Summa Contra Ontologiam

Simone Santini

Escuela Politécnica Superior
Universidad Autónoma de Madrid, Madrid, Spain
and University of California, San Diego, USA

Abstract. This paper is a critical analysis of the concept of ontology thus as it is used in contemporary computing science. It identifies three main problems with such a concept, two of which are intrinsic to it and one of which is extrinsic, so to speak, being related to the use that of ontology is made in applications.
The first problem with ontology is that the only accepted definition of its main artifact is teleological rather than structural as it would be proper in computing science. The second problem is that claiming that ontology is in any way a semantic discipline requires such a limited and outdated notion of semantic to be to all practical purposes useless. The third and final problem is that the limitations and misconceptions of ontology might make it a limiting factor, rather than a help, for many of the applications for which it is sought.
The article concludes that a profound reconsideration of the relation between computers and semantics might be overdue.

1 Introduction

The purpose of this paper (the lifting-cum-paraphrase of whose title I hope St. Thomas Aquinas will forgive) is to analyze the foundations and the value (in a rather broad sense) of what today is commonly known, among researchers and practicioners of information systems, as *ontology*. I will make three arguments: the first two will be endemic to computing science, while the third will be of a broader nature, touching upon the relation between the ontological offering of computing science and the disciplines to which the offer is made. Of the two endemic arguments, the first will be of a formal nature, viz. the investigation of an acceptable definition of ontology, while the second is of a more theoretical nature, so to speak, in that it deals with the common assumption that ontology can be used to formally specify the *semantics* of a certain domain of discourse. This division can be seen, *mutatis mutandi*, as a mirror of the linguistic division into which ontology would locate itself: the first section (the one on the definition of ontology) deals with its *syntax*; the second section deals with the *semantics* of ontology, and the third with the way ontology is applied in disciplines outside of computing science, that is, it deals with the *pragmatics* of ontology. The parallel with the three traditional levels of linguistics should not be taken too seriously, of course, but it should be located somewhere between the general guideline and the pure *divertissement*.

Before taking on these subjects, however, I should like to take a little space to settle, once and for all, a terminological matter. The word ontology originates in metaphysics where it is, according to the *Britannica*, the study of being as such, i.e. of the basic characteristics of all reality.

Ontology, as defined in metaphysics, is the study of *being*, not of *beings*: it is not a taxonomy of existing things and consequently, for instance, Linneus never claimed to be doing ontology (and quite correctly so). It is true that John Scot's *De divisionis Naturae* is generally regarded as an ontological work, but this is due to its Platonic assumption of the universals as the only reality, and not to the taxonomical structure of the work per se. The word ends with the suffix *-logy* and, due to the programmatic rather than methodological connotation of the suffix in this case, it has no plural: there are no different studies of being as such, but any way of studying it is part of the same discipline of ontology. The Oxford English Dictionary, quite correctly, doesn't report any plural for the word ontology.

In computing, the word ontology is used with two different connotations: as a discipline and as the artifacts that the discipline produces. While the term ontology is usable (by an admittedly rather daring metaphorical extension) in the first case, as a name for the artifact it is clearly improper: a better name in this case would be *ontonomy* (plural *ontonomies*)[1]. In this paper, I will keep the distinction and refer to the discipline as ontology and to the artifact as ontonomy.

Some readers might see into all this the expression of too fine a pont, an empty pseudo-intellectualistic annoyance, but I disagree quite emphatically with any such assessment: computing is a mathematical discipline, and precision in the terms that one uses is of the greatest importance for it, the unwanted connotations of a term often leading to confusion. Since confusion is precisely what I ascribe the existence of computational ontology to, it is important to try to avoid falling into easy connotational pitfalls, the risk of pedantry being in any case preferrable to that of imprecision.

2 Syntactic definition of ontology

Given the importance that these days is ascribed to ontology in many areas of information management, it is surprisingly hard to come across a mathematically acceptable definition of the discipline or of its artifacts. The most common definition or, at least, the one that I hear around the most, is along the following lines:

an ontology is a formalization of a conceptualization.

The most quoted source of this definition appears to be [7], but the definition is widely accepted (see [12, 2], for example). The problems of this definition are of several orders, and I will consider some of them in a short while but, for the moment, I would like to concentrate on the *functional* nature of it: this definition doesn't tell us what an

[1] Quite surprisingly (to me, at least), this is not a neologism. The term was used in 1803 by J. Stewart in his *Opus Maximum* as more or less as synonym of ontology. Given its derivation from $o\nu\tau o\varsigma$, (present participle of *to be*) and $\nu o\mu\iota\alpha$, (*distribution, arrangement*), I think that my connotation is better than Stewart's. A more appropriate name, with less metaphysical baggage, would be (from $o\iota\kappa o\varsigma$, *household*) the word *economy* but, as the readers undoubtedly know, the term is already used for an altogether different discipline.

ontology *is* but, rather, what it is (generally) used for. This kind of definition is of course unacceptable in computing science.

