathsf{P}x \wedge \neg z\mathsf{O}y)$	(*strong supplementation*)
a4	$x \prec_t y \to (\mathtt{T}x \leftrightarrow \neg \mathtt{T}y)$	($\mathtt{T}$ is a leaf type)
a5	$\mathtt{T}_1 x \wedge \mathtt{T}_2 y \wedge x \prec_t y \to \neg \exists z u t'(\mathtt{T}_1 z \wedge \mathtt{T}_2 u \wedge u \prec_{t'} z)$	($\mathtt{T}_1, \mathtt{T}_2$ leaf types)
d2	$x \lessdot_t y \triangleq x \prec_t y \wedge \neg \exists z(x \prec_t z \wedge z \prec_t y)$	(*direct grounding*)
d3	$s\mathsf{SM}\{a_1, \ldots, a_n\} \triangleq \forall z(z\mathsf{P}s \leftrightarrow (z\mathsf{P}a_1 \vee \ldots \vee z\mathsf{P}a_n))$	(*mereological sum*)

Levels are partially captured by (a finite set of) types that are assumed to be *non-empty* and *rigid* properties formally represented by (non temporally qualified) unary predicates $\mathtt{T_i}$. Types can be extensionally organized in a taxonomy. *Leaf* types, types with no subtypes, partition the domain. According to (a4), grounding always relies on a difference in type that is expressible in the theory,

[12] I will focus here only on objects present at some time.

[13] Sums need to be carefully managed because not all the summands necessarily exist at every time at which the sum exists.

i.e. grounding does not hold between objects belonging to the same leaf type. Together with (a5), it avoids grounding loops. (a4) and (a5) are basic requirement for structuring (leaf) types in *levels* that assure also the *maximality* (with respect to parthood) of the grounds.[14]

After all these technical details, I will now introduce three grounding relations useful to organize types in levels. The formal definitions characterize the semantic of these grounding relations, but, once understood, they can be used as conceptual modeling primitives. In this sense, according to the following quote, they can be seen as an abstraction, simplification, and hiding of the previous analysis: "The theoretical notions which are required for suitable characterizations of domain conceptualizations are of a complex nature. This puts emphasis on the need for appropriate computational support for hiding as much as possible this inherent complexity from conceptual modeling practitioners." ([8], p.9).

$\mathtt{T_1}$ is (directly) *specifically grounded* on $\mathtt{T_2}$ (a6), noted $\mathtt{T_1} \rhd \mathtt{T_2}$, if every $\mathtt{T_1}$-object is (directly) grounded on a single $\mathtt{T_2}$-object during its whole life, e.g. $\mathtt{Customer} \rhd \mathtt{Person}$. It is often motivated by emergent properties. `Customer` is no more modeled as a subtype of `Person`. `Customer` is now a rigid type, i.e. a customer is necessarily a customer, with specific attributes. I think this is a quite simplifying CM technique. Furthermore, the temporal extension of a customer is included in the one of the person (a different object) that grounds him, i.e., to exist, a customer requires a grounding person while persons do not require customers. We will see how (some of) the attributes of `Person` can be inherited by `Customer` and vice versa.

$\mathtt{T_1}$ is (directly) *generically grounded* on $\mathtt{T_2}$ (a7), noted $\mathtt{T_1} \blacktriangleright \mathtt{T_2}$, if every $\mathtt{T_1}$-object is (directly) grounded on some, but not necessarily the same, $\mathtt{T_2}$-object, e.g. $\mathtt{Statue} \blacktriangleright \mathtt{AmountOfMatter}$. It is often motivated by different persistence conditions.[15] Note that the proposed framework does not commit on a specific ontological theory of persistence. One can quantify on both statues and amounts of matter without including in the domain temporal slices, qua-entities, states of affairs, events, or tropes. Indeed without being forced to, the modeler can, through axioms that links statues and amounts of matter, make explicit the underlying theory of persistence (in addition to the unity criteria).

$\mathtt{T}$ is (directly and generically) *compositionally grounded* on $\mathtt{T_1}, \ldots, \mathtt{T_m}$ if every $\mathtt{T}$-object is (directly) grounded on some, but not necessarily the same, mereological sum of $\mathtt{T_1}$-,…,$\mathtt{T_m}$-objects. It is often motivated by structural relations among $\mathtt{T_1}$-,…,$\mathtt{T_m}$-objects. I distinguish *definite* compositional grounding (a8)[16], noted

[14] In general, levels are not necessarily linear and they can be conceived as collections of objects that obey the same laws of nature, have common identity criteria or persistence conditions. These are interesting points for CM that deserve future work.

