Towards a Mathematical Theory for Snapshot and Temporal Formal Ontologies

Andrew U. Frank, Department of Geoinformation, TU Wien, frank@geoinfo.tuwien.ac.at

Abstract: In order to achieve interoperability of GIS, the meaning of the data must be expressed in a compatible description. Formal methods to describe the ontology of data are increasingly used, but the detail of their definitions are debated.
In this paper I investigate the mathematical structure of formal ontologies as they are the background for ontology languages like OWL, which are increasingly used in GIS. I separate formal aspects of the ontology language from possible interpretations of the formulae in light of philosophical position. The paper gives formal description of a static and a temporal formal ontology. This clarifies what are assumptions (i.e., ontological commitments) and what are consequences of these. A formalized treatment leads to a consistent formal ontology and is the precondition for the integration of ontological descriptions of geographic data. The analysis shows that most of the important restrictions in ontologies can be expressed only in a temporal ontology and they are often related to what processes are included in the temporal ontology.

1. Introduction

The presentation of data from different sources is a natural requirement from users of geographic data; they want to combine all kinds of data found on the web to a comprehensive display. This requires interoperability of multiple databases. Technical interoperability is achieved with the standardization of query languages (ISO MM SQL), data types for simple features, etc. (OGC 1998). To achieve semantic interoperability (Riedemann et al. 1999) is more difficult. The semantics of data are at best expressed in some ontology language (Gangemi et al. 2002; SUMO 2003; Grenon et al. 2004; McGuinness et al. 2006) which have different formal and ontological foundations. In this paper these foundations are analyzed to allow schema merging as proposed by (Fonseca et al. 2002). Ontology has become increasingly backed by formalized descriptions. Papers on formal ontology were published (Smith 1998; Grenon 2003), but debates continue and terminological confusion persists (Kusnierczyk 2006). The efforts to standardize the top-level ontologies demonstrates the differences between well-reasoned ontologies and the difficulties to merge them (see for example the IEEE standardization effort at http://suo.ieee.org/ or (Smith 2006)). It seems that the currently debated contributions mix the purely formal apparatus of ontology with ontological commitments, which describe the author's ontological stance, i.e., his interpretation what "to exist" means (e.g., DOLCE (Gangemi et al. 2002), BFO (Smith 1998)). It is then difficult to see what is the minimal set of axioms to use and what are the consequences of the axioms included—separate from philosophical –isms (e.g., realism, nominalism, 3D- vs. 4D-ism).

A formalized ontology consists of two parts:

- a formal mathematical apparatus with symbols and rules and
- an interpretation of the symbols in the formal system in terms of the part of reality the ontology is set out to describe.

It is useful to separate the two, as the first is subject to mathematical rigor and formal proof whereas the second remains open to differences in opinion.

The formal part consists—as any part of mathematics—of a set of symbols and rules for inferences. If these inference rules are expressed as formulae and use widely acceptable theories (e.g., set theory, integer and real numbers) a discussion of this formal part of an ontology should be devoid of misunderstandings and consequences of the choices can be rationally discussed. The grounding of the symbols used in the formal ontology as an interpretation of what they mean in the real world is not debatable in formal terms and is influenced by philosophical (metaphysical) positions.

This paper tries to achieve two goals:

1. separate the formalized apparatus of an ontology from its grounding and
2. to describe a formalized apparatus for a core of a formal ontology and show how some of the more influential proposals for foundations of formalized ontologies relates to this.

The understanding of formal ontology here is narrower than the intention in the original paper by Smith (1998), where the separation of mathematical theory from the interpretation is not stressed. Smith follows in his choice of terminology a philosophical tradition, going back to Husserl (1900/01), I embrace the typical mathematical stance of constructing and analyzing an abstract system without assuming an interpretation.

2. Ontology as a Formal System

A formal ontology consists of a number of symbols and formulae giving inference rules for these symbols. It is customary to select symbols that remind a human reader of the intended interpretation, but the names used for the symbols are inconsequential for the mathematic treatment.

The text splits the formal ontology in two parts—first a discussion of a static (snapshot) ontology, some what similar to Smith and Grenon's SPAN (2004) and then a process based temporal ontology, roughly comparable to a SPAN ontology. In section 5 a motivation for some of the unusual consequences of the formalism is given using a geographic example.