Consider, as a parallel, the definition of formal grammar: if researchers in programming languages had used the same criteria of rigor (or lack thereof) as researchers in ontology, the definition would sound something such as:

a formal grammar is the specification of a programming language.

But this is unacceptable: for one thing, while computing scientists do use formal grammars in order to formalize programming languges, there is no reason why this should be their only use. Linguists (at least those adhering to the Chomskyan current of the Anglo-American philosophy of language) use it to describe parts of natural language, and there is no reason why other uses should not be found. Finding new uses will not change the nature of the artifact, but will invalidate functional definitions such as the one given above. A functional definition describes the use of an artifact, but it doesn't specify its nature and structure, how we can identify it: given an arbitrary string of symbols, a definition should allow one to determine whether the string is a formal grammar or not. To this end, a *structural* definition is necessary. In the case of formal grammar, the definition is the well known one: a formal grammar is a 4-tuple (N, T, S, P), where N is a finite set (called the set of non-terminals), T is a finite set, disjoint from N (called the set of terminals), etc.

With this definition, one can proceed to define the language recognized by the grammar, and its properties. In the case of ontology, with very few exceptions, a structural definition is not provided: researchers are building a huge edifice on a formal structure without knowing what that structure is. Artificial intelligence does indeed have a definition of ontonomies but, in that case, an ontonomy is defined as the collection of all symbols used in a logic system, with the indication of which names are functions, which are predicates, and which are constants [10]. There is nothing wrong with this definition, but it certainly bears little resemblance to what the information system ontologists are doing. In particular, this definition doesn't include any relation between the terms and doesn't lay any semantic claim.

* * *

Many researchers in information systems are blissfully unaware of the fact that they are building such an enormous edifice on such weak foundations, but not all. One interesting attempt at a formal foundation of ontology has been made by Guarino [8]. Guarino's starting point is the notion of *intensional relation*. Consider a relation such as $[\text{above}](x, y)$ (which contains all pairs x, y such that x is above y). One can consider a set of objects, say a, b, c, and d, and create a relation, say

$$[\text{above}] = \{(a, b), (a, d), (b, d)\}, \tag{1}$$

which states that a is above b and d, and b is above d. This definition makes the concept "above" dependent on the specific configuration of a, b, c, and d: if b were above a, instead of a being above b, the relation would change. This extensional notion of "aboveness" is in this sense unsatisfactory: the *concept* of one thing being above another should be independent of the particular world configuration that we are analyzing. Guarino solves this insufficiency by introducing the notion of *intensional relation*.

Let D be a set of elements. An n-ary relation on D is a subset of D^n and therefore 2^{D^n} is the set of all n-ary relations on D. Let W be a set of worlds, that is, *grosso modo*, a set of legal configurations of the elements of D. An intensional relation r is an assignment, to each possible world in W, of a relation (n-ary, in this example) on D, that is, an intensional relation is a function

$$r : W \rightarrow 2^{D^n}. \tag{2}$$

So, given a world w in which a is above b and nothing else is above anything, we would have

$$[\text{above}](w) = \{(a, b)\}\,. \tag{3}$$

Given a logical language $L(V)$ built on a vocabulary V, an extensional model for $L(V)$ is a pair (D, R)—where D is a set, and R a set of relations on D—such that V can be mapped to D and predicates of L to elements of R. Similarly to this standard definition, Guarino defines an intensional model for a language by replacing R with a set of intensional relations. An intensional model for $L(V)$ can be seen then as a function that maps any possible world w to an extensional model relative to that world.

This intensional interpretation of a language is also called an *ontological commitment*. An ontonomy (my term, of course, not Guarino's) is then defined as follows:

> Given a language L, with ontological commitment K, an [ontonomy] for L is a set of axioms designed in a way such that the set of its models approximates as best as possible the set of intended models of L according to K ([8])

There are three ways in which this definition is unsatisfactory, at least as a computing science theory (some points of the definition have a certain philosophical interest, but this is besides the point in a computing milieu).

