

Towards Ontological Foundations for UML Conceptual Models

Giancarlo Guizzardi¹, Heinrich Herre², Gerd Wagner³

¹ Centre for Telematics and Information Technology, Univ. of Twente
Enschede, The Netherlands
guizzard@cs.utwente.nl

² Institut für Informatik, Univ. Leipzig and
Institute for Formal Ontology and Medical Information Science (IFOMIS)
herre@informatik.uni-leipzig.de

³ Eindhoven Univ. of Technology, Faculty of Technology Management,
G.Wagner@tm.tue.nl,
<http://tmitwww.tm.tue.nl/staff/gwagner>

Abstract. UML class diagrams can be used as a language for expressing a conceptual model of a domain. We use the General Ontological Language (GOL) and its underlying upper level ontology, proposed in [1], to evaluate the ontological correctness of a conceptual UML class model and to develop guidelines for how the constructs of the UML should be used in conceptual modeling. In particular, we discuss the UML metaconcepts of classes and objects, power-types, association and aggregation/composition from an ontological point of view. We make some proposals of how to extend version 1.4 of the UML in order to obtain a more satisfactory treatment of aggregation.

1 Introduction

Conceptual modeling is concerned with identifying, analyzing and describing the essential concepts and constraints of a domain with the help of a (diagrammatic) modeling language that is based on a small set of basic meta-concepts (forming a meta-model). Ontological modeling, on the other hand, is concerned with capturing the relevant entities of a domain in an ontology of that domain using an ontology specification language that is based on a small set of basic, domain-independent ontological categories (forming an upper level ontology). While conceptual modeling languages are evaluated on the basis of their successful use in (the early phases of) information systems development, ontology specification languages and their underlying upper level ontologies have to be rooted in principled philosophical theories about what kinds of things exist and what are their basic relationships with each other.

Recently, it has been proposed that UML should be used as an Ontology Representation Language [2,3]. Moreover, in [2] it is argued that although UML lacks a precise definition of its formal semantics, this difficulty shall be overcome with the current developments made by the precise UML community [4, 5]. We believe, however, that defining UML constructs only in terms of its mathematical semantics, although essen-

tial, it is not sufficient to make it a suitable ontology representation language. The position defended here is that, in order to model reality, a conceptual modeling language should be founded on formal upper-level ontologies. In other words, it should have both, formal and ontological semantics.

In this paper we use the General Ontological Language (GOL) and its underlying upper level ontology, proposed in [1], to evaluate the ontological correctness of a conceptual UML class model and to develop guidelines that assign well-defined ontological semantics to UML constructs.

Due to space limitations we cannot examine all UML constructs here. This paper discusses, in particular, the ontological meaning of the UML metaconcepts of *classes and objects*, *power types*, *association* and *part-whole* relations (*aggregation/composition*). The UML metaconcepts of *abstract classes* and *datatypes* are addressed in a companion paper [6].

In addition we propose some extensions to version 1.4 of the UML in order to obtain a more satisfactory treatment of *part-whole* relations. Section 2 introduces the basic elements that form the upper level ontology of GOL. Section 3 uses this upper level ontology to define the real-world semantics for UML class diagrams for the purpose of conceptual modeling. Section 4 discusses some related work. Finally, section 5 presents some conclusions.

2 Basic Elements of the Upper Level Ontology of GOL

The basic elements of the upper level ontology of GOL can be visually described by means of the UML class diagram shown in Figure 1.

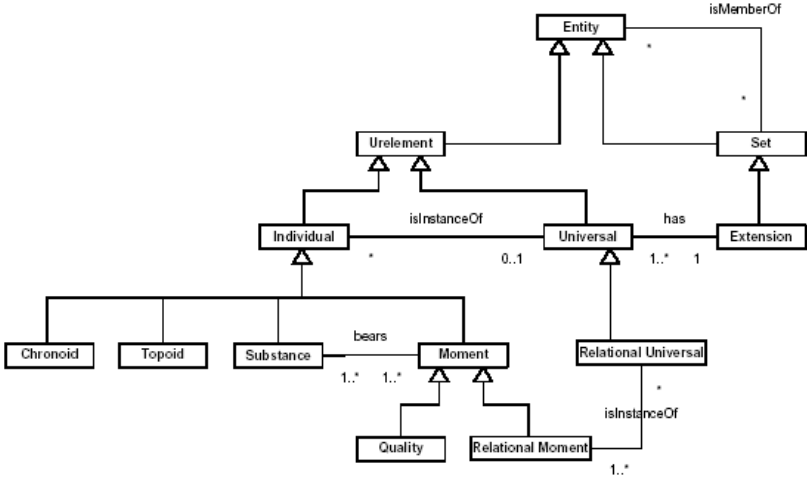


Fig. 1. A UML Class Diagram describing the basic concepts of GOL.