By temporal ontology I understand an ontology where individuals can change, whereas a dynamic ontology would be an ontology in which the rules for classification etc. change in time (Sowa 2006). In a temporal ontology, we have changing individuals, for example moving glaciers or flying airplanes, whereas a dynamic ontology would permit to change the classification what a plane, a glacier, etc. is (Sellis et al. 2003). Eventually, GIS will require both, but here only the first is addressed.

3. The Static Snapshot Ontology

A static atemporal ontology represents a snapshot, non-changing view of the world. It does not contain a notion of time and does not allow change, similar to Smith and

Grenon's SPAN ontology (2004). The analysis reveals what can be expressed in this setting and what other important ontological considerations (e.g., metaproperties (Guarino et al. 2000)) are better expressed in a temporal ontology, where entities change over time. The investigation concentrates on the static interaction between the relations (*instances_of*, *is_a*, and *part_of*) comparable to the work of Bittner at al. (2004). The construction of the static ontology here starts with individuals (like DOLCE (Masolo et al. 2003)) and not with universals (Bittner (2004)).

3.1. Individuals

The ontology consists of a set of symbols for individuals, denoted by lower case letters: $\boldsymbol{S}=\{a,b,\ldots\}$. The domain of individuals will be denoted as $\boldsymbol{S}$.

3.2. Properties and quality values

Individuals have properties, which are *functions* from individuals to quality values. This is similar but somewhat simplified compared to Gärdenfors treatment of quality spaces (Gärdenfors 2000) or (Masolo et al. 2003).

$$p\colon \boldsymbol{S} \to \boldsymbol{Q}$$

Without limitation of generality, quality values will be from a set of ordinary values $\boldsymbol{R}$ (e.g., a subset of real numbers or integers) extended by some special values:

$$\boldsymbol{Q}=\boldsymbol{R} \cup \{unknown,\ not\_applicable\}$$

The property value for an individual may exist but not be known; then the property function p returns *unknown*. *Not_applicable* is the value returned when the individual does not have the property (e.g., color applied to "dream").

The individuals in this static ontology are considered as non-changing; thus properties are proper and total functions from individuals to values.

Ex.: area (a) = 81.420 km^2
height (a) = 4.203 m

3.3. Part_of relation

A *part_of*: $\boldsymbol{S} \to \boldsymbol{S}$ relation is defined between individuals; *part_of a b* means *a* is a part of *b*. The *part_of* relation is reflexive, antisymmetric, and transitive. The *part_of* relation has been axiomatized (Bittner et al. 2004).

$$part\_of\ a\ a$$
$$part\_of\ (x, y)\ \wedge\ part\_of\ (y, x) \to x=y$$
$$part\_of\ (x, y)\ \wedge\ part\_of\ (y, z) \to part\ (x, z)$$

The *part_of* relation gives a partial order among the individuals.

3.4. Classes and instances of relation

$\boldsymbol{C}$ is the set of classes with special values *unspecified* and *nap* (for the class *not_applicable*). The relation *instance_of* relates the individuals to the classes:

$$\boldsymbol{C} = \{c_1, c_2, \ldots, unspecified, nap\}$$
$$instance\_of: \boldsymbol{S} \rightarrow \boldsymbol{C}.$$

Classification starts with selecting a property p and a partition $\boldsymbol{K}$ of the quality space $\boldsymbol{Q}$ of this property p. The elements in the partition $\boldsymbol{K}$ are pairwise disjoint and jointly exhaustive (JEPD), such that we have a total function:

$$class': \boldsymbol{Q} \rightarrow \boldsymbol{K}$$

and an isomorphism between the subset of classes introduced by this partition $C_K=\{c_1, c_2, \ldots c_n\}$ and the elements in the partition

$$class'': \boldsymbol{K} \rightarrow \boldsymbol{C}.$$

This gives a function *instance_of* as a composition of the function p, *class''* and *class'*:

$$instance\_of = class'' \cdot class' \cdot p,$$

which is total. The property p_g and the partition $\boldsymbol{K}_h$ create a set of classes $\boldsymbol{C}_{gh} \subset \boldsymbol{C}$. *Class'* maps the value *unknown* to *unspecified* and *not_applicable* to *nap*. From these definitions follows that all individuals, which are an instance of a regular class produced by property p_g and partition element k_h have property values for p, which fall into the partition element k_h, which is usually an interval of $\boldsymbol{R}$.

These definitions for classes construct classes from individuals and property values; this justifies to use the term class and not universal. These definitions for classes are intensional by the conditions on property values. They can be empty, if no individual with required property exists.