Firstly, the notion of intensional relation is in some way related to the Kripkean notion of possible worlds, but with some important distinguo. In Kripke, possible worlds are formal models indexed by a variable that corresponds to a degree of modality. A predicate is true (false) in a world depending on what it predicates about the extensional relation existing in the model corresponding to the degree of modality of that world. Extensional relations are what determine the essence of the world and, therefore, what determines the structure of a model.

In the case of ontology, however, we have to resort to the notion of possible worlds in order to define extensional relations, which implies that extensional relations can't be expressed in the world (if they were, the extensional relations would be logically prior to the intensional, and the latter could not be used to define the former). But, if this is the case, no possible world can have any structure, and not only can't they be used as a model in the Kripkean sense, but they can't even induce an extensional relation.

To put it in a different way: given a formal world of blocks, in order to instantiate the extensional relation [above], one needs to know whether block a is above block b; but the only way in which this can be known is to check whether $(a, b) \in$ [above]:

the worlds, that one needs in order to define the intensional relation, can only have structure by virtue of the extensional relations that the intensional ones are supposed to define. We are stuck in the middle of a circular argument. All this does not imply that intensional relations do not exist, but it does imply that, whatever they are, they are not a function from worlds to extensional relations, as the model requires.

Secondly, an ontology is defined as a system of axioms that defines (approximately, but this will be my third point) the set of models of a language L. On one hand, this definition leaves one with the complete freedom to choose the logic system in which these axioms are drawn while, on the other hand, it makes an ontonomy dependent on the choice of the language L. Given this latitude, it is not clear whether this definition defines anything worth defining. In order to dispense with the dependency on the language L (which runs quite against the common notion of ontonomy), one could say that an ontonomy is a system of axioms for which there is a language L such the axioms define the same set of models as the ontological commitment of L. But, presumably, for all non-contradictory set of axioms it is possible to define such a language so the definition would reduce simply to the statement that an ontonomy is any set of statement in any formal language. Such a definition is formally correct, but it is also so generic as to be of no use.

Thirdly, we have the presence of the word "approximates." With this addendum, any system of statements that admits at least one model that is also a model for a language L is an ontonomy for L. If we abstract from the language, then any set of statements that admits at least a model is an ontonomy. In particular, any set of tautologies is an ontology. Allowing for approximation, in other words, worsens the problem considered in the previous point: the definition is formally consistent, but too broad to be of any use: many things, from a C program to a very well structured grocery list, to a tax return form would qualify.

To this, one should add that Guarino's definition is not innocent of functionalism: given a set of statements in a certain logic system the only thing that makes them into an ontonomy is their intended use: if the statements are used to provide models consistent with an ontological commitment, then they form an ontonomy, otherwise they don't. This is not a *structural* definition of the type that computing science seeks. To come back to my previous example, given the definition of a grammar, and an arbitrary string, then one can decide whether the string is a grammar based on structural considerations only, even if one doesn't know that grammars are used to specify languages. The definition is structural and, in a sense, *closed:* it can be checked by making reference to the definition alone, without any teleological consideration. With Guarino's definition of ontonomy such a possibility does not exist.

* * *

A formally correct, structural definition of ontonomy that has been proposed in the literature is based on the algebraic theory of abstract data type, in particular on the notion of sub-typing. The theory has been proposed, e.g. by Bench-Capon and Malcom in [1], and its theoretical presupposition are in Goguen and Meseguer's *order-sorted algebras* [6]. An order-sorted algebra is a multi-sorted algebra $(\Omega, (A_\alpha | \alpha \in S))$ where the set of sorts S is endowed with a partial order relation called the *sub-sort* relation. Given a partially ordered set of sort names $\mathbb{S} = (S, \leq)$, a collection Σ of types equation symbols, and a set E of equations on the symbols of Σ, one obtains a *order-sorted equational theory* $T = (\mathbb{S}, \Sigma, E)$. If D is a model of T, then call (T, D) a *data domain*.

If we have a set of classes with attributes, Bench-Capon and Malcom use the sorts of the order-sorted equational theory to model the attributes, a data domain to model the attribute values, and a separate order model to model the classes:

Definition 1. *An* ontology signature *is a triple* $(\mathcal{D}, \mathcal{C}, A)$*, where* $\mathcal{D} = (T, D)$ *is a data domain,* $\mathcal{C} = (C, \leq)$ *is a partial order, called a* class hierarchy*, and* A *is a family of sets* $A_{c,e}$ *of* attribute symbols *for* $c \in C$ *and* $e \in C + S$*, where* S *is the set of sorts in* T*. The family is such that* $A_{c',e} \subseteq A_{c,e'}$ *whenever* $c \leq c'$ *and* $e \leq e'$[2].