2.1 Urelements and Sets

The main distinction of the GOL-ontology is between *urelements* and *sets*. We assume the existence of both urelements and sets in the world and presuppose that both the impure sets and the pure sets constructed over the urelements belong to the world. This implies, in particular, that the world is closed under all set-theoretical constructions. Urelements are entities which are not sets. They form an ultimate layer of entities without any set-theoretical structure in their build-up. Neither the membership relation nor the subset relation can unfold the internal structure of urelements. In GOL, urelements are classified into two main categories: individuals and universals. There is no urelement being both an individual and a universal. This is expressed in GOL by the following axioms:

$$\forall x (\text{Ur}(x) \leftrightarrow \text{Ind}(x) \vee \text{Univ}(x)) \quad (\text{U1})$$

$$\neg \exists x (\text{Ind}(x) \wedge \text{Univ}(x)) \quad (\text{U2})$$

2.2 Individuals

Individuals may be substances, moments, chronoids, or topoids.

Substances and Moments. A *substance* is that which can exist by itself, or does not need another entity in order to exist. Typical examples of substances are: an individual person, a house, the moon, a car. Every substance is founded on matter. Substances come into existence because the matter is formed in various ways which give rise to pieces separated in more or less stable ways from their surroundings and possessing qualities of different sorts.

A *moment*⁴ is an individual which can only exist in other individuals (in the way in which, for example an electrical charge can exist only in some conductor). Typical examples of moments are: a color, a connection, a purchase order. Moments have in common that they are all dependent on substances. Some moments are one-place qualities, for example color or temperature. But there are also *relational moments* – for example flight connections or purchase orders – which depend on several substances.

The inherence relation *i* – sometimes called ontic predication – glues moments to the substances which are their bearers. For example it glues your smile to your face, or the charge in a specific conductor to the conductor itself. Substances must bear moments, and moments must inhere in substances. This is axiomatically expressed as follows:

⁴ The origin of the notion of moment lies in the theory of individual accidents developed by Aristotle in his *Metaphysics* and *Categories*. An accident is an individualized property, event or process which is not a part of the essence of a thing. We here use the term “moment” in a more general sense and do not distinguish between essential and inessential moments.

$$\text{Subst}(x) \rightarrow \exists y (\text{Mom}(y) \wedge i(y,x)) \quad (\text{SM1})$$

$$\forall x (\text{Mom}(x) \rightarrow \exists y (\text{Subst}(y) \wedge i(x,y)) \quad (\text{SM2})$$

Chronoids and Topoids. Chronoids and topoids are instances of the universals *Time* and *Space*, respectively. Chronoids can be understood as temporal durations, and topoids as spatial regions having a certain mereotopological structure. On one version of the theory chronoids and topoids have no independent existence; they depend for their existence in every case on the situations which they frame. Every substance x has a certain maximal temporal extent, a chronoid which we denote by $\text{lifetime}(x)$. The substance x exists during $\text{lifetime}(x)$. Also, every moment m inhering in x has a lifetime , which is such that $\text{lifetime}(m) \leq^5 \text{lifetime}(x)$. Moreover, if n is a relational moment connecting substances x_1, \dots, x_k , then $\text{lifetime}(n) \leq \text{lifetime}(x_i)$, for i ranging from 1 to k .

2.3 Universals

A universal is an entity that can be instantiated by a number of different individuals which are similar in some respect. Following Aristotle, we assume that the universals exist in the individuals (*in re*) but not independently from them. As a consequence, in order to exist, universals must possess instances.

For every universal U there is a set $\text{Ext}(U)$, called its extension, containing all instances of U as elements. It is, however, not the case that every set is the extension of a universal (there is no such axiom in GOL).

There are two kinds of universals that are of particular interest: **quality universals**, such as color and weight, and **relational universals**, such as flight connection ('...is connected with...') or purchase ('...purchases...from...'). Every universal has an intension which, in GOL, is captured by means of an axiomatic specification, i.e., a set of axioms that may involve a number of other universals representing its essential features. A particular form of such a specification of a universal U , called elementary specification, consists of a number of universals U_1, \dots, U_n and corresponding functional relations R_1, \dots, R_n which attach instances from the U_i to instances of U , expressed by the following axiom:

$$\forall a (a::U \rightarrow \exists e_1 \dots \exists e_n \bigwedge_{i \leq n} (e_i :: U_i \wedge R_i(a, e_i)))$$

The universals U_1, \dots, U_n used in an elementary specification are called features. A special case of an elementary specification is a quality specification where U_1, \dots, U_n are quality universals and the instances of U are substances.

⁵ In context of this article, the symbols $<$ and \leq represent the *part-of and reflexive part-of* relations, respectively. These relations are discussed in section 2.5.

Humans, as cognitive subjects, grasp universals by means of concepts that are in their head and cannot capture the universals completely, but only as approximate views.

Meta-Universals of Finite Order. Ordinary universals are universals of first order and the instances of universals of $(n+1)$ -th order are universals of n -th order. Instantiation relations of n -th order are denoted by $::_n$, and the relation $::_1$ is also notated as $::$. Since no universal is a set, it follows that all universals (of whatever order) are urelements.