3.5. Subclasses (is_a relation)

Consider a class c_i, which results from property p and partition element k_j of $\boldsymbol{K_h}$. A finer partition $\boldsymbol{K_l}$ of $\boldsymbol{Q}$ is a refinement of a partition $\boldsymbol{K_h}$ of $\boldsymbol{Q'}$ if there is a function:

$$sub : \boldsymbol{K_l} \rightarrow \boldsymbol{K_h}$$
$$sub\ (k_i) = k_j \ \ (k_j \in \boldsymbol{K_h}, k_i \in \boldsymbol{K_l})$$

For a fixed property p, the elements in $\boldsymbol{K_l}=\{k_i^l\}$ gives rise to classes C_i^l and the elements in $\boldsymbol{K_h}=\{k_j^h\}$ give rise to classes C_j^h (note that lower indices refer to the element in the partition and the upper index to the partition used).

$$\forall k^h \exists k \ \ subclass\ c_i^l\ c_j^h = sub\ k_i^l\ k_j^h \wedge class'\ k_i^l\ c_i^l \wedge class'\ k_j^h\ c_j^h$$

From these definitions follows that an instance c' of a class c_j is also an instance of any superclass c_j of c_i (*superclass* is the converse relation to *subclass*).

$$superclass\ c_i\ c_j \equiv subclass\ c_j\ c_i$$

3.6. Universal parthood

Bittner et al. (2006) have defined a reflexive and transitive parthood relation between classes such that every instance of one class is a part of an instance of the other.

$$parthood\ CD = (instance\_of\ c, C) \wedge (\exists d\ (instance\ d, D)) \wedge part\_of\ (c, d).$$

4. A Bipartite Structure of the Classification

Classes are ordered by the subclass relation and classes form, if duly expanded by a top ("everything class" $\top$) and a bottom element ("nothing class" $\perp$), a lattice (Gill 1976). The above sketched classification method leads to a bipartite structure of classes:

- classes distinguished by properties,
- classes distinguished by property values.

In this section the combination of several classifications is investigated.

4.1. Classes distinguished by properties

A classification n is constructed by selecting a property p_g and partition $\boldsymbol{K}_h=\{k_1^{gh}, \ldots k_n^{gh}, unknown, not\_applicable\}$ of the value space of p_g, with the corresponding classes $C^{gh}= \{c_1^{gh}, c_2^{gh}, \ldots c_n^{gh}, c_{unk}, c_{nap}\}$. Note: I used superscripts to indicate the property and the partition that was used in the classification. c_{nap}^{g} stands for the class that results from classifying an entity to which the property p_g does not apply (nap= *not_appicable*).

Partitions are partially ordered by a refinement relation (Gill 1976). Consider the least refined partition: $\boldsymbol{K}^{go}=\{k_o^{go}, not\_applicable\}$ with the classes $c^{go}=\{c_o^{go}, \ldots c_{nap}^{go}\}$. A less refined partition is not acceptable because it would violate the principle that defined values must not be classified with the special values *not_applicable*. This produces the lattice (Figure 1):

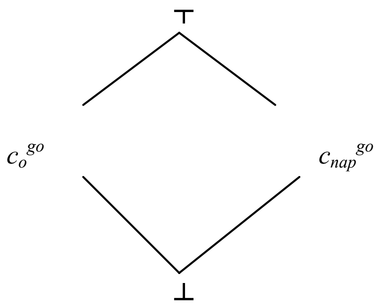

Figure 1

I suggest to call these classes generator classes, because they combine and generate the lattice of taxonomic classification (Frank 2006) (see 4.3).

4.2. Classes distinguished by property values

A requirement of a classification C_{gh} with property p_g and partition k_h is possible by refining the partition. If partition $\boldsymbol{K}_h$ is replaced by a finer partition $\boldsymbol{K}_l$. Then the finer classification C_{gl} results. The relation between the classification C_{gh} and C_{gl} can be shown as a lattice diagram:

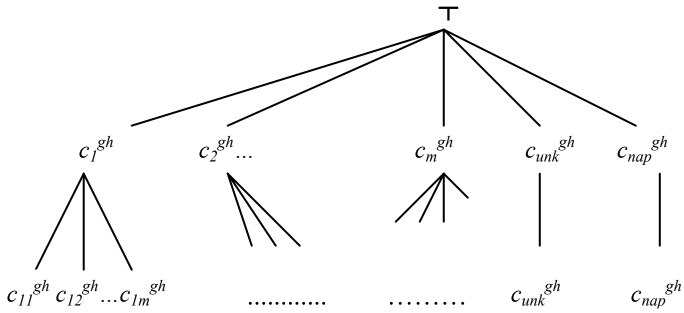

Figure 2

The refinement shown in Figure 2 is general. Such a refinement step can be applied to the least refined classifications, the generator classes, to produce any classification of increased refinement. In particular, starting with a generator class created with property p_g partition $\boldsymbol{K}_n$ (which is finer than $\boldsymbol{K}_o$). We obtain the lattice of Figure 3.