An ontonomy is then simply a pair (Σ, A), where Σ is an ontology signature and A a set of axioms. A model of such an ontonomy is a model of Σ that satisfies the axioms of A.

While this is a rigorous structural definition, it has the problem of reducing ontology to the specification of a type system whose components are structures[3]. The basic relation between types here is the sub-typing relation (the partial order $\leq$ among classes), while all other relations have to be introduced as attributes. In other words, this model is strongly oriented towards monocriterial taxonomies, although it is more general than a simple taxonomy in that it allows the classification structure to be a partial order (viz. a directed acyclic graph) rather than a tree. It is, in other words, too limited a definition to cover the uses that are being done of the idea of ontology: while the theory can serve as the foundation of *a* discipline of ontology, it is too weak to provide a sound basis to the current uses of the notion.

Even with all these defects, the model of Bench-Capon and Malcom has the clear advantage of providing the type of definition necessary for computing science, without resorting to unacceptable teleological notions. To the best of my knowledge this is, to this date, the most promising attempt at a definition of an ontonomy, and the one most likely to be eventually extended to an acceptable definition.

3 Ontology and semantics

While a correct structural definition is still very elusive (and while most practicioners seem blissfully unaware of—or unconcerned by—the problem), there is a general consensus that ontology is somehow involved with the *semantics* of an information system, that ontonomies, whatever their structural definition will turn out to be, contain *concepts* and relations between them. The idea is not immediately obvious: ontonomies contain symbols, not unlike any formal system (or not unlike any data base, for that matter), and one should wonder what makes the symbols contained in an ontonomy be

[2] [1].

[3] Bench-Capon and Malcom call these elements "classes," but they should not be confused with the classes used in object oriented models: the classes of this model are not abstract, since their attributes are explicitly declared.

concepts. This question constitutes the semantic problem of ontology, which I propose to discuss in this section.

It is worth reminding that, when we are talking about semantics in the context of ontology, we are talking about something very different than, say, the semantics of a programming language. In programming languages, semantics is simply a function from states to states of a certain abstract machine. In ontology, the semantic that is modeled is supposed to be the semantics of the group in which the system is inserted, that is, the relation between the data in an information system and the symbols of an ontology is supposed to be isomorphic to the signification relation between *signifier* and *signified* in human culture. So, while the development of a theory of programming language semantics doesn't require any cognitive endorsement, the assertion that there is a computational discipline of ontology requires the endorsement of a theory of signification. Unlike programming language semantics, information system semantics (and ontology with it) is not theoretically innocent, so to speak.

If we take the rather general view that an ontonomy is a set of symbol and of relations between them then, with respect to the problem of meaning, one might ask whether these relations are constitutive or not. A negative answer leads to an atomism á la Fodor, while a positive answer leads to a point of view resembling very much what Fodor himself calls *inference rôle semantics*. I will begin by assuming that the answer is negative, that is that conceptual atomism is the theory of choice.

* * *

The possibility that symbols signify "by themselves" that is, independently of the relations between them requires the endorsement of a very strong form of conceptual atomism found, as far as I can tell, only in Fodor's *informational semantics*:

> Informational semantics denies that "dog" means *dog* because of the way it is related to other linguistic expressions [...]. Correspondingly, informational semantics denies that the concept DOG has its content in virtue of its position in a network of conceptual relations[4].

Note that the "correspondingly" here requires a fairly important metaphisical investment since it maps conceptual structures to linguistic ones. This, *passim,* is the same investment that ontology requires when it takes a linguistic structure (constituted of words and relations) and calls it a conceptual model.

Informational semantics has to struggle hard to eradicate itself from radical nativism: by the account of representational theories of mind, only composed concepts can be acquired (through inference from their components), thus conceptual atomism seems to imply that *all* concepts are innate. For Fodor, acquiring a concept means "getting *nomologically locked* to the property that the concept represents"[5] but the way to acquire a concept is having the right kinds of experiences. So, if one doesn't want to throw away the conceptual atomism baby together with the radical nativism bath water,

---

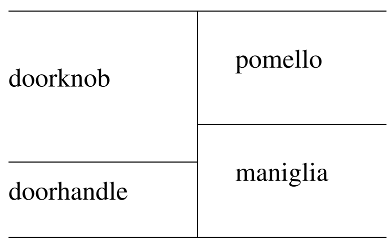
[4] [3], p. 73.

