Systems Science : the Ghost in the Ontologies ?

Jacky LEGRAND
Université Paris 2
legrand@u-paris2.fr

Résumé : Le but est de répondre, grâce au regroupement de réalisations pratiques et de formalisations théoriques, à la question : "quelle part de représentation d'un système peut-on trouver dans une ontologie ?".

La construction des ontologies en informatique a clairement réduit les fondements philosophiques à une vaste entreprise pragmatique. Les définitions et les pratiques (concernant les ontologies) qui en sont issues mettent en valeur un paradoxe : on semble contraint de choisir entre pure logique de description et bric-à-brac. Notre thèse est qu'il s'agit d'une *difficulté à expliciter* les systèmes dans les langages offerts par la représentation des connaissances (à ne pas confondre avec l'impossibilité de formaliser ou avec l'explicitation des processus).

Pour étudier cette "difficulté à expliciter" on confrontera l'arborescence avec la notion de hiérarchie, les propriétés avec leurs différents usages, les bases de données avec les modèles d'inspiration "associationistes". A chaque fois que la syntaxe se préoccupe un peu plus du sens "l'esprit" systémique est présent.

Abstract : An overview of historic and panoramic foundations in the field of conceptualizations questions the semantic models about the presence of implicit systemic features.

Loosing much of the original philosophical meaning, computer science turns ontologies into a vast program of description of the concepts of particular domains as well as the relationships in which they are involved. In practice the results (concerning ontologies) reveal a paradox : reliable work confines to pure logic of description while more expressive work hides a mess. Our thesis is that it is about a difficulty of formalizing systems in the languages offered by knowledge representation (not to be confused with impossibility of formalizing or the clarification of the processes).

In order to study the "explicitation pitfall" the inverted-trees are drawn closer to hierarchies, the *is-a* and *part-of* relationships are linked with the finding of the objects characteristics, the databases and conceptual modeling are investigated. The key point is that any modeling (even static and structural) in search for meaning drives back to systems science.

Keywords : Conceptual modeling, Knowledge representation, Ontology, Semantic model, Systems science, Terminological formalism,

1. Introduction

Our "ghost" does not introduce a philosophical or metaphysical talk fully in line with Derrida's writings about scholars [1]. The suggestive image stands neither for past and specter notions nor for invasive things. We suggest a haunting influence which is neither being nor non-being. That is the height of difficulty during the study of being! [2]

Modestly and cunningly, we address those considering that ontologies represent "dead" concepts.

Semantic models came to be used to deal with problems ranging from database design to retrieval and interchange of an exploding amount of on-line information, over different goals of Artificial Intelligence where the researchers supported the same ontological constructs.

Subsequently, ontologies are given as a consensual description for knowledge sharing.

([1]) We are hinting at Jacques Derrida, (1993), *Spectres de Marx*, Galilée, Paris. trans. Peggy Kamuf, *Specters of Marx*, (1994), Routledge.

([2]) Etymological and sometimes pretentious appellation for ontology in computer science. The novice reader is referred to the following 1999 issue of IEEE Intelligent Systems magazine for a clear and a short survey together with its bibliographic links and references : B. Chandrasekaran, John R. Josephson, V. Richard Benjamins, (1999), *What Are Ontologies, and Why Do We Need Them?*, Vol. 14, No. 1, pp. 20-26.

Loosing much of the original philosophical meaning, computer science turns ontologies in a vast program of description of the concepts of particular domains as well as the relationships in which they are involved. However, the same concept may be defined in different ways (naming problems or conflicts between modeling constructs) while a common vocabulary may hide instances heterogeneity.

As any trend in the spotlight, the branch of research goes through definition difficulties, indeed even overstatements including some re-labeling of usual software engineering activities.

We have no room here to review a decade of ontologies development. We choose to address the concern that has always been at the heart of formal ontology development.

Our interest in conceptual maintenance and ontologies alignment stresses, on the one hand, the intuitive aspect of the epistemological primitives of description and, on the other hand, the impact of the systemic features. It stands to reason since only cognitive science and systems science can give an ontological dimension to the data structures.

We already drew the colleagues' attention to the topic [3]. Ongoing research in formal ontology [4] offers approaches demonstrating that the traditional semantic modeling methods (resting on intuitive primitive description features) may have reached their limitations.

