

A taxonomy of types of granularity

C. Maria Keet

KRDB Research Centre, Faculty of Computer Science,
Free University of Bozen-Bolzano, Italy
Email: keet@inf.unibz.it

Abstract—Multiple different understandings and uses exist of what granularity is and how to implement it, where the former influences success of the latter with regards to storing granular data and using granularity for reasoning over the data or information. We propose a taxonomy of types of granularity and discuss for each leaf type how the entities or instances relate within its granular level. Such unambiguous distinctions can guide a conceptual modeler to better distinguish between the types of granularity and the software developer to improve on implementations of granularity.

I. INTRODUCTION

Granularity deals with articulating something (hierarchically) according to certain criteria, the granular perspective, where a lower level within a perspective contains knowledge (i.e. entities, concepts, relations, constraints) or data (measurements, laboratory experiments etc.) that is more detailed than the adjacent higher level. Conversely, a higher level ‘abstracts away’ – simplifies or makes indistinguishable – finer-grained details. A granular level is also called grain size and contains one or more entities and/or instances. Ideas about what granularity comprises can differ between research disciplines that tend to emphasize one aspect or the other. It combines efforts from philosophy, AI, machine learning, database theory and data mining, (applied) mathematics with fuzzy logic and rough sets, among others [15], for example [14] [10] [16] [17] [9]. Several interpretations of granularity capture subtle, but essential, differences in interpretation, representation, and/or emphasis. While for e.g. data clustering a certain extent of fuzziness or roughness can be accepted for *allocating* entities or instances to their appropriate level, if one wants to reason over granulated data and information (e.g. with transitivity of the parthood relation) and *retrieve* information, clearer distinctions and semantics have to be defined to achieve correct inferencing results, in part because it emphasizes a *qualitative* component of granularity, albeit not ignoring the quantitative aspects. We have structured the main types of granularity in a taxonomy, introduced in section II, followed by their consequences (section III). In section IV we describe how the entities or instances within a level of granularity relate, and their differences resulting from being of a different type of granularity. Last, we make some final remarks and aspects of further investigation (section V).

II. A TAXONOMY OF TYPES OF GRANULARITY

One can identify the main differences in types of granularity based on:

- 1) Arbitrary scale versus non-scale-dependent granularity, which includes partitioning one entity (/instance) according to one or more criteria versus applying granularity to multiple entities (/instances) and simple (arithmetic) aggregation versus more complex folding operations;
- 2) How levels (and its contents) in a perspective relate to each other;
- 3) The perception and (mathematical) representation, such as based on set theory versus mereology.

These differences do not imply one cannot switch from one to the other, represent one way into another, or let them work together orthogonally. Humans seamlessly shift granular perspectives and alternately emphasize the criterion used for granularity and the partitioning within a level itself, or taking an entity versus instance-inspired approach. But how can we teach a computer program to do so, when and how can it switch from one perspective to the other, from one level to another, and from property-based to data-based? A first step is to disambiguate types of granularity. A taxonomy of types of granularity is constructed for development of a Theory of Granularity (TOG). This emphasizes that there is not one granularity, but several types that with additional constraints extend the core TOG as root. Fig. 1 shows the basic taxonomy, which may require refinements or collapsing into fewer types to implement this, but these are software design considerations. In this section, we briefly describe the types, which are elaborated on in section IV where the relations between entities within a level are described and formalized.

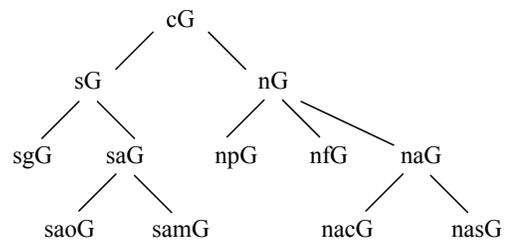


Fig. 1. A basic taxonomy of types of granularity.

- **cG**: core TOG, consisting of the basic elements common to all types of granularity, such as the domain demarcation, a granular perspective (e.g. time, human structural anatomy), levels within a perspective, relations between the levels, and constraints like that a granular perspective must have at least two levels.