[5] ibid. p. 125, emphasis in the original.

the problem that one faces is "why is it so often experience with doorknobs, and so rarely experience with whipped cream or giraffes, that leads one to lock to doorknobhood?"[6]

Explanations that rely on hypothesis testing turn out to deny atomism, so they can't be applied. Fodor's solution to this problem is to stipulate that doorknobhood is *constituted by how its strikes us,* viz. "*being a doorknob* is having the property that minds like ours come to resonate to in consequence to relevant experience with stereotypical doorknobs"[7]

The serious flaw of this notion is that, since it needs to avoid any oppositional or structural definition of meaning in order to save atomism, it can't take into account the dependence of a single concept on the way in which different cultures divide the semantic field. To stay on Fodor's example, the English words "doorknob" and "door handle" correspond (roughly) to the Italian words "pomello" and "maniglia." But the areas covered by these concepts are not the same: while pomelli are, in general, doorknobs, some of the things that English speakers call doorknobs would qualify, for the Italian, as maniglie. The schema is more or less the following:



Why is it then that Italian minds "resonate" with doorknobs differently than English minds? By Fodor's account, there must be something different between English and Italian minds. A consequence of the consumption of wine and olive oil?

It appears, in other words, that we can't give a sensible explanation of the difference between doorknobs and pomelli unless we consider them differentially and oppositionally in the context of their respective languages. Doorknob is not a positive term, but serves to establish a distinction, an opposition in the semantic field of a language. Different languages break the semantic field in different ways, and concepts arise at the fissures of these divisions. Consider, as an example, the way in which adjectives of old age are constituted in Italian, Spanish and French[8]. The basic adjective, *vecchio/viejo/vieux* is applied both to things and to persons. There are specific forms, however: in Spanish, *añejo* is an appreciative form used mainly for alcoholic beverages *(un ron añejo)*. The Italian adjective *anziano* applied mainly to people, and the correspondence is roughly *anziano/anciano/âgé*, but *anziano* has a broader meaning that the other two adjectives, being used in expressions such as "il sergente anziano" to denote seniority in a function, a situation in which the Spanish would use *antiguo* and the French *ancien*. Note that the Spanish also has the possibility of using the word *mayor* as a softer and more respectful

[6] ibid. p. 127
[7] ibid. p. 137, emphasis in the original.
[8] I am taking this example, with some adaptation, from [5].

form of denoting a person of old age, while the corresponding Italian and French words are never used in this sense. The correspondence is, in other words, according to this schema

Italian	Spanish	French
	añejo	
vecchio	viejo	vieux
anziano	anciano	âgé
	mayor	
	antiguo	ancien
antico		antique

Here too, in order to save atomism and the nomological relation between concepts and world that goes with it, one should explain why it is that Italian, Spanish, and French minds resonate differently with age and, this, of course, by making reference only to the relation between individual concepts and the state of affairs in the world: differential or oppositional explanations related to the semantic field are not allowed by atomism.

At the origin of these problems there is, among other things, a certain confusion that computational ontologists have been known to make between signification and *designation*: the general idea in ontology seems to be that A means B if and only if A designates B. It is important however to keep the distinction between the two and, for this, I will just consider a famous example from Husserl ([9], p. 47): *the winner at Jena/the loser at Waterloo*. We notice that the *meaning* of tehse two phrases is different, although their *designatum* is the same: Napoleon.

Designation is a relation between a linguistic plane and an extra-linguistic one, but signification is a purely linguistic relation. That is, *pace* Fodor, meaning is not a nomological relation between mind and world, but a collection of relations and oppositions within the language.

* * *

These facts point quite strongly away from atomism that is, away from the (admittedly naïve) hypothesis that a symbol in an ontonomy possesses meaning *qua* symbol, by virtue of its name alone, without reference to the other elements of the ontonomy[9].

We are thus led to considering the second hypothesis given in the opening of this section, namely that the relations that one finds in an ontonomy are *constitutive* of meaning. Consider, for instance, the following ontonomy:

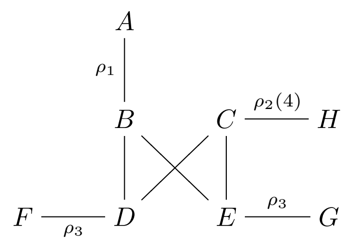
$$\begin{aligned} \text{car} &\sqsubseteq \text{motorvehicle} \sqcap \text{roadvehicle} \sqcap \exists\text{size.small} \\ \text{pickup} &\sqsubseteq \text{motorvehicle} \sqcap \text{roadvehicle} \sqcap \exists\text{size.big} \\ \text{motorvehicle} &\sqsubseteq \exists\text{uses.gasoline} \\ \text{roadvehicle} &\sqsubseteq \exists^4\text{has.wheels} \end{aligned} \tag{4}$$

[9] This is not conceptual atomism á la Fodor, which is all but naïve, but the naïve interpretation that of it has been given in computing science.