Our main thesis is that any modeling (even static and structural) in search for meaning drives back to systems science. However, the usual work focuses on how to go about building ontologies and undertakes an ontology of ontological constructs without being aware of systemic developments. To derive benefits from the "ghost" the two communities have to get closer, or, at least, the vision systems science reflects must become pervasive.

2. Methodology

In spite of some disagreements, the Gruber [5] definition for ontology is widely cited in the literature : "An ontology [6] is an explicit specification of a conceptualization".

In spite of this statistical consensus, ontologies are widespread in too many areas of information science. A reliable study makes necessary to point out different views about ontologies.

It can be said that an ontology is a "shared and formal conceptual modeling of a domain" in order to be consensual, but naming-centric ontologies remains far from agents-centric ontologies. The first category is close to a user-oriented formalism while the second one faces inference, task, and strategy challenges.

In order to investigate the different facets of the classic ontologies shortcomings, the work

---

([3]) Since 1987 with our first paper on the subject Jacky Legrand, (1987), *Contribution to the Achievement of a Syntax in Knowledge Engineering*, Expert Systems IASTED Conference, Geneva, Proceedings, Acta Press, Calgary. and for the systems science community in Jacky Legrand, (1996), *A Contribution to Conceptual Level Maintenance in Sofware Engineering*, Third European Congress on Systems Science, Proceedings, Edizioni Kappa, Roma.

([4]) We refer Laboratory for Applied Ontology (http://www.loa-cnr.it/). The publications of Nicola Guarino and al. are, from our perspective, the most exiting development over the past few years to drive ontology development from a craft to a science.

([5]) Thomas R. Gruber , (1993), *Toward Principles for the Design of Ontologies Used for Knowledge Sharing* in Formal Ontology in Conceptual Analysis and Knowledge Representation, Nicola Guarino and Roberto Poli (Editors), Kluwer Academic Publishers.

([6]) This definition supports the vocabulary shift which licenses computer scientists to use "ontology-ontologies" instead of "semantic modeling products based on an ontological approach". The reader must bear in mind the whole paper uses the license for short.

have to find the different epistemology-efficiency trades off for different communities.

While the amount of on-line information is exploding, web searching facing new challenges, focuses on developing more "semantic means" to access these collections. Without attempting to reason about the documents content, a domain-specific knowledge representation is used to enhance semantic retrieval. Terminology and interchange are the two motivations for commercial creating while evolution, merging and alignment of ontologies are the motivations for fundamental research.

Knowledge sharing is not necessarily linked to an implementation of reasoning. In the field of Artificial Intelligence, ontological engineering [7] is a process that facilitates construction of the knowledge base. To support inferencing, the tradeoff is an extension of the classic terminological-assertional one. What is already described in the structure has not to be written as rules. It interferes with the pitfall of the expressiveness-tractability tradeoff (the more that can be left unsaid, the more it is difficult to compute the entailments).

To model in a way one can understand is the motivation of symbolic Artificial Intelligence to capture the full spectrum of human knowledge, while a well-defined semantics which does enable logic-based deductions is the motivation for further ontological searching.

Software engineering community (and by the way object-oriented programming languages community) manages events, interactions and processes. The software engineering methodologies and the associated industrial tools deliver formalisms to address pre- and post-conditions of actions. However, the information system being elaborated (merely process steps) has not to be confused with the system being modeled. With other words, surprisingly, the extension of the classic algorithm-structure tradeoff (what is stored is not computed) is a "parsimony rule" : how the syntactic form of knowledge allows reuse of complex definitions (absence of redundancy).

Databases community considers explicitly [8] or implicitly the ontological level [9] as capping the knowledge level [10]. The ontologies have many similarities with the conceptual modeling.

Developing the action schemes is the motivation for behavioral modeling while expliciting the dynamic knowledge embedded in the conceptual schemes to optimize computing is the motivation for "semantic relationships" investigations.