- **nG**: non-scale-dependent granularity, where other types of entities reside in each finer-grained level; it consists of the **cG**, additional constraints and types of relations between levels (see below).
- **npG**: levels of non-scale dependent granularity are ordered through one type of (primitive) relation in a perspective. E.g.: (structural-)*partOf*, (spatially-)*containedIn*.
- **nfG**: folding, whereby levels of non-scale dependent granularity are ordered by simultaneous folding ≥ 2 different types of entities, such as folding events and states, and consequently possibly folding (primitive) relations, upon shifting grain size. E.g.: ‘black boxes’ approach in biology such as the *SecondMessengerSystem*, the Abstraction Hierarchy.
- **naG**: non-scale-dependency with some form of ‘aggregation’ (specified in the types it subsumes).
- **nasG**: non-scale-dependency using aggregation of the same collection of instances of one type that subsequently can be partitioned in various ways at lower levels of detail using semantic criteria. The entity at a lower level is a subtype of the type at the coarser-grained level. E.g. a collection of phone points and finer-grained land-line mobile phone points.
- **nacG**: non-scale dependency using aggregation attributed to the notion of an entity labeled with a collective noun – i.e. with an existing semantics –, the instances of the aggregate are different from instances of its members, and a change in its members does not affect the meaning of the whole. E.g. *Population* with *Organisms of type x*, or *Team* as aggregate of its *Players*.
- **sG**: scale-dependent granularity where the contents is structured according to a (more or less obvious) arbitrary scale; consists of the **cG** and additional constraints. E.g. calendar hierarchy, rounding off of altitude lines on a cartographic map.
- **sgG**: scale dependency with relation to grain size, or the resolution, scale-based zooming. E.g. *CellWall* as line, as lipid bi-layer, and as three-dimensional structure, or two objects (e.g. wallpaper and the wall) touching each other. This might need further analysis and refinement.
- **saG**: scale dependency with some form of ‘aggregation’ (specified in the ones it subsumes) and its immediate parts are of one type.
- **samG**: scale dependency and using aggregation of the same collection of instances of the same top type or Urelement that subsequently can be partitioned in various ways at lower levels of detail using a mathematical function. E.g. *Second*, *Minute*, and *Hour*, where 60 seconds go in a minute.
- **saoG**: the carving up of the same entity at each level that neither is a collective noun nor an ad hoc collection. E.g. the earth with its isobars, where the isotherms are in steps of 10 degrees, 5 degrees, 1 degree detail.

Other categorizations of types of granularity are possible, but these are less advantageous. One could have chosen to make the main criterion some type of aggregation versus ‘granularity by other means’ instead of the (non) scale dependency because one has both **saG** and **naG** each with their subtypes. However, using aggregation emphasizes the internal structure of a level, how entities and instances relate semantically or is implementation-driven, but does not take into account the properties how to make the distinction between types because having a remainder group of types of granularity does not capture the semantics adequately. In addition, using aggregation as distinguishing criterion *implicitly* makes a distinction between set theory and set theory-or-mereology, but this is a representational issue (see also next paragraph). Last, aggregation is underspecified, both with respect to its ontological nature and variants in implementations (discussed in the next paragraph). In contrast, the proposed taxonomy takes a purely *semantic*, ontological, approach, thereby separating (formal) representation and implementation from the meaning.

III. EFFECTS OF TYPES OF GRANULARITY

The difference between scale and non-scale dependency and their formal representations roughly fits with Sowa’s [13] epistemic and intentional granularity. Sowa bases his three types of granularity on Peirce’s three categories of Firstness, Secondness and Thirdness.

1. Firstness maps to actual granularities with axioms for discrete, continuous or lumpy aggregates [13] concerning the entities that populate a level.
2. Secondness for granularity uses epistemic logics involving measurements [13] and corresponds to the scale-dependent granularity.
3. The Thirdness for granularity, intentional, requires a three-place predicate relating “an agent a to an entity x for a reason r ” [13].