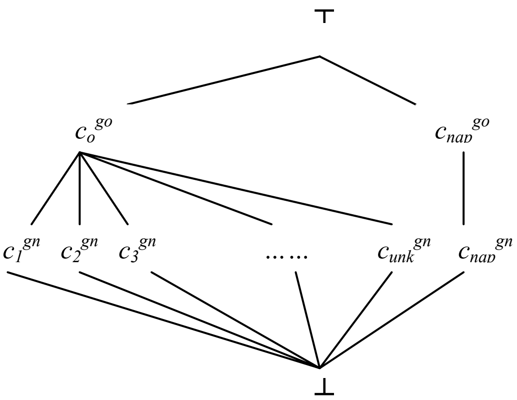

Figure 3

4.3. Combination of property based classes

Assume two generator classifications, one using property p_g and the other p_h, both using a partition $\boldsymbol{K}_o$ (Figure 4).

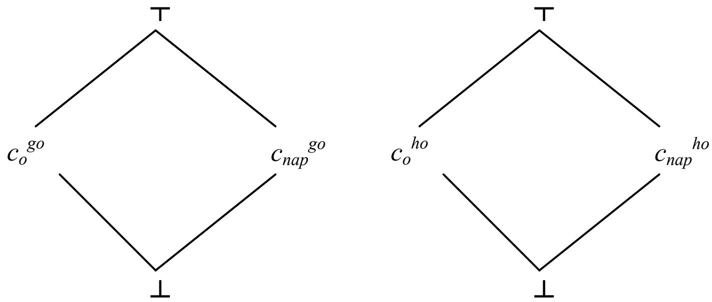

Figure 4

The combination of these two generator classifications gives the lattice (Figure 5):

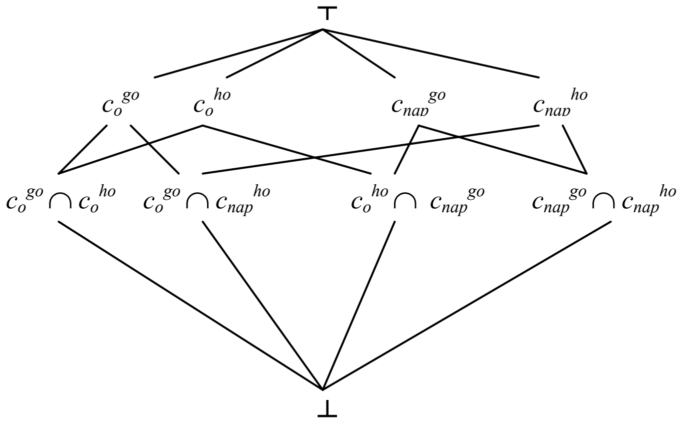

Figure 5

Further refinement of the classification through distinctions applied to the property values are possible.

5. Some Example from Geography as Motivation

The here described construction of classifications of individuals with properties is formal. In this section I will show how it applies to a simple geographically motivated ontology. The instances encountered in a subset of reality, here imperfectly invoked by the sketch of a topographic map Figure 6 could be classified by distinctions like:

- natural – man made,
- point – line – area features,
- land use,
- ownership, etc.

I will show here how such classifications are constructed based on properties.