2.4 Relations and Relational Universals

Relations are entities which glue together other entities. Without relations the world would fall into many isolated pieces. Every relation has a number of relata or arguments which are connected or related by it. The number of a relation's arguments is called its arity. Relations can be classified according to the types of their relata. There are relations between sets, between individuals, and between universals, but there are also cross-categorical relations for example between urelements and sets or between sets and universals.

We divide relations into two broad categories, called material and formal, respectively. The relata of a material relation are mediated by individuals which are called relators. Relators are individuals with the power of connecting entities; a flight connection, for example, is a relator that connects airports.

A formal relation is a relation which holds between two or more entities directly – without any further intervening individual. Examples of formal relations are: *5 is greater than 3*, *this day is part-of this month*, and *N is subset of Q*.

Holding Relation and Facts. One important formal relation is called the holding relation. If r is a relator connecting the entities a_1, \dots, a_n , $n \geq 1$, then we say that r, a_1, \dots, a_n (in this order) stand to each other in the holding relation, symbolically $h(r, a_1, \dots, a_n)$. The fact that h holds directly success to block the obvious regress which would arise if a new material relation were needed to tie h to r, a_1, \dots, a_n , and so on. Holding holds directly.

If r connects (holds of) the entities a_1, \dots, a_n , then this yields a new individual which is denoted by $\langle r: a_1, \dots, a_n \rangle$. Individuals of this latter sort are called *material facts*.

A material fact $\langle r: a_1, \dots, a_n \rangle$ has a duration, which depends on the lifetime of the relator r . We write $\langle r: a_1, \dots, a_n; t \rangle$ if t is a chronoid which is a part of the lifetime of r , i.e. this fact exists at least during the chronoid t .

Relator Universals. A relator universal is a universal whose instances are relators. For every relator universal R there exists a set of facts, denoted by $facts(R)$, which is defined by the instances of R and their corresponding arguments. We assume the axiom that for every relator universal R there exists a factual universal $F = F(R)$ whose extension equals the set $facts(R)$. Take, for example, the relator universal *Conn* whose instances are individual flight connections. Then we may form a factual univer-

sal $F(Conn)$ having the meaning ‘An airport X is connected to an airport Y’ whose instances are all facts of the form $\langle c:a,b \rangle$, where c is an individual connection and a, b are individual airports.

Formal Relations. A formal relation is a relation which holds between two or more entities directly – without any further intervening individual. A formal relation may be either an extensional relation (i.e. a set) or it may be given by a relational universal (having an intension and an extension). If R is a formal relation and $[a,b]:R$ then $\langle R:a,b \rangle$ is called a formal fact.

2.5 Basic Ontological Relations

We can distinguish a number of basic ontological relations which form an important part of the upper level ontology of GOL. The first and most familiar one is set-theoretic membership, denoted by \in . Further basic relations include:

- the proper and reflexive part-of relations, denoted by $<$ and \leq
- the contextual part-of relation, denoted by $<_U$, where the universal U denotes the context
- the holding relation h
- the inherence relation, denoted by i
- the instantiation relation, denoted by $::$
- the relational instantiation, denoted by $:$

We discuss some of these basic ontological relations in more detail.

Instantiation. The symbol $::$ denotes the instantiation relation. Its first argument is an individual, and its second a universal. If $x::u$, then u is a certain time- and space-independent pattern of features and x is an individual in which this pattern of features is realized. x might be, for instance, a molecule of DNA, u a pattern of features shared by all exactly similar molecules, where the notion of exact similarity is determined by the granularity and point of view of genetic science. The symbol $:$ denotes the relational instantiation. Its first argument is a list of entities, and its second a relation universal. Note, that the components of the instantiating list are not necessarily individuals.

Part-Whole Relation. There are many different part-whole relations between individuals. They can be classified by means of the axioms they satisfy. All part-whole relations are asymmetric and transitive. In addition to formal part-whole relations, there are also material part-whole relations. Part-whole relations may be either proper (denoted by $<$) or reflexive (denoted by \leq). We use the following definitions:

$$\text{ov}(x,y) =_{\text{df}} \exists z (z \leq x \wedge z \leq y) \quad (\text{overlap})$$

$$\mathbf{x \leq y =_{df} (x = y \vee x < y)} \quad \text{(reflexive part-whole)}$$

A proper part-whole relation $<$ is a strict partial order, that is, it satisfies the following axioms:

$$\neg \mathbf{x < x} \quad \text{(irreflexivity)}$$

$$\mathbf{x < y \rightarrow \neg y < x} \quad \text{(asymmetry)}$$

$$\mathbf{x < y \wedge y < z \rightarrow x < z} \quad \text{(transitivity)}$$

In addition, it may satisfy some of the following axioms:

$$\mathbf{x < y \rightarrow \exists z (z \leq y \wedge \neg \text{ov}(z, x))} \quad \text{(weak supplementation)}$$

$$\neg \mathbf{x \leq y \rightarrow \exists z (z \leq x \wedge \neg \text{ov}(z, y))} \quad \text{(supplementation)}$$

$$\mathbf{(z < x \wedge z < y) \rightarrow (x \leq y \vee y \leq x)} \quad \text{(exclusivity)}$$