Depending on how one uses granularity in a subject domain, devising *levels* does not require asking oneself questions if entity x has at least one atom as part, if there is an infinite regress of parts that is cut at the lowest level defined, or if the entity is lumpy (point 1 above). More precisely, in mereology, an *Atom* is an entity that has no proper parts (1). The allocating of entities to a given level does make use of the aggregates and entities. There are three kinds of aggregates (with “ \leq ” as part-of and “ $<$ ” as proper-part-of): *Discrete*: everything has at least one atom as part (2); thus, that things can be subdivided up to the point where nothing is left but atoms. *Continuous*: everything has at least one proper part (3), which permits indefinite subdivision, implying that there are no atoms, and *Lumpy*: some things are atoms, some are continuous (4). [13]. Basing granularity on mereology does not require the ‘ultimate part’ or Urelement needed for set theory-based granularity [2].

$$Atom(x) \triangleq \neg \exists y(y < x) \quad (1)$$

$$\forall x \exists y (Atom(y) \wedge y \leq x) \quad (2)$$

$$\forall x \exists y (y < x) \quad (3)$$

$$\exists x \text{Atom}(x) \wedge \exists y \forall z (z \leq y) \rightarrow \exists w (w < z) \quad (4)$$

Ease, difficulty, or even impossibility, to identify an Urelement is illustrated in the following example.

Example 1. Taking calendar entities and set-theory based granularity, one may argue entities like *Week*, *Month*, *Quarter*, and *Year* can be built from a chosen Urelement *Day* and can be represented by distinct sets of days. However, taking isotherms, then what can – should – be chosen as Urelement? For instance, using *Degree* as smallest element and building up coarser-grained isotherms: with a set $\{15, 16, 17, 18, 19, 20, 21, 22, 23, 24\}$ as the extension of *Isotherm20* – where *Isotherm20* is a subtype of *Isotherm* that has other subtypes like *Isotherm30* etc. – there are some problems. The extension of a set is not equal to the entity; even more so, the numbers are not degrees but integers.

With biology, identifying or choosing a smallest element is more challenging. Say, a general practitioner does not think smaller than tissue, i.e the Urelement – Atom – for the lowest level is *Tissue*, which means that all higher-level structures are composed of tissue *only*: we *know* this is biologically incorrect and thereby not a good representation of nature. In addition, if one takes, for instance, the lowest level *BiologicalMacromolecule* of the FMA [11] [3] and deem that the coarser-grained level *Body* is the set of all its macromolecules – omitting the other molecules without which a human body cannot survive, H₂O to name just one – then a body would change identity every instance a biological macromolecule is synthesized/metabolized, which happens continuously, resulting in the situation that a body has no enduring identity but be in flux (this view is not entirely uncommon [4], but topics like the four-dimensionalism of perdurantists is outside the current scope). Likewise, entomologists study the same ant colony over time, even though its workers are born and have died.

Regardless if the model is logically valid and corresponding knowledge base in a legal state, basing inferencing on represented knowledge that is not adequately grounded in the reality it aims to represent can lead to undesirable outcomes for patients, ecosystems and the like. \diamond

Both ways of representing granularity, through *isA* with set theory and mereological *partOf*, are from a logical viewpoint interchangeable, but not from an ontological viewpoint as the intended meaning captured in a formalization is distinct. This difference has been recognized earlier [12] and are not considered to be *competing* interpretations of granularity, but *both* considered as distinct, valid ways of understanding granularity. One does not have to force one type of granularity in the straightjacket of the other; doing so anyway always will deprive another type of granularity from representing nature as accurate as possible. Capturing granularity in one’s preferred version of logic is restrictive, unless clear semantics of the intended meaning is provided with it, i.e. the formalism is *ontological* in nature instead of only a logical theory [6].

Reason *r* in Sowa’s point 3 above is particularly interesting for granular perspectives in non-scale dependent granularity, although not necessarily modeled as a triadic predicate; separating and reusing the reason benefits scaling up the granularity framework. However, the fundamental difference of the latter with arbitrary scales may be clear. The differences in types of granularity have, at the meta-level, a major effect on the type of relation between granular levels. This because scale-dependent levels are identified and ordered according to a combination of a property and an arbitrary scale whereas non-scale-dependent levels are ordered according to a combination of properties where level identification is much less straightforward. Also, the latter requires closer ontological investigation into the possible types of relation that link adjacent levels.