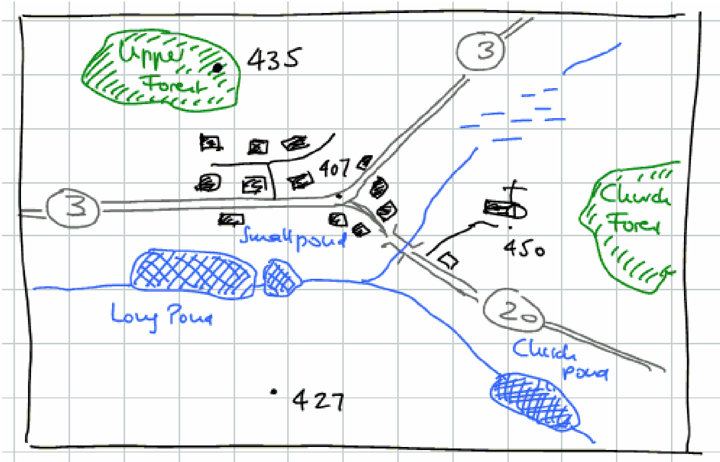

Figure 6: Sketch of a map Neukirch

In the village Neukirch and its environment we see individuals: 3 ponds, 2 forests, 13 buildings, 1 church, 3 road segments, etc. From experience we know that these classes are effective, because they group things by the interaction we can have with them. Human cognition focuses on object classes and not so much on the properties, even though the property values are what we perceive and use to classify. Consider the classification by dimension of the object, which can be constructed with two properties p_1, p_2, which observe area, p_2, and length, p_1 (Figure 7).

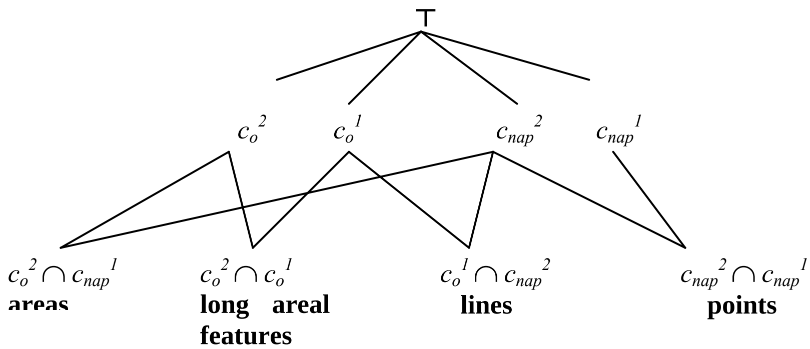

Figure 7

Individuals to which neither a length nor an area property applies are points; to lines only length applies and to proper areas only area applies. The unusual class where both area and length applies could be used to classify e.g., a road, which has a length but also a surface area. A classification of areas by land-use needs a property p_l, which depends on p_2 (meaning it applies only to areal individuals, see later 6.2). This generator class 'land used' is then refined with a partition for the values $\boldsymbol{K_l}$*={agricultural, forest, lake built}*; further refinements are possible (Frank et al. 1997). Combining man-made p_{mm} with the dimension gives

$c_o^2 \cap c_o^{mm} \rightarrow Building$
$c_o^2 \cap c_o^1 \cap c_o^{mm} \rightarrow Road$, *etc.*

It must be pointed out that the lattice construction produces all possible constructions of properties and partitions for their subdivision. Many are not possible on logical grounds and many others are impossible because "this is not the way the world is" or not relevant for an application. These unused classes are not a problem—they are there if one needs them, and before they are only potentially existing. Compare with the infinite set of integers—only those we need for a calculation are realized. The classes in a taxonomic lattice operate the same way.

6. Limitations of a Static Ontology

It is easily observable that the described formal structure is not capturing all that is important to characterize a conceptualization and communicate it. Important are restrictions, Guarino et al. proposed to distinguish identity, unity, essence, and dependence and introduced meta-properties to capture these ontologically important notions (Guarino et al. 2000). Not all of these metaproperties can be expressed in a static (snapshot) ontology; Guarino and Welty used a modal logic to express them but I prefer to defer treatment to the temporal ontology.

6.1. Identity

A property p is said to provide an identity condition ρ (ρ is an equivalence relation) if

$$\rho\,(p\,(x), p\,(y)) \leftrightarrow x{=}y.$$

In a static ontology only a *synchronic* identity can be expressed.

6.2. Dependence

Dependence of an individual on another individual is similar to a metaproperty. Simon has given an extensive discussion of dependence (Simons 1987). To say that an entity a depends on another entity b means that a relates to b through a relation r:

$$\forall a \exists b\ r(a, b).$$

6.3. Rigidity

For a static view, rigidity means that all instances of a class must have a property. Given that classes are constructed from properties and values, classes so constructed capture a static rigidity. Individuals that do not have the property (value of p is *not_applicable*) fall into the special class *nap*. We will see later (8.1) that rigidity is generally useful to express other metaproperties.