Contextual Part-Whole Relation. The contextual part-whole relation $x <_U y$ has the meaning: “ U is a universal and x is a part of y in the context of U ”. Briefly, if x is a U -part of y in this sense, then x and y are parts of instances of U and $x \leq y$. But more is involved, since again the notions of granularity and point of view are an issue. We propose the following axiom: for every universal U there are universals U_1, \dots, U_n such that $x <_U y$ implies that x, y are instances of one of the U_i 's and every instance of one of the U_i 's is part of an instance of U .

Consider the following example, taken from the domain of biology. Let T be the biological universal whose instances are those organisms called trees. Then $x <_T y$ describes the part-whole relation based on the granularity of the context of whole trees. A biologist is interested in describing the structure of trees only in terms of parts of a certain minimal size. He is not interested in atoms or molecules. There is a finite number of universals $\{U_1, \dots, U_n\}$ by which the biologically relevant parts of trees are demarcated. All such parts of trees are either instances of some U_i , $1 \leq i \leq n$, or they can be decomposed into a finite number of parts, each of which satisfies this condition. Examples of universals U_i within the granularity of the tree context would be branch of a tree, leaf of a tree, trunk of a tree, root of a tree, and so on.

We have the following axioms:

$$\forall \mathbf{xyU} (x <_u y \rightarrow \text{Univ}(U) \wedge x < y) \quad \text{(CPW1)}$$

$$\forall \mathbf{xyzU} (x <_u y \wedge y <_u z \rightarrow x <_u z) \quad \text{(CPW2)}$$

3 Ontological Foundations of the UML

In the sequel, we refer to the *OMG UML Specification 1.4*, when we cite text in italics using page references in the form of [p.2-31].

For simplicity, we simply say *conceptual model* when we mean a conceptual model of a domain in the form of a *UML class model*. Whenever the context is clear, we omit the name space prefix UML and simply say ‘object’, ‘class’, etc., instead of ‘UML object’, ‘UML class’, etc.

3.1 Classes and Objects

In the UML, “an object represents a particular instance of a class. It has identity and attribute values.” While in the UML objects are instances of classes, individuals are instances of universals in GOL.

A “*Class describes a set of Objects sharing a collection of Features, including Operations, Attributes and Methods, that are common to the set of Objects.*” [p.2-26] “*The model is concerned with describing the intension of the class, that is, the rules that define it. The run-time execution provides its extension, that is, its instances.*” [p.3-35] Attributes come with associated data types. Since in conceptual modeling, the behavior of objects is normally not taken into consideration, we exclude the ‘operations’ and ‘methods’ of an object from our discussion.

We may observe a direct correspondence between universals and classes of a certain kind, as stated in the following principles.

Principle 1 (Class) In a conceptual model, any universal U of the domain may be represented as a concrete class C_U . Conversely, for all concrete classes (of a conceptual model of the domain) whose instances are basic objects or links (representing individuals), there must be a corresponding first-order universal in the domain.

Principle 2 (Attribute) If there is an elementary specification for a universal U , then any feature of it may be represented as an attribute of the corresponding class C_U in a conceptual model of the domain.

In a conceptual model, any individual of the domain that is an instance of a universal may be represented as an object (or link) of the class representing the universal. Notice that classes are not sets. In general, two classes C_1 and C_2 with identical extensions, $Ext(C_1) = Ext(C_2)$, even if they have the same set of attributes, are not equal, $C_1 \neq C_2$.

3.2 Powertype

“*A Powertype is a user-defined metaelement whose instances are classes in the model.*” A powertype is a special class, designated with the stereotype ‘powertype’. It represents a higher-order universal of order n whose instances are universals of order

$n-1$. Unfortunately, the UML does not provide higher-order ‘*isInstanceOf*’ relationships.

Figure 2 shows an example of a *powertype* represented in UML. In this figure, a Pigeon is a first-order universal whose instances are particular pigeons. However, individual breeds of birds, such as Pigeon and Peafowl, are also instances of the second-order universal Breed. In GOL, these two instantiation relations can be represented as $x::Pigeon::_2Breed$, where x is a particular pigeon.

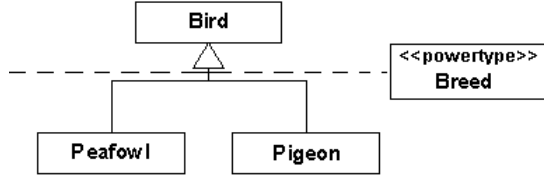


Fig. 2. Example of a second-order universal *Breed* represented in UML as a powertype.