IV. STRUCTURES OF THE CONTENTS FOR LEAF TYPES OF GRANULARITY

Another consequence of different types of granularity can be observed for the domain entities contained in a level. As mentioned in the previous section, in scale-dependent levels it can be that the same entity is partitioned with increasingly smaller cells, whereas with entities sorted according to a non-scale-dependent perspective, different types of entities taken from an ontology or knowledge base reside in each level. This distinction influences the structure of the contents, including possible completeness and disjointness requirements of the subject matter. A grid is automatically disjoint and, depending on the level and implementation decisions, complete. With non-scale-dependent granularity, it may be neither disjoint nor complete. It also affects the ‘loading’ and subsequent browsing, querying, and inferencing over contents of levels.

Considering **cG** and possible data sources to load/apply a granularity framework, we can state several general conditions that the structure within a level must satisfy.

- i. The contents of a level can be either entities (taken from an ontology or knowledgebase) or instances (taken from e.g. a knowledgebase or database), but not both.
- ii. The entities/instances in a particular level have at least one property (value) in common.
- iii. The entities/instances are disjoint, but
- iv. They are not necessarily exhaustive: this may not be possible due to our gaps in knowledge of the natural world. Within a closed world assumption, they are disjoint exhaustive.
- v. Provided an entity/instance is not an orphan and the subject domain is covered fully with granular perspectives, it must reside in at least one granular level.
- vi. An entity/instance never can reside in more than one granular level within the same granular perspective.
- vii. The entity/instance in a particular granular level may reside also in ≥ 1 other levels, provided that each level the entity/instance resides in is contained in a distinct granular perspective.

Because not every subject domain is granulated in the same way, the structure of the data and relations between the entities/instances in the levels differ too, independent of the actual

data source. It uses several generic relations and operations for retrieval and assignments, which are introduced here first. Beforehand, two general conventions will be given.

- A. The data manipulation is applicable to both entities (universals) and particulars, for we do not specify a single data source granularity is applied to (for instance, an ontology, knowledgebase, database, or corpus). In the formalization, we adhere to the universals defined and characterized in DOLCE [7], like *PT* for particular and *PRO* for process, to type the instances – as DOLCE is a foundational ontology for particulars. One can also read the formalizations presented here not applicable to instances but to entities, where the entities are of the type indicated with the DOLCE category. Thus, a particular amount of matter x , such as *my blood*, is denoted as $M(x)$ if the data source is a database or the ABox of a Description Logics knowledgebase, whereas if the data source is an ontology of universals or TBox, then the x in $M(x)$ stands for its universal *Blood*.
- B. An operation (function) in FOL has the form $f(A_1, \dots, A_n) : R$, with f denoting the function, A_1, \dots, A_n the arguments, and R the type of the result. Summarising [1], this corresponds to an $(1 + n + 1)$ -ary predicate f_{A_1, \dots, A_n} , with as first object the object of invocation, n arguments and the last one represent the result. This predicate must satisfy:

$$\forall x, a_1, \dots, a_n, r (f(x, a_1, \dots, a_n, r) \rightarrow \bigwedge_{i=1}^n A_i(a_i)) \quad (5)$$

$$\forall x, a_1, \dots, a_n, r, r' (f(x, a_1, \dots, a_n, r) \wedge f(x, a_1, \dots, a_n, r') \rightarrow r = r') \quad (6)$$

$$\forall x, a_1, \dots, a_n, r (PT(x) \wedge f(x, a_1, \dots, a_n, r) \rightarrow R(r)) \quad (7)$$

where (5) ensures correct typing of the arguments, (6) says that the object with given arguments determines in a unique way the return value, and (7) ensures the correct type of result. This said, we use a simplified way of representing this to avoid repetitive axioms that are essentially the same for the functions introduced in this chapter, where a function like $grain(x)$ is constrained as above.