While it remains unclear how best to carve the world at its joints [11], the specific tasks in mind (inferences, queries, ...) point out the crucial need for explicit assumptions regarding the *intended meaning* of the formal vocabulary [12]. Any metamodeling trend is of interest

---

(7) From 1989 in the first issue of Knowledge and Data Engineering IEEE Transactions, Douglas B. Lenat's paper [Douglas B. Lenat, (1989), *Ontological Versus Knowledge Engineering*, Knowledge and Data Engineering IEEE Transactions, Vol. 1, Issue 1, p. 84-88] opens the realm of ontological engineering. The Cyc program : [Douglas B. Lenat and FINIR R.V. Guha, (1990), *BuiIding Large Knowiedge-Based Systems*, FINIR Reading, Mass.: Addison-Wesley] is the well-known corresponding application.

(8)Yair Wand, Veda C. Storey, Ron Weber, (1999), *An ontological analysis of the relationship construct in conceptual modeling*, ACM Transactions on Database Systems, Vol. 24 , Issue 4, pp. 494-528.

(9) Nicola Guarino, (1994), *The Ontological level*, *in* R. Casi, B. Smith and G. White (Editors), Philosophy and the Cognitive Science. Holder-Pichler-Tempsky, Vienna.

(10) Newell's "Knowledge-Level" AAAI Presidential Address at Stanford (August 19th, 1980) printed in AI Magazine Vol. 2, Issue 2, 1980. See also Allen Newell, (1982), *The knowledge level*, Artificial Intelligence Journal, Vol. 18, Issue 1 , pp. 87-127, 2(2): Summer 1981, 1-20, 33

(11) Plato, Phaedrus [Socrates 265e] : "That of dividing things again by classes, where the natural joints are, and not trying to break any part, after the manner of a bad carver." (Perseus digital library translation)

(12) Nicola Guarino, (1998), *Formal Ontology in Information Systems*, Proceedings of FOIS'98, Trento, Italy, 6-8 June. Amended version of a paper appeared in Nicola Guarino, (1998) *Formal Ontology and Information Systems* in: *Formal Ontology in Information Systems*, Nicola Gaurino (Editor), IOS Press, Amsterdam,. pp. 3-15.

whatever it originates in strict ontological cleaning or in actual implementation worry.

The remainder of this paper follows the quest of meaning in the syntactical representations as the Ariadne's thread. It is noteworthy that we face a dilemma : how to argue for precision and to survey many branches offering a heterogeneous vocabulary [13]. We close the matter by a contrastive discourse. We do not discuss the strengths and weaknesses of different kinds of representations but we enlighten the points they brought out. For example, questioning the concept of "chair" in taxonomy (location, product, furniture, ..) is a way to discover that, in the systems view, relationships are ontologically prior to things (irrespective of the syntactical choices for dealing with "chair" in a particular enterprise).

Section 3 starts from the basic and elementary construct, the hierarchy. This step stresses the miscellaneous semantic interpretations needed for the inverted-tree structures. Section 4 goes on through associational modeling and investigates general relationships.

3. Basic Abstractions

3.1 Hierarchy and inverted-trees structure

Hierarchical organization of data is ubiquitous. The aim of this paper is not to discuss whether it comes from the nature of the world or from the way humans organize their thoughts to survive or whether the inverted-trees are only a convenient data structure for programming [14].

The word "ontology" stands for semantic models with different degrees of structure but many existing ontologies exhibit a hierarchy in order to classify concepts. However, many ontologies available today were not constructed with precise definitions in mind and remains partially intuitive The complexity of the word "hierarchy" has been swept under the carpet.

The point is that the inverted-trees are syntax. A great diversity of knowledge may be fitted into since hierarchy is defined in mathematical terms (a partially ordered set or a restricted kind of graphs). However, the ontological commitment of a knowledge representation, as much as the software use of inheritance [15], require having precise ideas in mind.

The confusion when hierarchical links remain untyped is far from new. Old hierarchical databases exhibit structures as "Suppliers being parents of Parts-manufactured". It is more intriguing when modern conversion to XML [16] is an excuse to exhibit "Regions being parents of Departments being parents of Trips being parents of Car-rental being parents of People". Inverted-trees may mix together "is-a", "part-of", "has-a" and many more.

A lot of work has been done concerning the *is-a* (generalization) and *part-of* (aggregation) links creating the most common hierarchies. Unfortunately, in some cases the two relationship types are not clearly distinguished and, moreover, when examined in some detail, a systemic perspective points out different kinds of generalization or aggregation.