3.3 Association

In the UML, the ER concept of a relationship type is called association. “An association defines a semantic relationship between classifiers. The instances of an association are a set of tuples relating instances of the classifiers. Each tuple value may appear at most once.” [p. 2-19] “An instance of an Association is a Link, which is a tuple of Instances drawn from the corresponding Classifiers.” [p. 2-20]

The OMG UML Specification is somehow ambiguous in defining associations. An association is primarily considered to be a ‘connection’, but, in certain cases (when ever it has ‘class-like properties’), an association may be a class: “An association class is an association that is also a class. It not only connects a set of classifiers but also defines a set of features that belong to the relationship itself and not any of the classifiers.” [p.2-21]

An association A between the classes C_1, \dots, C_n of a conceptual model can be understood in GOL as a relation R between the corresponding universals U_1, \dots, U_n induced by the relational universal whose extension consists of all relational moments corresponding to the links of A . Let $\phi(a_1, \dots, a_n)$ denote a condition on the individuals a_1, \dots, a_n . Then

$$[a_1 \dots a_n]: R_A (U_1 \dots U_n) \leftrightarrow \bigwedge_{i \leq n} a_i :: U_i \wedge \phi (a_1 \dots a_n)$$

An association is called *material* if there is a relator universal F such that the condition ϕ is obtained from F as follows:

$$\phi(a_1 \dots a_n) \leftrightarrow \exists k (k :: F \wedge h(k, a_1 \dots a_n))$$

An example of a ternary material association is *purchaseFrom* corresponding to a relator universal *Purchase* whose instances are individual purchases. These individual

purchases connect three individuals: a *person*, say *John*, an individual *good*, e.g. the book *Speech Acts by Searle*, and a *shop*, say *Amazon*. Thus,

[John, SpeechActsBySearle, Amazon]:R_{purchFrom}(Person, Good, Shop)

since *John::Person*, *SpeechActsBySearle::Good* and *Amazon::Shop*, and there is a specific purchase event *p::Purchase* such that

***h*(p, John, SpeechActsBySearle, Amazon).**

We obtain the following definition for the triple $[a_1, a_2, a_3]$ being a link of the association *purchFrom* between *Person*, *Good* and *Shop*:

$[a_1, a_2, a_3]:R_{\text{purchFrom}}(\text{Person}, \text{Good}, \text{Shop}) \leftrightarrow a_1::\text{Person} \wedge a_2::\text{Good} \wedge a_3::\text{Shop} \wedge \exists p (p::\text{Purchase} \wedge h(p, a_1, a_2, a_3))$

3.4 Aggregation and Composition

“An association may represent an aggregation; that is, a whole/part relationship. In this case, the association-end attached to the whole element is designated, and the other association-end of the association represents the parts of the aggregation. Only binary associations may be aggregations. Composite aggregation is a strong form of aggregation, which requires that a part instance be included in at most one composite at a time and that the composite object has sole responsibility for the disposition of its parts. This means that the composite object is responsible for the creation and destruction of the parts. In implementation terms, it is responsible for their memory allocation. If a composite object is destroyed, it must destroy all of its parts. It may remove a part and give it to another composite object, which then assumes responsibility for it. If the multiplicity from a part to composite is zero-to-one, the composite may remove the part, and the part may assume responsibility for itself, otherwise it may not live apart from a composite.” [p.2-66]

“A shareable aggregation denotes weak ownership; that is, the part may be included in several aggregates and its owner may also change over time. However, the semantics of a shareable aggregation does not imply deletion of the parts when an aggregate referencing it is deleted. Both kinds of aggregations define a transitive, antisymmetry relationship; that is, the instances form a directed, non-cyclic graph. Composition instances form a strict tree (or rather a forest).”[p.2-67]

A part-whole relation is by default expressed as an aggregation in the UML. Otherwise, if the parts in the part-whole relation are non-shareable, then the relation is expressed as a composition (black-diamond) – the use of composition implies the maximum cardinality of 1 w.r.t. the whole. An example of part-whole relation with shareable parts is the relation between researchers and research groups (researchers can be part of several research groups) – figure 2(a) Conversely, the relation between

a person and one of his organs is an instance of a non-shareable part-whole relation (a human organ cannot be shared by more than one human body).

3.4.1 Mandatory aggregates and Mandatory parts

Asides from the distinction between shareable and non-shareable parts, UML allows to express the distinctions between mandatory and optional aggregates and between mandatory and optional parts.

A *mandatory aggregate* refers to whether a part can exist without being part of an aggregate of certain class. This is the case, for instance, of a human heart that must be attached to a human body but not necessarily the same body forever. This is represented in UML by a minimum cardinality of (at least) 1 in the aggregate side. This example also depicts the notion of *mandatory parts*: refers to whether a whole can exist without having a part of a certain class – a human body must have a human heart but not necessarily the same heart for the entire life. This is represented in UML by a minimum cardinality of (at least) 1 in the part side of the relation. Figure 3(b) depicts this example as a non-shareable part-whole relation with mandatory aggregate and mandatory part.