6.4. Rigidity properties and part_of relation

The interaction between property and the *part_of* relations allows to characterize some properties as clay-like. A clay-like property p, which applies to an individual a_o applies also to all other parts of a_o. The terminology was introduced for temporal reasoning, but it seems to be applicable for ordinary parts as well (Shoham 1988; Beard et al. 1991; Barrera et al. 1992).

$$\textit{clay-like } p \leftrightarrow \textit{part\_of } a\ b \ \wedge\ p\ a \neq n/a \rightarrow p\ b \neq n/a$$

7. Formal System for a Temporal Ontology

GIS must represent a changing reality and must be built with a dynamic ontology. Geographers stress the processes that occur in reality and shape our environment (Abler et al. 1971). Therefore I develop a temporal ontology in this section with processes, and select—unlike other treatments in the literature—an algebraic view with functions that change states. This avoids the frame problem that cannot be avoided when using logic and situation calculus as did Bittner et al. (2004).

7.1. Time

In a temporal ontology states are indexed by time. We assume a dense and continuous time $\boldsymbol{T}$ with time points t. Time is ordered by a relation *before* $(\leq)$ in the usual sense (Galton 1987).

7.2. Individuals

The set of individuals is written as $\boldsymbol{I}$ and individuals with lower case letters *a, b, ….* Individuals exist for intervals of time. Individuals have a function *exists* from time to a Boolean value: $\boldsymbol{T} \rightarrow \boldsymbol{B}$. This is the characteristic function of the relation *exist:* $\boldsymbol{T} \rightarrow \boldsymbol{I}$.

7.3. Individuals have changing states

The state of an individual can change in time; the state at time t of an individual in a temporal ontology corresponds to an individual in a static ontology. The function *state* is the mapping from a dynamic ontology (similar to a SPAN ontology (Smith et al. 2004)) to a SNAP ontology. The “snapshot” function *state :* $\boldsymbol{I} \rightarrow \boldsymbol{T} \rightarrow \boldsymbol{S}$ maps the individual to its (static) state at t. If the individual does not exist at t the state has a special value *notExist*. The set of state $\boldsymbol{S}$ must be duly extended; in the following $\boldsymbol{S}$ means $\boldsymbol{S}_D$. ($\boldsymbol{S}_s$ is the state from the static ontology)

$$\boldsymbol{S}_D = \boldsymbol{S}_s \cup \{\textit{not\_exist}\}$$

This allows to define the relation *exist* as:

$$\textit{exist } i\ t \leftrightarrow (\textit{state } i\ t \neq \textit{not\_exist}).$$

7.4. Processes

Processes take individuals from one state to another state

$$a : \boldsymbol{T \to T \to S \to S}$$

and preserve identity of the individual

$$a\ t_1\ t_2\ s_1 \leftrightarrow s_2 \to \exists i\ state\ i\ t_1 = s_i\ \wedge state\ i\ t_2 = s_2.$$

The letter *a* (for *a*ction) is used to denote processes. For any two states of an individual *i* at t_1 and t_2 $(t_1 < t_2)$ exists a process *a*, which converts *state i* t_1 to *state i* t_2.

7.5. Property values of individuals change

The property values obtained by a property function from individuals may change. This is a consequence of the changing states of individuals.

$$v_1 = p_j(state\ i\ t_1) \qquad v_2 = p_j(state\ i\ t_2)$$

v_1 and v_2 may be different, if a process a_k was active on the individual between t_1 and t_2 and if *a* affected the property p_j. We say the process a_k affects property p_j if

$$affects\ a_k\ p_j = \exists_i\ \exists t_1 \exists t_2\ a_k\ i\ t_1\ t_2\ \ p_j(state\ i\ t_1) \neq p_j(state\ i\ t_2)$$

Property values are *undefined* if an object does not exist at t_1, but have a definite value (or *unknown*) at any time the object exists

$$\forall_p\ p\ not\_exist = undefined$$

7.6. Classification of individuals

The classification of individuals follows the same principles as described before (3.4). A function *dynclass* is defined as

$$dynclass : \boldsymbol{I \to T \to C}$$

$$dynclass\ i\ t = class\ (state\ i\ t)$$

and the class of an individual can, in general, change with time, i.e., the *instance_of* relation changes over time.

8. Restrictions

An important part of an ontology describes restrictions that must hold between the individuals. These restrictions in a temporal ontology are either stated in a temporal logic ("it is never the case that", "it is always the case that") or—as we will show here, by excluding certain types of changes.