([13]) A discussion about the differences between
concepts/collections/classes /entity-types,
instances/individuals/objects/entities,
characteristics/properties/attributes/template-slots/roles,
may motivate a whole paper (indeed a whole conference).

([14]) We refer to the idea according to the general theory of complex systems : complexity naturally takes the form of hierarchy. Herbert A. Simon, (1969), *The Sciences of the Artificial*, MIT Press, Cambridge, Massachusetts.

([15]) Inheritance is intuitively a process where the specific (refining) characteristics are supplied while the more general ones may be "copied".

([16]) XML (eXtensible Markup Language) is a standard for data interchange in the Internet world.

Before discussing the problematic usage of *is-a* and *part-of*, one choose to exhibit another type of hierarchy. Let's consider the case where the inclusion relation is considered among different partitions (for example a container-oriented structure that describes a location at a specific position [17]). The abstract mathematical structure joined to the different semantical interpretations of the partitions introduces a lot of confusion.

The "container" relationship among elements may express the hierarchical structure of document elements. Although being a "weak whole", the structure of the document seems to be a taxonomy of *part-of*. Less abstract, a district is *part of* a town just as a sub-sub-section is *part-of* a sub-section. But the member of a partition is sometimes reifed as a classification concept. The "species" are no more *part-of* "genres" than one eagle *is-a* species [18].

The members of a partition are often used as values of properties for objects classification, the partition is a property type. This is why taxonomy of properties [19] clears up the mess in a better way than linguistic considerations about naming. "The problem is that linguistic hyperonymy is not a "pure" *is-a* relation. (p. 21)" [20].

We argue that describing and classifying properties is a way to question the systemic statute of the reifed elements of `P(E)`, `P(P(E))` where `E` stands for Extension and `P` for Power set.

But other ways exist to catch a glimpse of the ghost. The first one comes within Software Engineering.

Notwithstanding the constraint of efficient implementation in Object Oriented Programming, most of the intuitions about various types of *is-a* hierarchies have not yet delivered a consensus into precise definitions. The reason may be that inheritance has a role beyond attributes propagating : behavioral inheritance. In spite of ad hoc operational solutions much research is still needed.

Harel & Kupferman [21] notice (in their words) that major portion of approaches to inheritance in the software literature refers to a structural *is-a* relationship. It says little about the behavioral conformity of classes and subclasses. They add that full behavioral conformity between a type and its subtype is technically very difficult, so most modelers do not expect that subtype can do anything type can do and in the very same way.

The second one comes within the opposition of bottom-up and bottom-down semantic modeling. The bottom-down approach suggests classes before populating with instances. The bottom-up approach considers instances before classifying [22].

The definition of generalization as regarding sets of similar objects is the cause of many further studies. The word "set" allows the grouping of objects together and the shift to *is-*

---

(17) Hanan Samet, The Quadtree and Related Hierarchical Data Structures, (1984), ACM Computing Surveys, Vol. 16, Issue 2, pp. 187–260.

(18) Christopher A. Welty, David A. Ferrucci, (1999), *Instances and Classes in Software Engineering*, Intelligence Magazine, ACM, Vol. 10, Issue 2, pp. 24-28.

(19) Nicola Guarino and Christopher Welty, (2000), *A Formal Ontology of Properties*, Proceedings of 12th Int. Conf. on Knowledge Engineering and Knowledge Management, Lecture Notes on Computer Science, Rose Dieng (Editor), Springer Verlag.

(20) Violaine Prince, Mathieu Lafourcade, (2003), *Mixing Semantic Networks and Conceptual Vectors: the Case of Hyperonymy*, IEEE Proceedings of the 2nd International Conf. on Cognitive Informatics (ICCI'03), pp. 121-128

(21) David Harel, Orna Kupferman, (2002), *On Object Systems and Behavioral Inheritance*, IEEE Transactions On Software Engineering, Vol. 28, Issue 9, pp. 889-903

(22) Jeffrey Parsons, Yair Wand, (2000), *Emancipating Instances from the Tyranny of Classes*, ACM Transactions on Database Systems, Vol. 25, Issue 2,, pp. 228-268.

member-of relationship type. Some authors [23] stress the difference between grouping and conceptual abstraction in types. However, the pitfalls are numerous. An example can be given with "Customer" being a subset of the instances of "Person" (a customer is specified from the general person) and "Person" being a subset of the instances of "Customer" (the customers may be specified as humans or as firms). When the concepts ceased to be "self-standing", existing outside time and space, the notion of role [24] shows that the exegeses of *is-a* relationship type have reached their limitations.