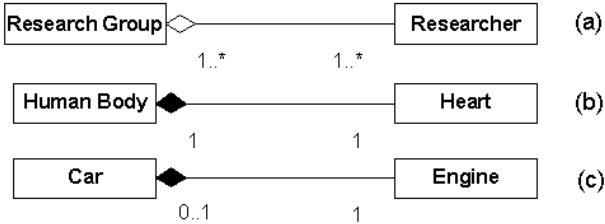


Fig. 3. A part-whole relation with sharable part; Fig 3.b part-whole relation with non-sharable and mandatory part and mandatory whole; Fig 3.c part-whole relation with non-sharable and mandatory part and optional whole.

3.4.2 Inseparable and Essential Parts

The concept of Inseparable parts refers to whether an object can exist without being part of a particular whole. UML is quite imprecise when dealing with the difference between Inseparable Parts and Mandatory aggregates. The specification prescribes that “*the composite object is responsible for creation and destruction of its parts*” and that, “*if the composite is destroyed it must destroy all its parts*”. Let w be a whole and p be one of its parts, this means that, $lifetime(p) \leq lifetime(w)$ and that the destruction of whole implies the destruction of the part.

Lifetime dependency is a characteristic of part-whole relations with inseparable parts – i.e. parts that are dependent always on the same whole. In this sense, we disagree with examples such as the one used in [7] to justify the existence of separable parts but that share same destruction as the whole: “*a car wheel is independent of the car but if the wheel is in the car during the car’s destruction then it is also de-*

stroyed”. In this case, the wheel is clearly separable from the car, it just happened to be the same event that caused the destruction of both objects (if the wheel was separated from the car the car’s destruction would not propagate to the wheel). Nonetheless, the UML specification also states “it [the whole] may remove a part and give it to another composite object, which then will be responsible for it” which give us the clearly impression that the composition notation with one-to-one cardinality refers to mandatory wholes but not to inseparable parts. This is confirmed by the statement “if the multiplicity from the part to composite is zero-to-one, the composite may remove the part, and the part may assume responsibility for itself, otherwise it may not live apart from a composite”. In sum, the UML composition notation means: (i) zero-to-one cardinality – non-shareable separable parts with optional aggregate – figura 3.c; (ii) one-to-one cardinality: non-shareable separable parts but with mandatory aggregate – figura 3.b; In both cases, there is no necessary relation between the chronoids $lifetime(p)$ and $lifetime(w)$.

We can conclude that in its current form (version 1.4), UML does not allow to express the distinctions between *separable* and *inseparable* Parts. Likewise, it also does not allow expressing the distinctions between *essential* and *nonessential* parts. The concept of *essential* part refers to whether an object can exist without having a particular object as a part. An example is relation between the car and its chassis. The removal of the car chassis breaks the identity criteria of the car. The human brain in the relation with a Person is an example of an essential part (actually this is an example of both essential and inseparable part) – figure 4.

In order to represent these two important ontological distinctions we therefore propose to extend UML by adding two Boolean-valued tags for the part association-end of an aggregation: *inseparable* and *essential*.

Proposal 1 (Aggregation with Inseparable and Essential Parts) UML 1.4 has to be extended by adding two predefined Boolean-valued tags for the part association end of an aggregation: *inseparable* and *essential*, whose default value is false.

A lifetime dependency between a part and a whole, referred to in the sentence “If a composite object is destroyed, it must destroy all of its parts” [p.2-66], is only in effect in the case of an inseparable part. In the case of a merely mandatory aggregate, it would be admissible to reassign the part when the mandatory aggregate is destroyed.

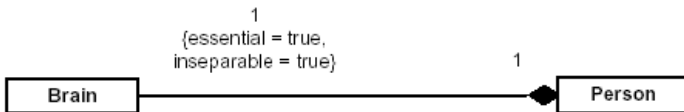


Fig. 4. An example of a composition with parts that are inseparable and essential.

According to the notation we propose, a composition with parts that are inseparable and essential would be expressed as in Fig. 4. As this example seems to suggest, one may be inclined to assume that an inseparable part is always non-shareable. But this is not the case, as the example in Fig. 5 shows.

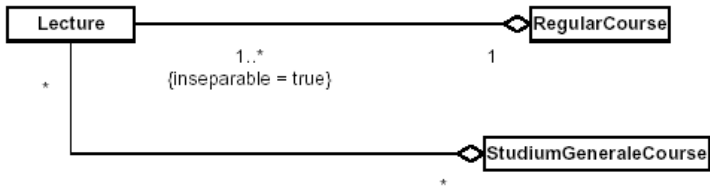


Fig. 5. An example of an aggregation with shareable parts that are inseparable: although a specific lecture of a regular course is an inseparable part of that course, it may be shared with any number of studium generale courses.