Next, we introduce some essential definitions of components of granularity: the framework/domain D , granular perspective GP , and the hierarchically ordered levels GL it contains.

$$\forall x (D^{fw}(x) \triangleq \exists! y, z (DF(x, y) \wedge CN(x) \wedge OD(x, z) \wedge D^{sd}(z))) \quad (8)$$

$$\forall x (D^{sd}(x) \triangleq \exists! y ((CN(x) \vee PT(x) \vee U(x)) \wedge DF(x, y))) \quad (9)$$

$$\forall x (GP(x) \triangleq \exists! y, z (CN(x) \wedge DF(x, y) \wedge RC(x, z) \wedge C(z))) \quad (10)$$

$$\forall x (GL(x) \triangleq \exists! v, w, y, z (DF(x, y) \wedge C(z) \wedge RC(w, z) \wedge GP(w) \wedge R(v) \wedge hasValue(z, v) \wedge RE(x, w))) \quad (11)$$

Definition *DF*, concept *CN*, one-sided dependent *OD*, particular *PT*, and region *R* are taken from the DOLCE (formal) foundational ontology [7], U stands for universal, criterion C – properties, qualities, attributes – is related to a *GP* via the relation *RC*, and *RE* is the relation between the framework elements (a particular level is contained in (part of) a perspective), and the *hasValue* relation denotes that a quality x has a quale (value) in the region y (as a shorthand for readability). Bearing this in mind, (12) says there is a relation between an entity x and the level it is of (resides in), with (13, 14) analogous to Kumar et al’s *gran* function [5], returning level y where entity, or instance, x resides. For (15, 16), where $x \in U$ or $x \in PT$ with U as the set of universals and PT the set of instances (particulars), and y a granular level of a perspective in a subject domain, the function $assignGrainLevel(x, y)$ assigns a level to an entity or instance.

$$\forall x, y (isOfLevel(x, y) \triangleq (PT(x) \wedge GL(y) \wedge grain(x) = y)) \quad (12)$$

$$\forall x \exists y ((grain(x) = y)) \quad (13)$$

$$\forall x (PT(x) \rightarrow \exists y (grain(x) = y \wedge GL(y))) \quad (14)$$

$$\forall x \exists y (assignGrainLevel(x, y)) \quad (15)$$

$$\exists x, y (assignGrainLevel(x, y) \rightarrow (GL(y) \wedge PT(x) \wedge isOfLevel(x, y))) \quad (16)$$

The respective structures for the leaf types of granularity are as follows.

saoG: all instances in the ordered set belonging to a particular level are all instances of the same type. The amount of whole instances is not necessarily determined by the size of the entity that is partitioned. For instance, when one partitions the area of a lake by overlaying a coarser or finer-grained grid and include rules alike ‘if the cell is filled with $> 50\%$, then include cell, else discard the area of the cell’. The instances are automatically disjoint, i.e. when $a, b \in X$ then $a \cap b = \emptyset$, because the partitioning is grid-wise (the cells of a grid are not necessarily square-shaped). Further, although the scale-dependent granularity has a mathematical relation between entities of different levels (17), within the same level the instances in the set is the set-extension of its corresponding universal (18) such as a set of plots of km^2 where the amount of plots depends on the entity/instance that is partitioned (like a large or small lake) and on the decisions to include or discard ‘partial’ plots.

$$\forall x (RL(x_i, x_j) = partOf(x_i, x_j) \leftrightarrow RL(x_j, x_i) = hasPart(x_j, x_i)) \wedge \neg(x_i = x_j) \quad (17)$$

$$\forall x \exists y, z ((PT(x) \wedge \{x_1, \dots, x_n\} \leftrightarrow Set(z) \wedge GL(y) \wedge isOfLevel(z, y)) \quad (18)$$

samG: all instances in the ordered set belonging to a particular level are instances of the same type and at that level they are whole instances. Also, there is an exact, known,

number of instances that can be in that level. In addition, the entities and instances at the higher levels are ultimately composed of the chosen Urelement at the lowest granular level. In contradistinction with **saog**, the set is grouped into particular amounts, like {Hour 1, ..., Hour 24} at the *Hour*-level gp_1gl_i , which are ultimately built up from the same Urelement *Second* at level gp_1gl_{i+2} etc. (20).