Nevertheless, the contributions of the notion of role will be discussed in a section 4. Since it cannot be done without introducing, complex objects and, by the way, wholes and components. The *part-of* relationship type will now be investigated by considering the ambiguous intermix between part-whole modeling and aggregation abstraction.

3.2 Object characteristics and aggregation of attributes

The relational model (introduced by Codd) may be seen either as the primary commercial record-based data processing application or as rigorous mathematical database foundations but, in no way, as a knowledge representation. A *relation* (relational table) is a subset of a Cartesian product and has not to be confused with a relationship (understood as an associational link between things).

According to the notation we introduced in the previous paragraph, a row in a table stands for an instance of the real world and is $\in$ E. A cell in a table is $\in P(E)$. An attribute is $\in P(P(E))$. A relation is $\in P(P(P(E)))$. Being now aware of the great diversity for $P(E)$ (property), $P(P(E))$ (property-type), $P(P(P(E)))$ is neither a set or a system but a hotchpotch [25].

Several years after Codd's initial publication [26] many modeling methodologies attempt to provide the necessary semantic content [27]. One of the first attempts is the paper by Smith and Smith [28] in which abstractions (generalization and aggregation) were employed for database modeling. Generalization has been discussed in the previous paragraph. Aggregation, "refers to an abstraction in which a relationship between objects is regarded as a higher level object. (p. 106)", and suggests an orthogonal hierarchy. Unfortunately, the nature and underlying meaning of "relationship" is quite unclear.

A fundamental consequence is to show that the aggregation of the attributes (as used in the Codd relational approach) or aggregation of the slots (as in frames [29]) is not orthogonal to

---

(23) Elke A. Rundensteiner, Lubomir Bit, (1992), *Set Operations in Object-Based Data Models*, IEEE Transactions On Knowledge and Data Engineering, Vol. 4, Issue 3, pp. 382-398.

(24) Friedrich Steimann, (2000), *On the representation of roles in object-oriented and conceptual modelling*, Data & Knowledge Engineering Journal, Elsevier, Vol. 35, Issue 1, pp. 83-106.

(25) During this step of abstraction, the distinction between subsets and systems vanishes. The normalization of the database cannot argue to restore any clean ontology.

(26) Edgar F. Codd, (1970), *A relational model of data for large shared data banks*, Communications of the ACM Vol. 13, Issue 6, pp. 377-387.

(27) For a survey that samples a decade of "semantic" data model, see Richard Hull, Roger King, (1987), *Semantic database modeling: survey, applications, and research issues*, ACM Computing Surveys, Vol. 19, Issue 3, pp. 201-260. Joan Peckham, Fred Maryanski, (1988), *Semantic Data Models*, ACM Computing Surveys, Vol. 20, Issue 3, pp. 153-189.

(28) John Miles Smith, Diane C. P. Smith, (1977), *Database abstractions: aggregation and generalization*, ACM Transactions on Database Systems, Vol. 2 , Issue 2, pp. 105-133.

(29) Thus named with reference to the seminal publication of Marvin Minsky (MIT-AI Laboratory Memo 306, June, 1974) edited as Marwin Minsky, (1975) : *A framework for representing knowledge*, The psychology of computer vision, Mac Graw Hill, New York, London, Paris.

generalization. It remains an abstract or *ad hoc* n-ary notation and the *part-of* link may be suspect from a systemic point of view.

As Motsschnig-Pitrik & Kaasb¢ll [30] notice, the fact that the term aggregation may be used synonymously with part composition or relationship is definitely incorrect for real-world objects.

To escape the trap two main directions have been followed : the part-whole study and the relationships construct enrichment.

Fortunately, many expressive notations have been developed, crossroads of networks and frames, providing an object–centered associational representation with a natural graphical form.