8.1. Rigid meta properties

A meta-property (e.g., the *instance_of* relation) is rigid if there is no action, which affects the property p , which is used for the classification.

$$M\ K_l = rigid \equiv \forall t \quad M'\ K_l\ t$$

$$M\ K_l = rigid \quad \neg \exists p\ affects\ p\ a \ \wedge\ depend\_on\ M\ p$$

The relation *depends_on* is changing in time if in the definition of M the property p appears. For example, a classification C depends on p if p is used to obtain the quality value that is used for the classification. This applies for example to the identity condition to extend it from synchronous to dynamic. The synchronous identity depends on a property p_i, which is obtained from both individuals and compared by ρ. If no process a which *affects a p* is included in the ontology then the identity is preserved diachronically. With this meta^2-property rigid a general class of conditions in a temporal ontology can be specified. For example, the rigidity of any class, relation, identity, or *part_of* relation can be described.

The condition that no process must be included in the ontology that affects the property in the context where rigidity is expected leads to a centrally controllable consistency constraints. Properties that can only be set initially, when an individual is created and for which no process to change it exists, are *constraints*.

8.2. Examples

In the ontology style I currently favor, the fact that a property is applicable to an individual is rigid. A road remains a "long areal feature" but its classification as 1st or 2nd class road can change. Therefore quality values *undefined* are only assignable when an individual is initially created, but never a property that was obtainable can disappear—for example something that had some matter may not become abstract (immaterial). Therefore transition from $R \cup \{unknown\}$ to *undefined* are not allowed for any action a. This maintains the set of classifications that are based on properties. Most top-level ontologies assume for the included distinctions rigidity in this sense. For example, I assume that the twelve top classes by Sowa (1995, 72) have this property: a *history* cannot become a (physical) *object*, a *reason* not a *process*.

9. Conclusions

This paper has shown that a purely formal discussion of the foundations of an ontology yields valuable insights. It becomes clear what are the interactions between formal rules and what are choices depending on debatable ontological comments. The analysis of a static ontology has shown the limitations of what can be described statically. Many important aspects of a meaningful ontology (e.g., rigidity of a property) cannot be expressed directly. The definitions found in the literature use a modal logic; in this paper I have shown that they can be expressed in a temporal ontology with processes, albeit using a functional approach not restricted to FOL.

The paper has shown the interactions between the relations

- *instance_of*
- *is_a*
- *part_of*

in a static and temporal ontology. Rigidity as a meta^2-property can be related to the absence of actions (processes) that can affect the properties some metaproperty is based on.

In this framework general preservation laws can be introduced for some properties—to reflect the preservation of matter or energy in the real world. This will result in a more abstract formalization of matter and should correspond to Bennett's account (Bennett 2001).

The combination of classifications that are missing one property introduces much flexibility but maintains the mathematical rigor of lattice theory (which has been shown useful in the related context of Formal Concept Analysis FCA (Wille 2000; Krötzsch et al. 2005; Priss to appear). The taxonomic lattice contains any class possibly useful in a context with a given set of properties,—ready to serve when necessary, but not cluttering the description of the ontology before. The notation in this paper put the focus on the construction, a more succinct notation was given before (Frank 2006) where simple operations to merge taxonomies were demonstrated.

Whitehead said "We must be systematic, but we should keep our systems open" (Sowa 1995, 75). The construction of an ontology from pieces that combine at will seems to satisfy this goal better than to fix a top level ontology forever.

References:

Abler, R., J. S. Adams and P. Gould (1971). Spatial Organization - The Geographer's View of the World. Englewood Cliffs, N.J., USA, Prentice Hall.

Barrera, R., M. J. Egenhofer and A. U. Frank (1992). Robust Evaluation of Spatial Queries. 5th International Symposium on Spatial Data Handling, Charleston, South Carolina, USA, IGU Commision on GIS.

Beard, M. K., B. P. Buttenfield and S. B. Clapham (1991). NCGIA Research Initiative 7 - Visualization of Spatial Data Quality. NCGIA Technical Paper, National Center for Geographic Information and Analysis.

Bennett, B. (2001). Sapce, Time, Matter and Things. FOIS'01.

Bittner, T. and M. Donnelly (2006). A Theory of Granular Parthood Based on Qualitative Cardinality and Size Measures. International Conference on Formal Ontology in Information Systems, Baltimore, Maryland, IOS Press.

Bittner, T., M. Donnelly and B. Smith (2004). Individuals, Universals, Collections: On the Foundational Relations of Ontology. Third Conference on Formal Ontology in Information Systems FOIS-04, Turin, IOS Press.

Fonseca, F. T., M. J. Egenhofer, P. Agouris and G. Câmara (2002). "Using Ontologies for Integrated Geographic Information Systems." Transactions in GIS 6(3): 231-57.