[15] Customer are not completely determined by persons, nor statues by amounts of matters. Grounding does not necessarily imply reduction, it differs from *determination* used to explain supervenience, e.g. "The mental is dependent on the physical, or the physical determines the mental, roughly in the sense that the mental nature of a thing is entirely fixed by its physical nature" ([10], p.11).

[16] In (a8) and (a9) $\neg\mathtt{T_i}(x+y)$ is a shortcut for $\exists s(s\mathsf{SM}\{x,y\} \wedge \neg\mathtt{T_i}s) \vee \neg\exists s(s\mathsf{SM}\{x,y\})$.

$\mathtt{T}\blacktriangleright\langle(n_1)\mathtt{T_1};\ldots;(n_m)\mathtt{T_m}\rangle$, e.g. $\mathtt{Table}\blacktriangleright\langle\mathtt{Surface};(4)\mathtt{Leg}\rangle$[17], i.e. when a table exists it is grounded on exactly one surface and four legs, from (at least) *indefinite* compositional grounding (a9), noted $\mathtt{T_1}\blacktriangleright(\geqslant n)\mathtt{T_2}$, e.g. $\mathtt{Organism}\blacktriangleright(\geqslant 2)\mathtt{Cell}$, i.e. organisms are grounded on at least two cells even though the exact number of grounding cells can vary in time.[18] To count the grounding objects one must rely on clear principles that identify unitary objects. For example, I would exclude $\mathtt{Statue}\blacktriangleright(\geqslant 2)\mathtt{AmountOfMatter}$ and $\mathtt{Statue}\blacktriangleright(2)\mathtt{AmountOfMatter}$. Here I just assume a mereological principle, i.e. the grounding $\mathtt{T_i}$-objects does not overlap and their sums are not of type $\mathtt{T_i}$ (see (a8) and (a9)).[19]

a6 $\mathtt{T_1}x \rightarrow \exists y(\mathtt{T_2}y \wedge \forall t(\mathsf{E}_t x \rightarrow y \otimes_t x))$ (*specific direct grounding*)

a7 $\mathtt{T_1}x \rightarrow \forall t(\mathsf{E}_t x \rightarrow \exists y(\mathtt{T_2}y \wedge y \otimes_t x))$ (*generic direct grounding*)

a8 $\mathtt{T}x \rightarrow \forall t(\mathsf{E}_t x \rightarrow \exists y_{11} \ldots y_{1n_1} \ldots y_{m1} \ldots y_{mn_m} s$
$\mathsf{E}_t y_{11} \wedge \ldots \wedge \mathsf{E}_t y_{mn_m} \wedge s\mathsf{SM}\{y_{11},\ldots,y_{mn_m}\} \wedge s \otimes_t x \wedge$
$\mathtt{T_1}y_{11} \wedge \ldots \wedge \mathtt{T_1}y_{1n_1} \wedge \neg y_{11}\mathsf{O}y_{12} \wedge \ldots \wedge \neg y_{1,n_1-1}\mathsf{O}y_{1n_1} \wedge \neg\mathtt{T_1}(y_{11}+y_{12}) \wedge \ldots$
$\ldots$
$\mathtt{T_m}y_{m1} \wedge \ldots \wedge \mathtt{T_m}y_{mn_m} \wedge \neg y_{m1}\mathsf{O}y_{m2} \wedge \ldots \wedge \neg\mathtt{T_m}(y_{m1}+y_{m2}) \wedge \ldots$

a9 $\mathtt{T_1}x \rightarrow \forall t(\mathsf{E}_t x \rightarrow \exists s(s \otimes_t x \wedge \forall z(z\mathsf{P}s \rightarrow \exists u(u\mathsf{P}z \wedge \mathtt{T_2}u)) \wedge$
$\exists y_1 \ldots y_n(\mathsf{E}_t y_1 \wedge \ldots \wedge \mathsf{E}_t y_n \wedge y_1\mathsf{P}s \wedge \ldots \wedge y_n\mathsf{P}s \wedge \mathtt{T_2}y_1 \wedge \ldots \wedge \mathtt{T_2}y_n \wedge$
$\neg y_1\mathsf{O}y_2 \wedge \ldots \wedge \neg y_{n-1}\mathsf{O}y_n \wedge \neg\mathtt{T_2}(y_1+y_2) \wedge \neg\mathtt{T_2}(y_1+y_3) \wedge \ldots)))$