$$\begin{aligned} \forall x \exists y, z, w (Urelement(x) \wedge GL(y) \wedge \\ (\{x_1, \dots, x_i\} \rightarrow PT(z)) \wedge (\{x_1, \dots, x_n\} \rightarrow PT(w)) \\ \wedge isOfLevel(x, y_j) \wedge isofLevel(z, y_i) \wedge \\ isOfLevel(w, y_1) \wedge y_j \prec y_i \prec y_1) \end{aligned} \quad (19)$$

$$\begin{aligned} \forall x, y (RL(x_i, x_j) = (partOf(x_i, x_j) \wedge funct(y))) \\ \leftrightarrow (RL(x_j, x_i) = (hasPart(x_j, x_i) \wedge \\ funct^{-1}(y))) \wedge x_i \prec x_j \end{aligned} \quad (20)$$

sgG: this involves a ‘zooming in’ and ‘zooming out’ factor, where at a coarser-grained level, for instance, the wall and wallpaper touch each other, but at a finer resolution, greater magnification, there is wall-glue-wallpaper, and again in smaller detail looking at the molecules in the paper, glue, and wall. The zooming factor is like a grain size when relating levels of granularity, where *within one* level one can distinguish instances of e.g. $\geq 1\text{mm}$, but instances $< 1\text{mm}$, metaphorically, fall through the sieve and are indistinguishable from each other, but are distinguishable at lower levels of granularity. This is different from a partonomic relation where, say, the *Cell*-level is *conceptually* always part of at least one higher-level structure such as *Organ*, but is not necessarily the case that all cells are always *physically* smaller than all higher-level entities and instances, with *ChickenEgg* at the *Cell*-level as an obvious example. Here differences in physical size is *the* criterion. *sizeOf* is defined as a function that returns a value in e.g. length, square or cubic size (21), where *PR* is the Physical Region as given in DOLCE. Thus, the instances recorded at some level w_j in (22), are physically smaller than the instances at a higher level (w_i) but larger than those residing in a lower level (w_k) and thereby are related to each other by the relation that they fall within the same physical size range.

$$\forall x \exists y (sizeOf(x) \rightarrow (PT(x) \wedge PR(y) \wedge sizeOf(x) = y)) \quad (21)$$

$$\begin{aligned} \forall x, y, z \exists w (PT(x) \wedge PT(y) \wedge PT(z) \wedge \\ (sizeOf(x) < sizeOf(y) < sizeOf(z)) \\ \wedge isOfLevel(z, w_i) \wedge GL(w) \wedge \\ w_k \prec w_j \prec w_i) \end{aligned} \quad (22)$$

npG: the entities in a level are of a different type, but all are of the same kind, such as all being non-agentive physical objects (*NAPO*) or *PRO* etc. For instance, at the *Cell*-level, there are many *types* of cells, but they are all of the kind of *NAPO* structural component (*HemalCell*, *Leukocyte*, ...), an *OrganicMolecule*-level of its functions (*HormoneExcretor*, *InsulinExcretor*, ...), and so forth. Or, take a *ProteinUnitStructure*-level with parts such as α -helices and β -sheets. Thus, the entities residing in a level are structured in

a tree-shaped hierarchy, like a taxonomy or partonomy, where the direct subtypes/parts are in a lower level of granularity than its supertype. In the level, without further specification, the entities form an unordered set. However, it may be that the contents have some other additional structure within the level orthogonally positioned alike a **nasG**, or another **npG** structure; for instance, a *Cell*-level in a human structural anatomy perspective where the taxonomy of human cell types is preserved in the granular level. Alternatively, one can re-group the unordered set such that it takes into account the tree structure, where each subset of entities correspond to a different branch; this can be dealt with trivially if implemented with a reasoner. Either way, the entities are disjoint thanks to the underlying tree structure.

nfg: the entities in a level can be of different kinds, such as folding non-agentive physical objects with their processes and states, combining types of entities into *one* entity residing in an adjacent higher level ((23), where *ED* stands for *endurant* and *PD* *perdurant*). It is not the case that the entities contained in the lower granular level is an (un)ordered set, but the entities (or its instances) are always related, denoted with the relation *Rel*, to at least one other entity (instance) within that level (24); hence ‘folding’ of entities occurs when going from a finer-grained level to a coarser-grained level, which can involve folding different types of relations too.