3.4.3 The Scope of Transitivity

Among the primary characteristics of part-whole relations, the most discussed one is transitivity. In the domain of philosophy and mathematics there is almost a consensus that transitivity should be included in the basic axiomatization of a part-whole theory [1,8,9]. Likewise, in Object Oriented modeling, several authors propose transitivity as a primary characteristic of aggregation [10]. More specifically, In UML, *“both kinds of aggregations define a transitive, antisymmetric relationship”*. We can say, for example, that the hand is part of the arm, the arm is part of the human body, ergo, the hand is part of the human body. However, there are also various authors who claim that transitivity does not always hold. As an example, the brain is part of person, this person is part of a Research Group but it sounds strange to state that the person’s brain is part of the Research Group.

On the basis of linguistic and cognitive studies, Winston, Chaffin and Herrmann (WCH) proposed a distinction among various sorts of part-whole relationships, with the aim to overcome these apparent transitivity paradoxes [11]. The main idea was capture the different ways in which parts contribute to the structure of the whole. This framework has been further refined in [12] by isolating three basic kinds of wholes: *masses*, *collections* and *complexes* (whose parts are respectively called quantities, members and components). *Masses* are homogeneous aggregates which are similar to its quantities (*homogeneous*) – e.g. an amount of sugar. *Complexes* are characterized by structural (configurational) relationships among the parts – e.g. the human body and its parts. Finally, in *collections* a part is regard not to have an essential role w.r.t. the whole – e.g. a tree is part of forest, a person is part of research group [7].

In [10], it is claimed, *“as long as we are careful to keep a single sense of part, it seems that the part-whole relation is always transitive. However, when we inadvertently mix different meronymic relations problems with transitivity arise”*. In [7], Henderson-Sellers and Barbier disagree with this statement showing the following counter-example: *“I am member of a club (collection) and my club is a member of an International body (collection). However, it does not follow that I am a member of this International body since this only has clubs as members, not individuals”*.

In the example given above, a *research group* is a sort of social system, i.e. an instance of whole whose parts comprise a set of socially linked persons. The *brains* of such individuals are parts of the latter but do not qualify as members or components of a social system because they do not enter independently in social relations: only entire

persons can enter independently in those type of social relations. In other words, the set of parts of a *research group* is not the collection of all its parts but the collection of its atoms, i.e. the collection of those parts that are *socially coupled*. This particular notion of atomic composition is expressed in GOL in terms of the contextual part-whole relation. In this case, if x is part of y in context of a *research group*, then either x is an instance of the universal *Person* or x can be decomposed in finite parts that are instances of *Person* (e.g. a subgroup is composed of members which are instances of *Person*, thus, a subgroup can be considered a part of a research group).

In [1] and in [13], a distinction is made between a formal part-whole relation that is unrestrictedly transitive and material part-whole relations whose transitivity is scoped to a certain context. Moreover, according to [13], absolute transitivity is a characteristic that only makes sense in a mathematical perspective, for the axiomatic definition of the theory. In the ontological and cognitive sense, part-whole relations should only be interpreted w.r.t. a certain context.

The distinction between formal and material (contextual) part-whole relations seems to be a much more elegant solution to the apparent transitivity paradox – transitivity always holds for contextual part-whole relations (axiom *CPW2*). We therefore propose to add the concept of a contextual part-whole relation, as described in section 2.5, to the UML. For defining the context of such a relation, we propose to use the UML construct of a *package*.

Proposal 2 (Contextual Aggregation) UML 1.4 has to be extended by adding a contextual aggregation construct. Such an aggregation is defined within a package, and any part class participating in it must also be in that package. Transitivity, then, only holds between the contextual aggregations within a package, but not across packages.

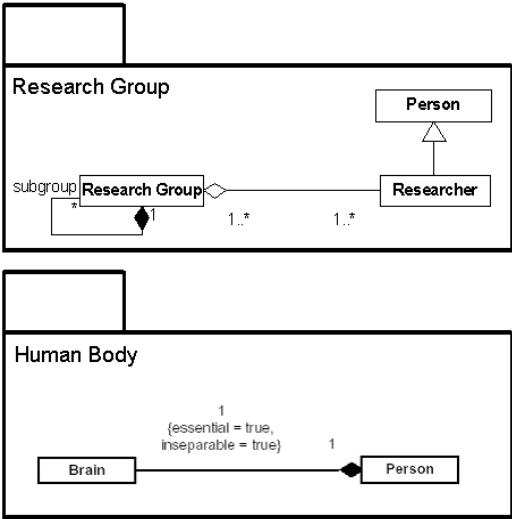


Fig. 6. The same individual researcher participating in different contextual part-whole relations.

4 Related Work

The upper level ontology of GOL is under development at the Institute for Formal Ontology and Medical Information Science at the University of Leipzig, Germany. The project is a collaboration between philosophers, linguists and other cognitive scientists and computer and information scientists. For a comparison between GOL and other upper-level ontologies, such as the IEEE Standard Upper Ontology [14], KIF [15], Sowa [16], Russel and Norvig [17] and LADSEB [18,19], one should refer to [1].

The goal of this paper is to use the upper level ontology of GOL to analyze the ontological correctness of a conceptual UML class models and to develop guidelines that assign well-defined ontological semantics to UML constructs. For this reason, we limit our discussion of related work w.r.t. this point .