The meaning of the word ”car” is not given here by the juxtaposition of the three letters /c/, /a/, and /r/, but by its (structural) relation with the terms ”motorvehicle,” ”roadvehicle,” ”size,” and ”small,” together with the relation of these terms with other terms and so on. In other words, one can say that the meaning of the word ”car” is given by the following structure

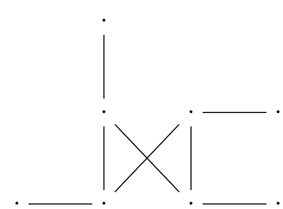
$$\begin{aligned} D &\sqsubseteq B \sqcap C \sqcap \exists \rho_3.F \\ E &\sqsubseteq B \sqcap C \sqcap \exists \rho_3.G \\ B &\sqsubseteq \exists \rho_1.A \\ C &\sqsubseteq \exists^4 \rho_2.H \end{aligned} \tag{5}$$

which we can represent by the following diagram



(6)

The meaning of the word “car” is to be found in the *structure* of this definition that is, essentially, in the following diagram:



(7)

Replacing D with ”car” in (6), we obtain the structural meaning of the concept CAR. (Here I am using capitalization to denote concepts and words in quotes to denote linguistic entities; a Saussurean semiotician would say that ”car” is a signifier and CAR the corresponding signified.)

The problem with the position that the structure (6) is the meaning of the word ”car” (that is, to be completely clear, that (6) *is* CAR) comes from the following structure:

$$\begin{aligned} \text{dog} &\sqsubseteq \text{animal} \sqcap \text{quadruped} \sqcap \exists \text{size.small} \\ \text{horse} &\sqsubseteq \text{animal} \sqcap \text{quadruped} \sqcap \exists \text{size.big} \\ \text{animal} &\sqsubseteq \exists \text{ingests.food} \\ \text{quadruped} &\sqsubseteq \exists^4 \text{has.leg} \end{aligned} \tag{8}$$

which is isomorphic to (6), that is, to CAR. Unless one is ready to concede that CAR = DOG (and I expect quite a few people to object to this identification on ground of affection either toward their poodle or toward their BMW), one must admit that there is something wrong in our definition.

The structural definition of meaning can be saved, in this circumstance, by noticing that quadrupeds are animals, while road vehicles are not necessarily motor vehicles (a horse-drawn cart, a small omnibus, or one of those four wheels bicycles that are often rented out at seaside resorts are examples of road vehicles with four wheels but no engine) so, in (8) we can affirm

$$\text{quadruped} \sqsubseteq \text{animal} \tag{9}$$

and change the first two relations to

$$\text{dog} \sqsubseteq \text{quadruped} \sqcap \exists \text{size.small} \tag{10}$$

$$\text{horse} \sqsubseteq \text{quadruped} \sqcap \exists \text{size.big} \tag{11}$$

If this new structure is still not enough to differentiate between different concepts, we can add more predicates. The question is: when can we stop? The answer is that we can't: if meaning is in the structure (and we have already ruled out the hypothesis that meaning is in the symbols themselves), then the meaning of a sign is given by the *trace* on it of all the other signs of the language, and no part of the system can self-sustain once detached from the whole.

* * *

But even an hypothetical (and impossible) macrostructure containing the whole language and its lexical relations would not be sufficient to save the semantic programme of ontology: the ontological meaning would still be normative, the codification of an author's intention at the time the text was written, and would omit the essential active rôle of the reader in the construction of meaning. In order to maintain the possibility of recording meaning once and for all, ontology must anchor it to a pre-linguistic intentional act of the author: reconstructing meaning means reconstructing this intentional status. The only way in which ontology can keep a stable meaning is by constant policing and an authoritarian normativism that sets, once and for all, the 'true' intentions of the author. There is no social participation in this construction of meaning: the reader can be replaced by an algorithm. To the Barthesian death of the author, ontology opposes a drastic "death of the reader." The underlying philosophy here is that of signification as a market, an old pre-structuralist view associated with the bourgeois individualism: meaning belongs to the author like a commodity, and language is just the currency that allows one to exchange this meaning-commodity with someone who is also an owner of meaning. But it is unlikely that such a privately owned, pre-linguistic intentional act may exist: meaning arises in language, and it is a product of a shared system of signification. What can be articulated and understood depends on this shared code. One can no longer see an intentional act that pre-dates language and that language simply reflects (the essentialist view of the ontologists notwithstanding). Reality, and the writing subject, are the product of language. And just as language interacts with other social and cultural systems, so does the act of reading. It is reading—historically and conceptually situated—that constructs meaning connecting the cues that the text gives with the complex network of conventions, discourses, and situatedness in which it occurs.