Generic (or specific) grounding relations can be easily combined. For example, $\mathtt{Kitchen}\blacktriangleright\langle\mathtt{Table};\mathtt{Oven};(\geqslant 2)\mathtt{Chair}\rangle$. To mix specific and generic (compositional) grounding, one just needs to introduce more elaborate definitions. E.g., $\mathtt{Car}\rhd\!\blacktriangleright\langle\rhd\mathtt{Chassis};\blacktriangleright\mathtt{Engine};\blacktriangleright(4)\mathtt{Wheel};\blacktriangleright(\geqslant 1)\mathtt{WindscreenWiper}\rangle$ ($\rhd\!\blacktriangleright$ is heterogeneous grounding) stands for “cars specifically depend on a chassis and generically depend on an engine, four wheels, and at least one windscreen wiper”.

Methodologically, one can start from the *fundamental* types, types that are not grounded[20], and then, according to the grounding relations, progressively arrange the other (leaf) types in layers. Figure 1 depicts a simple example (with only a fundamental type, namely AmountOfMatter) that shows the weakness of the notion of level: types can be grounded on types that have a different distance from the fundamental level as in the case of Exhibition.

Inheritance. All the types involved in grounding relations are *rigid* and *disjoint* from the ones on which they are grounded. Customers, statues, and tables are

[17] I write $\mathtt{Table}\blacktriangleright\langle\mathtt{Surface};(4)\mathtt{Leg}\rangle$ instead of $\mathtt{Table}\blacktriangleright\langle(1)\mathtt{Surface};(4)\mathtt{Leg}\rangle$. This is consistent with the fact that $\mathtt{T_1}\blacktriangleright\mathtt{T_2}$ is equivalent to $\mathtt{T_1}\blacktriangleright\langle(1)\mathtt{T_2}\rangle$, i.e. generic compositional grounding is an extension of generic grounding.

[18] ‘At most’ indefinite compositional grounding, *cardinality* constraints (for example, $\mathtt{FootballTeam}\blacktriangleright(11\ldots22)\mathtt{FootballPlayer}$). Moreover, indefinite compositional grounding can also be used to introduce levels of *granularity*, even though additional constraints are necessary (see [11] for a preliminary discussion).

[19] *Specific* compositional grounding can be defined starting from the corresponding generic case by considering the ‘form’ in (a6) instead of the one in (a7).

[20] The existence of a (unique) fundamental level is debated in philosophy. However, in applicative terms, I don’t see any drawback in accepting fundamental types.

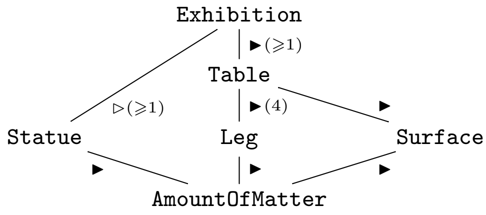

Fig. 1. Structuring object types according to grounding relations.

such during their whole life. Grounding and subtyping are separate relations, therefore the problems due to isa overloading trivially disappear. As drawback, we loose the power of the inheritance mechanism. However, Baker [3] observes that constitution (a specific grounding) provides a *unity*, it allows the constituted entity to *inherit* (to derivatively have) *some* properties from the constituting one *and vice versa.*[21] E.g. amounts of matter (persons) can inherit the *style* (*right to vote* for student representatives) from the statues (students) they ground.

On the basis of these observations, following [1], the inheritance of attributes of grounded types must be controlled. By default, $T_1 \rhd T_2$ or $T_1 \blacktriangleright T_2$ implies that all the attributes of T_2 are inherited by T_1. $T_1[A_1^1, ..., A_n^1] \rhd T_2[A_1^2, ..., A_m^2]$ means that only the T_2 attributes $A_1^2, ..., A_m^2$ are exported to T_1 while the T_1 attributes $A_1^1, ..., A_n^1$ are exported to T_2. Similarly in the case of generic grounding. Statue[Style] ▶ AmountOfMatter means that Statue inherits all the attributes of AmountOfMatter, while the last type inherits only the attribute Style from Statue. In this way, attribute hiding can be trivially modeled. Attribute overriding can be approached by systematically override the attributes of the grounding type or by localizing all the attributes as in [1]. The case of compositional dependence is interesting. Some attributes of the grounded object can be obtained from a 'composition' of the attributes of the grounds. For example, the weight of tables is the sum of the weights of the grounding legs and surfaces. If necessary these rules can be explicitly added as constraints. Alternatively, one can add dependences among the values of attributes.