4. Semantic relationships

4.1 Conceptual modeling and holes

The most widely used semantic data models is the Entity-Relationship Model [31] (and its various extensions [32]). Relationships constructs are offered to the modeler in order to capture the meaning of the associations among objects

The main problem of such semantic models is the semantic relativism due to the equivalence among the constructs of the model. For example, to be French might be interpreted as to be inserted into the "French persons" type or as valuing the "citizen" attribute or as to be involved in the "lives in" relationship. Usually, the application goals decide in databases view form.

We suggested in our previous theoretical publications [33] that a *word of typification* might be added to each component of the conceptual model. This strategy consists in superimposing a syntactic construction. This addition clarifies what is already present and points out that components and links are missing in such ontological engineering. The holes explain the computational intractability of knowledge transformations.

For example, let's consider the structure of a "student" class defined by the properties "name", "age", "college", and "vehicle" [34] and populate the attribute "vehicle" with "bike". The hole contains the fact that a bike is a transport included in a hierarchy of moves. And what to say about the next-door "college" not requiring a "car" or the "age" under driving license limit ?

This is exactly what Lenat [35] points out when giving a typical example of "how to solve a problem dishonestly" with the use of very detailed, complex predicates (such as "LaysEggsInWater") without defining them.

---

(30) Renate Motschnig-Pitrik, Jens Kaasb¢ll, (1999), *Part-Whole Relationship Categories and Their Application in Object-Oriented Analysis*, IEEE Trans. on Knowledge and Data Engineering, Vol. 11, Issue 5, pp. 779-787.

(31) Entity-Relationship seminal paper is : Peter Pin-Shan Chen, (1976), *The Entity-Relationship model : toward a unified view of data*, ACM Transactions on Database Systems, Vol. 1, Issue 1, pp. 9-36.

(32) See for example : Toby J. Teorey, Dongqing Yang, James P. Fry, (1986) *A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model*, ACM Computing Surveys, Vol. 18, Issue 2, pp. 197-222.

(33) We provide a self-contained introduction to the notion of species of structure (defined in Bourbaki). For the bibliographic references see note number (see note number 3)

(34) Anna Formica, Michele Missikoff, (2004), *Inheritance Processing and Conflicts in Structural Generalization Hierarchies*, ACM Computing Survey Vol. 36, Issue 3, pp. 263–290.

(35) See note number 3 for bibliographic reference.

Moreover, the progressive elaboration of ontology (colleges and vehicles may be introduces as entities involved in relationships) with an associational methodology turns the intrinsic characteristics of the entities into links (defined in terms of the underlying physical components) and the extrinsic characteristics surprisingly form the attributes (defined in terms of the external environment).

For example, the "price" which captures, beyond the relationship between money and goods, the notion of exchange, the existence of economy and its dependence on time and events can be "outsourced" in full.

4.2 Relationships enrichments

The relationships between the things are often described in a similar way independent of their nature [36]. Relationships are more difficult to model than entities. Much research is still needed : "In our view, problems arise with relationships in conceptual modeling because their nature and underlying meaning are unclear." [37]

Many authors make the term "part" escape its previous intuitive meaning. Firstly, whole-part relationship must be set apart from abstract aggregation [38]. Further, the intuitive idea of inclusion as a basic property for *part-of* description must be specified. The famous example [39] "a head has ears", "ears have lobes" : "has a head lobes?" show that subrelations of *part-of* can be considered transitive only as long as "a single sense of part" is kept [40]. It is now easy to understand that when the parts themselves may act as wholes to other parts the hierarchy management must be subject to particularly precise constraints.

These constraints are not clearly defined in most object-oriented modeling approaches because it cannot be said that a property is emergent (in the sense that it depends upon some part) or resultant (emergent in the sense that it directly depends upon all parts).

The *part-of* relationship embodies some aspects of existence dependency (if a whole disappear , the part still exist as object). This remark leads to set *can be a part-of* apart from *part-of.*

This kind of ontological commitment is often neglected until the application at hand requires it. For example, "non-human vertebrates also have hearts, we can state *human being has_ part heart*, but not *heart part_ for human being."* [41]

Just as for *is-a* relationship the propagation of events may not be described in the structural inheritance of properties but rather in the functional relationship to the whole.