Consider a sign on a door that says "trespassers will be prosecuted." The context necessary to understand this sign is considerable. I must understand, for instance, that this sign is not informative in the sense that a newspaper headline is: I am not being informed that there have been trespassers somewhere and that they will be prosecuted sometime in the future: in western societies at least, information of this kind is not written on signs hanging from doors, especially if we see that the sign is made of plastic or wood (and therefore is durable) and the writing is not dated. Such a sign typically is a threat, the word "trespasser" refers to me (the reader) in case I decide to walk through the door, and it threatens me of prosecution if I do so. The threat also implies that prosecution is likely to result in punishment. I must understand that trespassing in this context means to cross *this* door, not some door in the palace of the king of Siam. I have to have a general knowledge of private property to understand that preventing people from entering into a building is one of the rights that society grants to proprietors (while, for instance, preventing people from looking at the building is in general not such a right), that there are authorities that will guarantee the respect of these rights, and that they will punish people who infringe these rights, that the sign has been placed there with their tacit approval, and so on...

None of these elements, necessary for understanding, is *in* the text: they must be supplied by a specific situation. The text here takes meaning by being situated (viz. placed in a situation: a door on a building rather than, say, a shelf on a store that sells signs) and in a certain relation with other texts that are not present, namely the political discourse that regulate private property, the speech through which certain customs have been implanted in the reader, and so on. Finally, all this linguistic discourse rests on a substratum of human practices and action: the political relation of power between authority and citizens, and the fact that in order to understand punishment one must understand pain (psychological pain, at least).

There is more to meaning, in other words, than just relations between terms: the creation of signification is a back-and-forth process between the text and the reader; the reader, influenced by the text, creates a frame of reference in which the text itself can be given meaning. This is what Gadamer [4] called the *hermeneutic circle:* the parts of the text can be understood in terms of the whole context, and the context becomes intelligible by means of the parts.

Ontology is trying to break this circle by removing the reader from it: it removes the creative act of the reader and tries to encode the *essence* of the meaning in such a way that it can be read without interpretation.

In this, it falls into the trap of believing that a text is just an author's intended meaning, and that therefore it is possible to re-code the text leaving the meaning unaltered. But if the meaning arises through an historically situated interaction of the reader with the text, then the text is the only possible closure of the hermeneutic circle, the only possible representation of itself, and changing the code will change the meaning. There is, in other words, no objective, essential or immutable meaning that can be encoded, either through ontological or other means in such a way that its interpretation will not require the active, culturally and historically situated, participation of the reader.

4 The pragmatics of ontology

The previous section suggested quite decisively and conclusively that the abmitious semantic programme of ontology is unattainable. Yet, as we all know, ontonomies are eagerly sought after, to the point of being hailed as a cornerstone of the constituenda semantic web. It is interesting to question why this is the case. A full answer to this question would require investigating in some depth the sociology of the computing enterprise and its relation with other economic forces, and analysis that is in the absolutest terms beyond the scope of this paper. I will venture only the briefest of comments.

The view of meaning as a commodity and of signification as a market transaction fit quite well, of course, with certain commercial aspects of the web, and it is no surprise that, in an age in which intellectual property rights are expanded to levels never dreamt before, a view of signification that allows the commodification of meaning (and thus, potentially, its patentability) should be regarded with more than a passing interest in the semantic web area. It is, in other words, not a surprise that the semantic web is these days probably the more fertile ground for the application of ontology.

It is also not surprising that the idea that meaning is to be sought in a series of taxonomies (naïve as it might be) should have arisen in an environment close to the programming profession, where taxonomies have been popularized as a programming discipline by object oriented methods: a lot of the ontological vocabulary, especially in the vicinity of the semantic web, shows a definite debt to that of the programming profession. Of course, the wide adoption of a taxonomy in a certain discipline tends to confine the discourse around certain terms an to establish an orthodoxy which might stifle alternative discourses: whether this is useful depends on the status of the discipline. The taxonomy of Linneus was a great help to the relatively mature discipline of zoology, but the taxonomization of all the elements into air, water, fire, and earth contributed to the failure of the Greek to develop a science of nature. Given the pervasiveness of computers, and the social pressure to use them, the terms and taxonomies that they impose tend to become strong norms. By forcing computerized data bases, normative semantics, and taxonomies on a vital but not yet settled discipline we might take away its vitality more than help it.