$$\begin{aligned} \exists >^1 x, y \exists z (isOfLevel(x, y_j) \wedge isOfLevel(z, y_i) \wedge \\ (ED(x) \vee PD(x)) \wedge GL(y) \wedge \\ PT(z) \wedge properPartOf(x, z) \wedge \\ (\{x_1, \dots, x_n\} \rightarrow z) \wedge y_j \prec y_i) \end{aligned} \quad (23)$$

$$\forall >^1 x \exists y (Rel(x_i, x_j) \wedge isOfLevel(x, y) \wedge GL(y) \wedge (ED(x) \vee PD(x))) \quad (24)$$

nacG: like **samG**, all instances belonging to a particular level are all of the same type and at that level they are whole instances. It is not necessarily the case that the amount of instances in a particular level is known and can be computed. For instance, *SportsTeam* does have a predefined amount of instances of *Player* per team, but sales department members of a company do not have to have always the same amount of members. The instances of such populations, generally labeled with a collective noun, change over time but the entity and its meaning endures. Thus, looking at the structure of the data, it is at least an unordered set though can be an ordered set of instances, where the instances populating the set can vary over time, although the entity, labeled with a collective noun, keeps its identity. It may be possible, to have not an (un)ordered set but a taxonomy or other additional aggregation within the level alike a **nasG** or **npG** structure, such as an employee hierarchy (junior and senior sales person, trainee, manager, etc), or aggregated by the organizational unit (teamA1, teamA2, etc).

$$\begin{aligned} \forall x \exists y, z (ED(x) \wedge GL(y) \wedge (\{x_1, \dots, x_n\} \rightarrow \\ ED(z)) \wedge isOfLevel(x, y_j) \wedge \\ isOfLevel(z, y_i) \wedge y_j \prec y_i) \end{aligned} \quad (25)$$

nasG: the structure of the data is alike **samG**, but if one combines the subsets at each level, then the amount of unique instances residing in *each* level is always the same amount as there are instances of the chosen Urelement. For instance, at granular level gp_1gl_2 in some perspective gp_1 there are 100 phone points, then so are there 100 phone points at the top-level gp_1gl_1 , where $gp_1gl_2 \subseteq gp_1gl_1$, although in gp_1gl_2 the 100 phone points may be divided into three subsets *LandLine*, *Mobile*, *PhoneOverIP* each with, say, 2, 35, 63 elements of the set respectively, hence $Mobile \subset PhonePoints$ and there might be a gp_1gl_3 with *Vodafone* and *O2* partitioning *Mobile* phone points, that each have 20 and 15 elements in the set respectively (adding up to the 35 elements for *Mobile* of the higher level gp_1gl_2). Note that this assumes all *Mobile* instances are categorized as either *Vodafone* or *O2*, but it is possible in another perspective or level that there is a ‘rest group’ like *TheOtherProtists* in the tree of life [8]. Thus, at each level there are subsets with instances as elements of the set, which, depending on the semantic criterion of partitioning, are disjoint.

$$\begin{aligned}
& \forall x \exists y, z, w (Urelement(x) \wedge GL(y) \wedge \\
& (\{x_1, \dots, x_i\} \rightarrow PT(z)) \wedge \\
& (\{x_1, \dots, x_i\} \rightarrow PT(w)) \wedge \\
& isA(z, w) \wedge isOfLevel(x, y_j) \wedge \\
& isOfLevel(z, y_i) \wedge \dots \wedge isOfLevel(w, y_1) \wedge \\
& y_j \prec y_i \prec \dots \prec y_1)
\end{aligned} \tag{26}$$

One may opt for the design decision to demand from the chosen criterion that the sets never overlap, or for the ‘just in case’ scenario, create two subtypes of **nasG** where one subtype of granularity does allow overlapping sets and with the other subtype this is disallowed (a discussion about the ontological aspects of (dis)allowing overlapping sets is outside the current scope).

Some types of relation between the entities or instances within a level described above can be combined, because one does not have to take into account that some are partitioned according to arbitrary scale and others are not: this (non-)arbitrary scale division is relevant for the relations between levels, but do not always act out on the relation between entities/instances contained within a level. **nasG**, **nacG**, **npG**, and **sgG** may be unordered sets, **samG** and **saoG** may be ordered sets, and **nfG**, **npG**, **sgG**, and **nacG** can have a more complex additional orthogonal structure of the data inside the level that itself may be subject to a granular structure.