To describe what objects do and how they interact contributes to the formalization of constraints, which restrict the behavior of the implemented relationships.

---

(36) "The relationships between the things should be modelled using the relationships allowed by the language. This implies that all the "real" relationships will be modelled in a similar way independant of their semantics." *in* José Parets-Llorca, (1993), *On some epistemological challenges of object-oriented software engineering : if objects were systems*, Second European Congress on Systems Science, Oct. 5-8 1993, p. 447.

(37) Yair Wand, Veda C. Storey, Ron Weber, (1999), *An ontological analysis of the relationship construct in conceptual modeling*, ACM Transactions on Database Systems, Vol. 24 , Issue 4.

(38) Brian Henderson-Sellers, Franck Barbier, (1999), *What is This Thing Called Aggregation?*, IEEE Proceedings of 29th Conference Technology of Object-Oriented Languages and Systems, p. 236-250.

(39) Cited by Stefan Schulz, Udo Hahn, (2001), *Mereotopological Reasoning about Parts and (W)Holes in Bio-Ontologiest*, Proceedings of FOIS'01, Ogunquit, Maine, USA, October 17-19, pp. 210-221.

(40) M. Winston, R. Chaffin, D.J. Herrmann, (1987), *A Taxonomy of Part-Whole Relationships*, Cognitive Science,Vol. 11, pp. 417–444.

(41) Barry Smitha, Cornelius Rosseb, (2004), *The Role of Foundational Relations in the Alignment of Biomedical Ontologies*, MEDINFO 2004, M. Fieschi et al. (Editors), IOS Press, Amsterdam, p.446.

The most common structural properties are : degree (number of participants), cardinality (number of instances of the participant). Operational properties may be added concerning operation propagation and existence dependency. At an ontological level it deals with interrelationship constraints (inclusion dependency, exclusion dependency, derived relationships).

It is somewhat intriguing that, notwithstanding the success of aggregation abstraction, a debate exists since the binary model of Abrial [42] about the necessity of formalization for n-ary relationships.

Although many semantic models restrict relationships to be of degree 2, the transformations of higher degrees relationships prove to be problematical [43]. The cited paper offers a good example (we have no room to paste it here) of the systemic implication in structural description. We only reproduce the conclusion of the author :

> DESIGN IMPLICATION 1. Real-world decisions are considered dissimilar if they involve different decision makers or are made during different time-frames. Two facts arising from two independent decisions should not be merged into a single relationship instance.

An answer to the binary/n-ary controversy stands the complex entities [44]. For example, the idea of embedded sub-structures protects a "hotel-reservation" from being a flat (person-hotel-room-date) aggregation or a flattened n-ary relationship.

To conclude about relationships explicitation, we come back to the notion of role. Steinmann [45] gives a list of 15 features he has identified for the roles in literature. The eighth one is : "roles can play roles". He mentions 3 common ways of representing roles. One seems to be particularly his own. The remaining two are : 1) roles as named places of a relationship, 2) roles as a form of generalization.

Roles as named places of a relationship confirm the importance of complex objects to satisfy the eighth feature. Entity-relationship modeling cannot include it.

Roles as a form of generalization are often considered as classes that may vary with time, "it emerges that the notion of role is inherently temporal (p. 88)" [46]. In fact the ontological philosophy of Bunge [47] allows the distinction between "binding" (relate an entity to other entities) and "time dependent" (relate an entity to events).

In fact roles encapsulate behavioral aspects and define contexts. They provide a convenient way to represent the teleological notion of purpose [48]. A study of the whole research about roles in semantic modeling ought to be the closest contact with the ghost.

---

(42) J.R. Abrial, (1974), *Data semantics* in Data Base Management. North-Holland, Amsterdam, pp. l-59.

(43) Debabrata Dey, Veda C. Storey, Terence M. Barron, (1999), *Improving database design through the analysis of relationships*, ACM Transactions on Database Systems, Vol. 24, issue 4, pp. 453-486

[44] Mauricio .J.V. SILVA, C.Robert CARLSON, (1995), *MOODD, a method for object-oriented database design*", Data and Knowledge Engineering, Vol. 17, Issue 2, Elsevier, pp. 159-181.