5 Conclusions

Semantics (in the sense of information systems semantics) seems to have become a most powerful "buzz-word" in the computing profession (a profession that is becoming, alas, very responsive to hype and word clout), and ontology seems occasionally to ascend to the status of a panacea. But, from within the point of view of computing science, there are at least two serious problems with this panacea.

The first is that, while we appear eager to use ontonomies in the most diverse applications, we are quite unable to define them with precision. More than a specific ontological problem, this issue is a symptom of a preoccupying relaxation of the standards of rigor of computing and, as such, should not be taken lightly, even from researchers not directly connected with ontology.

The second problem is in the idea of semantics to which ontology makes reference, an idea that is clearly insufficient in the light of all that is known about the process of signification. In a sense, the choice of such a model of signification reveals a certain cultural autism, so to speak, of computing science, and a certain historical arrogance. The problem of signification has been studied at least since the debate between the Stoics and the Epicureans on the nature of the sign, has been an important concern in medieval philosophy, and has been absolutely central in the philosophy of the XX century. But, faced with the problem of signification, computing scientists chose to disregard all this and to start from zero.

This problem is not specific to computing science, but it appears to be fairly common among technologists: "High tech, in fact, appears not only as optimistic about the future but also more indifferent toward and, in other contexts, more manipulative of the past than earlier technologies have been"[10]. Computing science is falling here in the very typical technological fallacy of considering history—cultural history, in this case, as "irrelevant save as the supposed contrast with the golden age ahead. It is as if high tech arrived and flourished in an historical vacuum of no more than a few decades and as if everything before it can simply be forgotten"[11]. But, of course, to think this way is illusory. The problem of signification and the viable relations between a syntactic manipulation device such as a computer and semantics are much more complex than the simple schematism of ontology would imply.

Given the importance that the presence of computing devices has in the disciplines in which they are used; given the influence that these devices have in promoting or constraining certain discourses in these disciplines, the computing profession has the responsibility of rejecting facile solutions. It has the responsibility of understanding the different perspectives on semantics, of being aware of the history of such problem and, ultimately, to re-analyze the relations between computers and the process of signification. Computers are syntactic machines, and it not immediately obvious that they can be of any help in dealing with semantics. The problem is worth exploring, but in a more complex way, without prejudices: the first question should not be *how* to we use a computer to represent semantics, but *whether* we should do so. We computing professional must have the acumen to discover the answer, and the cultural humility to accept it, whatever it might be.

References

1. T. Bench-Capon and G. Malcom. Formalizing ontologies and their relations. In T. Bench-Capon, G. Soda, and M. Toja, editors, *Proceedings of DEXA 99*, pages 250–259, 1999.
2. D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. Patel-Schneider. Oil: an ontology infrastructure for the semantic web. *IEEE intelligent systems*, March-April 2001.
3. Jerry Fodor. *Concepts*. Oxford:Oxford University Press, 1997.
4. H. G. Gadamer. *Truth and Method*. London, 1975.

[10] [11], p. 177
[11] ibid. p. 187

5. Horst Geckeler. *Semántica estructural y teoría del campo léxico*. Madrid:Gredos, 1976. Spanish translation of *Strukturelle semantik und wortfeldtheorie* by Marcos Martínez Hernández.
6. J. Goguen and J. Meseguer. Order;sorted algebras I: equational deduction for multiple inheritance, overloading, exceptions, and p[artial operations. *Theoretical computer science*, 105(2):217–273, 1992.
7. Thomas Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal ontology in conceptual analysis and knowledge representation*. Kluwer Academic Publishers, 1993.
8. N. Guarino. Formal ontology and information systems. In *Proceedings of FOIS 98, Trento, Italy, 6–8 June*, pages 3–15. Amsterdam:IOS Press, 1998.
9. E. Husserl. *Logische Untersuchungen, Part II: Untersuschungen zur Phänomenologie und Theorie der Erkenntnis*. Halle, 1901.
10. S. Russell and P. Norvig. *Artificial inteligence, a modern approach*. Upper Saddle River:Prentice Hall, 1995.
11. H. Segal. The cultural contradictions of high tech: or the many ironies of contemporary technological optimism. In Y. Ezrani, E. Mendelsohn, and H. Segal, editors, *Technology. pessimism, and postmodernism*. Amhrest:University of Massachusetts Press, 1994.
12. H. Wache, T. Vöegle, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information–a survey of existing approaches. In *Proceedings of IJCAI/01 Workshop: ontologies and information sharing*, pages 108–117, 2001.