Grounding and subtyping. It is trivial to prove that if $T_1 \Rightarrow T_2$[22] and $T_2 \rhd T_3$ then $T_1 \rhd T_3$. Vice versa, from $T_1 \Rightarrow T_2$ and $T_1 \rhd T_3$, $T_2 \rhd T_3$ does not follow. Moreover, from $T_1 \rhd T_2$ and $T_2 \Rightarrow T_3$ it follows that $T_1 \rhd T_3$ but one looses the information about the specific subtype on which T_1 is grounded. A 'parsimonious approach' considers only maximally informative grounding relations $T_1 \rhd T_2$: T_1 is *maximal* with respect to subtyping, while T_2 is *minimal*. This criterion (together with the fact that only direct grounding relations are considered) allows to clarify the nature of abstract types like MaterialObject. Let us assume Leg ⇒ MaterialObject, Surface ⇒ MaterialObject, and Table ⇒ MaterialObject

[21] However, high-level properties are not always reducible to properties of substrata.

[22] ⇒ represents the subtyping relation. The following results hold also for generic dependence. Here I do not consider the composition of grounding relations.

and compare the model that considers all the grounding relations in Figure 1 with the one with only `MaterialObject`▶`AmountOfMatter`. Given the same taxonomical information, only the first model makes explicit that `MaterialObject` is an abstract and multi-level type.

Acknowledgments. I would like to tank Emanuele Bottazzi, Nicola Guarino, Laure Vieu, and the reviewers for the helpful comments and discussions. This work has been carried out in the framework of the EuJoint project (Marie Curie IRSES Exchange Project n.247503) and the ICT4Law project (Converging Technologies project financed by Regione Piemonte).

References

1. Al-Jadir, L., Michel, L.: If we refuse the inheritance ... In: Bench-Capon, T., Soda, G., Tjoa, M. (eds.) Proceedings of DEXA'99. pp. 569–572. Springer Verlag (1999)
2. Armstrong, D.M.: A World of States of Affairs. Cambridge University Press (1997)
3. Baker, L.R.: The Metaphysics of Everyday Life. Cambridege Univerity Press (2007)
4. Borgo, S., Vieu, L.: Artefacts in formal ontology. In: Meijers, A. (ed.) Handbook of Philosophy of Technology and Engineering Sciences, pp. 273–308. Elsevier (2009)
5. Correia, F.: Existential Dependence and Cognate Notions. Ph.D. thesis, University of Geneva (2002)
6. Fine, K.: Acts, events and things. In: Sixth International Wittgenstein Symposium, Kirchberg-Wechsel (Austria). pp. 97–105 (1982)
7. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Ph.D. thesis, University of Twente (2005)
8. Guizzardi, G., Halpin, T.: Ontological foundations for conceptual modeling. Applied Ontology 3(1-2), 1–12 (2008)
9. Gupta, A.: The Logic of Common Nouns. Phd thesis, Yale University (1980)
10. Kim, J.: Mind in a Physical World. MIT Press (2000)
11. Masolo, C.: Understanding ontological levels. In: Lin, F., Sattler, U. (eds.) Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010). pp. 258–268. AAAI Press (2010)
12. Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., Guarino, N.: Social roles and their descriptions. In: Dubois, D., Welty, C., Williams, M.A. (eds.) Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 04). pp. 267–277 (2004)
13. Mizoguchi, R., Sunagawa, E., Kozaki, K., Kitamura, Y.: A model of roles within an ontology development tool: Hozo. Applied Ontology 2(2), 159–179 (2007)
14. Rea, M. (ed.): Material Constitution. Rowman and Littlefield Publishers (1996)
15. Sider, T.: Four-Dimensionalism. Clarendon Press, Oxford (2001)
16. Simons, P.: Parts: a Study in Ontology. Clarendon Press, Oxford (1987)
17. Steimann, F.: On the representation of roles in object-oriented and conceptual modelling. Data and Knowledge Engineering 35, 83–106 (2000)
18. Vieu, L., Borgo, S., Masolo, C.: Artefacts and roles: Modelling strategies in a multiplicative ontology. In: Eschenbach, C., Gruninger, M. (eds.) Proceedings of Fith International Conference on Formal Ontology and Information Systems (FOIS 2008). pp. 121–134. IOS Press (2008)