Frank, A. (2006). Distinctions Produce a Taxonomic Lattice: Are These the Units of Mentalese? International Conference on Formal Ontology in Information Systems, Baltimore, Maryland, IOS Press.

Frank, A. U., G. S. Volta and M. McGranaghan (1997). "Formalization of Families of Categorical Coverages." IJGIS 11(3): 215-231.

Galton, A., Ed. (1987). Temporal Logics and Their Applications. San Diego, CA, USA, Academic Press.

Gangemi, A., N. Guarino, C. Masolo, A. Oltramari and L. Schneider (2002). Sweetening Ontologies with DOLCE. EKAW 2002.

Gärdenfors, P. (2000). Conceptual Spaces, MIT Press.

Gill, A. (1976). Applied Algebra for the Computer Sciences. Englewood Cliffs, NJ, Prentice-Hall.

Grenon, P. (2003). BFO in a Nutshell: A Bi-categorial Axiomatization of BFO and Comparison with DOLCE, IIFOMIS.

Grenon, P., B. Smith and L. Goldberg (2004). Biodynamic Ontology: Applying BFO in the Biomedical Domain. Ontologies in Medicine. D. M. Pisanelli. Amsterdam, IOS Press: 20-38.

Guarino, N. and C. Welty (2000). A Formal Ontology of Properties. Proceedings of EKW-2000, 12th Int. Conference on Knowledge Engineering and Knowledge Management (2-6 October, 2000). R. Dieng and O. Corby. Berlin Heidelberg, Springer-Verlag.

Husserl (1900/01). Logische Untersuchungen. Halle, M. Niemeyer.

Krötzsch, M., P. Hitzler, M. Ehrig and Y. Sure (2005). Category Theory in Ontology Research: Concrete Gain from an Abstract Approach. Karlsruhe, Germany, Institut AIFB, Universität Karlsruhe: 6.

Kusnierczyk, W. (2006). Nontological Engineering. International Conference on Formal Ontology in Information Systems, Baltimore, Maryland, IOS Press.

Masolo, C., S. Borgo, A. Gangemi, N. Guarino and A. Oltramari (2003). WonderWeb Deliverable D18 (Ontology Library). Trento, Italy, Laboratory For Applied Ontology - ISTC-CNR: 247.

McGuinness, D. and F. van Harmelen. (2006). "OWL Web Ontology Language Overview." from http://www.w3.org/TR/owl-features/.

OGC (1998). OpenGIS Simple Features Specification for SQL. Wayland, MA, Open GIS Consortium.

Priss, U. (to appear). "Formal Concept Analysis in Information Science." Annual Review of Information Science and Technology 40: 22.

Riedemann, C. and W. Kuhn (1999). What are Sports Grounds? Or: Why Semantics Requires Interoperability. Interoperating Geographic Information Systems (Interop'99, Zürich). A. Vckovski, K. E. Brassel and H.-J. Schek. Berlin, Springer-Verlag. 1580: 217-230.

Sellis, T. and e. al., Eds. (2003). Spatiotemporal Databases: The Chorochronos Approach. Lecture Notes in Computer Science. Berlin, Springer-Verlag.

Shoham, Y. (1988). Reasoning about Change. Cambridge, Mass., The MIT Press.

Simons, P. (1987). Parts - A Study in Ontology. Oxford, Clarendon Press.

Smith, B. (1998). The Basic Tools of Formal Ontology. Formal Ontology in Information Systems. N. Guarino, IOS Press: 19-28.

Smith, B. (2006). Against Idiosyncrasy in Ontology Development. International Conference on Formal Ontology in Information Systems, Baltimore, Maryland, IOS Press.

Smith, B. and P. Grenon (2004). "SNAP and SPAN: Towards Dynamic Spatial Ontology." Spatial Cognition and Computing(4): 69-103.

Sowa, J. F. (1995). "Top-Level Ontological Categories." International Journal of Human-Computer Studies 43(5-6): 669-685.

Sowa, J. F. (2006). A Dynamic Theory of Ontology. International Conference on Formal Ontology in Information Systems, Baltimore, Maryland, IOS Press.

SUMO. (2003). "Overview of the SUMO." Retrieved 11.03., 2003.

Wille, R. (2000). Boolean Concept Logic. Conceptual Structures: Logical, Linguistic and Computational Issues. B. Ganter and G. Mineau. Berlin-Heidelberg-New York, Springer. LNAI 1867: 317-331.