(45) Friedrich Steimann, (2000), *On the representation of roles in object-oriented and conceptual modelling*, Data & Knowledge Engineering, Vol. 35, Issue1, Elsevier, pp. 83-106.

(46) Valentina A.M. Tamma & Trevor Bench-Capon, (2001), *An Enriched Knowledge Model for Formal Ontological Analysis*, Proceedings of FOIS'01, October 17-19, Ogunquit, Maine, USA, ACM, p. 81.

(47) Mario Bunge, (1977). Treatise on Basic Philosophy: Vol. 3: Ontology I: The Furniture of the World. D. Reidel Publishing Co., Inc., New York, NY.

Mario Bunge, (1979). Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems. D. Reidel Publishing Co., Inc., New York, NY.

(48) James Fan, Ken Barker Peter Clark, Bruce Porter, (2001), *Representing Roles and Purpose*, ACM Proceedings of the International Conf. on Knowledge Capture, Victoria, British Columbia, Canada, October 22-23, pp. 38-43

5. Conclusion

In this paper we have critically inspected semantic modeling, conceptual schemes (application–specific use of the previous ones) and their relationships to clearly circumscribe the presence of invariants and the underlying heterogeneity of tasks.

Through a walk among a vast amount of researches motivated by a more and more heavy ontological commitment, we have been using a panoramic and a historical point of view to emphasize that the quest for meaning takes the static structural modeling enterprises back to the systems they deal with.

The existing research about hierarchies, inheritance, roles, whole-part relationships notwithstanding, metamodeling is still in its early stages. Our main concern in future work is the quality of the ontologies transformations [49] (to improve evolution, merging or alignment of otonlogies) since conceptual maintenance rests in the coding of the semantics of the basic constructs [50]. A suggested subject of further work in Software Engineering refers to the schema transformations in UML [51]. In short [52] the UML class diagrams provide an extended Entity-Relationship notation and a mapping onto various activity diagram types requires identifying properly which features one must know about.

During our process of clarification some new qualitative concepts have been suggested. Our strategy consists in superimposing a syntactic construction (the words of typification) and in supplementing the set of explicit assumptions (regarding the intended meaning) to preserve the global comprehension of systems, often lost during the analytical findings.

We emphasize that intellectual assignment of systemic features is part of artificial intelligence approach to formal ontology.

Relationships, in the systems view, are ontologically prior to things. The paper clarifies the implicit role systems science plays during the semantic achievements in a formal language of structure. Full of mistakes can be avoided if there is an awareness of this.

Realizing this potential synergy requires mutual understanding. We hope to succeed in helping the systems science community to understand exactly what is and is not modeled in ontologies constructs. The resulting benefit stands as much for systemic modeling as for construction of ontologies.

---

(49) In fact structure reformatting ranges from the view integration in databases to ontologies alignment.
Carlo Batini, Maurizio Lenzerini, Shamkant. B. Navathe, (1986), A comparative analysis of methodologies for database schema integration, ACM Computing Surveys, Vol. 18, Issue 4, pp. 323-364.
Alignment establishes direct mappings, adding or deleting concepts if needed. Conversely, articulation maps concepts of both ontology into concepts of a shared view.
Helmut Meisel, Ernesto Compatangelo, (2002*), EER-CONCEPTOOL: a "reasonable" environment for schema and ontology sharing*, IEEE Proceedings, International Conf. on Tools with Artificial Intelligence, pp. 527-534.

(50) It is argued that when the constructed attributes receive a systemic interpretation, it predicts the influence of structure reformatting. Unfortunately, the matter is as yet hardly explored.

(51) The Unified Modeling Language (UML) was adopted in 1997 by the Object Management Group (OMG) as a language for object oriented (OO) analysis and design.
Background on UML may be found in Hans-Erik Eriksson, Magnus Penker, (2000), *Business Modeling with UML: Business Patterns at Work*, John Wiley & Sons, New York.

(52) Petri Selonen, Kai Koskimies, Markku Sakkinen, (2001), *How to Make Apples from Oranges in UML*, IEEE Proceedings of the 34th Hawaii International Conference on System Sciences, Vol. 3, pp. 3054.