V. CONCLUDING REMARKS

We have proposed a domain and implementation-independent taxonomy of types of granularity and discussed for each of the seven leaf types how the entities or instances relate within a granular level. Such unambiguous distinctions can guide a conceptual modeler to better distinguish between the types of granularity and the software developer to improve on implementations of granularity, in particular when used for reasoning over the data or information. For instance, one

can discover implied relations between entities/instances by positioning orthogonally a taxonomy and a partonomy, and make valid inferences w.r.t. (spatial) inclusion of ecological and/or GIS data. We are currently working on an ontology-inspired framework for granularity, the effects of the types on granularity for the various operations of data manipulation and reasoning, and we are conducting experiments on implementability with the subject domains of infectious diseases and nuclear hormone receptors.

ACKNOWLEDGMENT

The author would like to thank Alessandro Artale for proofreading the draft, and this research has benefited from a discussion with Enrico Franconi.

REFERENCES

- [1] Berardi, D., Calvanese, D., De Giacomo, G. Reasoning on UML class diagrams. *Artificial Intelligence*, 2005, 168(1-2):70-118.
- [2] Bittner, T., Smith, B. A Theory of Granular Partitions. In: *Foundations of Geographic Information Science*, Duckham, M., Goodchild, MF, Worboys, MF (eds.), London: Taylor & Francis Books, 2003, pp117-151.
- [3] Foundational Model of Anatomy (FMA). 2003. <http://fme.biostr.washington.edu:8089/FME/index.html>.
- [4] Hawley, K. Temporal Parts. *The Stanford Encyclopedia of Philosophy (Winter 2004 Edition)*. Zalta, E.N. (ed.). <http://plato.stanford.edu/archives/win2004/entries/temporal-parts/>.
- [5] Kumar, A., Smith, B., Novotny, D.D. Biomedical Informatics and Granularity. *Comparative and Functional Genomics*, 2005, 5(6-7): 501-508.
- [6] Laboratory of Applied Ontology. *Formal Ontology for Knowledge Representation and Natural Language Processing*. Doctorate Course, ICT School, University of Trento, Italy. <http://www.loa-cnr.it/PhD.html>. Date accessed: 30-6-2005.
- [7] Masolo, C., Borgo, S., Gangemi, A., Guarino, N. and Oltramari, A. *Ontology Library*. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003). <http://wonderweb.semanticweb.org>. 2003.
- [8] Patterson, D.J. The other protists. *The Tree of Life Web Project*. 2000. http://tolweb.org/tree?group=The_other_protists&contgroup=Eukaryotes.
- [9] Peters, J.F., Skowron, A., Ramanna, S., Synak, P. Rough sets and information granulation. In: T.B. Bilgic, D. Baets, and O. Kaynak (eds.), *Proceedings of 10th International Fuzzy Systems Association World Congress*, Lecture Notes in Artificial Intelligence 2715, Springer-Verlag, pp370-377. 2002
- [10] Reformat, M., Pedrycz, W., Pizzi, N. Building a software experience factory using granular-based models. *Fuzzy Sets and Systems*, 2004, 145:111-139.
- [11] Rosse, C., Mejino, J.L.V.: A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics*, 2003, 36: 478-500.
- [12] Salthe, S.N. Summary of the Principles of Hierarchy Theory. http://www.nbi.dk/~natphil/salthe/hierarchy_th.html (accessed: 10-10-2005). November 2001. Based on: Salthe, S.N. *Evolving hierarchical systems: their structure and representation*. Columbia University Press, 1985.
- [13] Sowa, J.F. *Knowledge representation: logical, philosophical, and computational foundations*. China Machine Press. 2000.
- [14] Yao, Y.Y. A partition model of granular computing. *LNCSTransactions on Rough Sets*, 2004, 1:232-253.
- [15] Yao, Y.Y. Perspectives of Granular Computing. *IEEE Conference on Granular Computing 2005*, Vol. 1, pp85-90.
- [16] Zadeh, L.A. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 1997, 90(2):111-127.
- [17] Zhang, J., Silvescu, A., Honavar, V. *Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction*. Technical Report ISU-CS-TR 02-13, Computer Science, Iowa State University. 2002.