The approach found in the literature that is closest to the one presented here is the approach by Evermann and Wand [20] and, Weber, Storey and Weber [21]. In these two articles, the authors report their results in mapping common constructs of conceptual modeling to an upper level ontology. Their approach is based on the BWB ontology, a framework created by Wand and Weber on the basis of the original metaphysical theory developed by Mario Bunge in [8] and [13].

The concepts of substance (in GOL) and of thing in BWB are both based on the Aristotelian idea of substance: (i) an essence which makes a thing what it is; (ii) that which remains the same through changes; (iii) that which can exist by itself, i.e., which does not need a ‘subject’ in order to exist. In [20], it is proposed that “*only substantial entities in the world are modeled as objects*” (Rule 1). As a direct consequence, classes in UML should only represent universals that denote substances. Conversely, in our approach, concrete classes can represent any type of universals. In [20], instantiation relations of higher-order are not considered, therefore, no guideline for the UML concept of *powertype* is provided.

In BWB, a thing has necessarily at least one property. Likewise, a property exists only in connection with things. A property whose existence depends only on a single thing is called an intrinsic property (e.g. the height of a person). A property that depends on two or more things is called a mutual property (e.g. being a student is a mutual property between a person and an educational institution). Again the concepts of BWB’s intrinsic and mutual properties can be directly related to the concepts of quality and relational moments in GOL. Nevertheless, in BWB only things possess properties. As a consequence, a property cannot have properties. This dictum leads to the following modeling principle: “*Associations should not be modeled as classes*”, (Rule 7) in [20]. Contrary to this principle, GOL allows associations, representing relational universals, to have attributes and to participate in second or higher-order associations. Thus, while the BWB approach prohibits to use association classes in conceptual modeling, they are allowed in GOL.

In [22], Opdahl et al. uses BWB ontology as a foundation for a conceptual framework defining a taxonomy of part-whole relations in terms of its primary (e.g. reflexivity, asymmetry and transitivity), secondary (shareability, mutability, separability, etc...) and consequent properties (e.g. ownership, propagation of operations, encapsulation). Moreover, the article analyses the different kinds of part-whole relations in

terms of ontological expressiveness, i.e., if the proposed concepts are meaningful in terms of real-world semantics. Finally, some UML stereotypes are proposed in order to provide syntactical representations to the proposed ontological distinctions.

In our approach we are interested on those properties that have meaning in terms of ontological correspondence. For that reason, we have chose to ignore properties that are relevant only in terms of implementation decision (e.g. ownership, propagation of operations, encapsulation, by-value or by-reference, used or not used). In terms of the primary characteristics of part-whole relations there are two points of disagreement between the two proposals:

Transitivity: Opdahl et. al. claim that transitivity is valid only for certain types of relation and, thus, should not be considered as a primary characteristic. As we have discussed in section 3.4, proposal 2, both in the ontological and cognitive sense, part-whole relations should only be interpreted w.r.t. a certain context. Moreover, contextual part-whole relations are always transitive. In our approach we, thus, consider transitivity w.r.t. a certain context as a primary characteristic of part-whole relations.

Emergent and Resultant properties: both in [21] and [22] it is proposed that a thing should only be modeled as an aggregate if we are interested in modeling its emergent and resultant properties. In [22], it is presented a UML class diagram with the classes *Department* and *Employee*. In this example they state that in the problem domain all *Departments* are aggregates of *Employees*. However, it is claimed unacceptable to add a part-whole between these two classes since the diagram would comprise no resultant/emergent property of *Department* relative to *Employee*. We strongly disagree with this view and we think that this restriction arises from a misinterpretation of Bunge's ontology. According to Bunge, every aggregate will certainly have emergent and resultant properties. However, it is important to notice that his Ontology makes explicit the distinction between the properties possessed by a thing and the representations of these properties, namely attributes. According to Bunge, there are no bare individuals, i.e., things without properties: a thing posses at least one substantial property, even if we humans are not or cannot be aware of them. Humans get in contact with the properties of things exclusively via the things attributes, i.e. via a chosen representational view of its properties. In sum, we agree that emergent/resultant properties are basic characteristics of part-whole relations, in the sense they are present in all of them. What we do not agree is to use the existence of resultant/emergent attributes as a criteria for representation of part-whole relationships. In other words, these properties will always exist but we do not have always to be interested in them and sometimes we cannot even be aware of them. Additionally, we think that the representation of these attributes is not a necessary condition for one to benefit from the representation of part-whole relations in terms of communicability, understanding and problem-solving.

5 Conclusions

The development of a well-grounded, axiomatized upper level ontology is an important step towards the definition of real-world semantics for conceptual modeling diagrammatic languages. In this paper we use the General Ontological Language (GOL) and its underlying upper level ontology to evaluate the ontological correctness of UML as a conceptual modeling language. Moreover, we develop guidelines that assign well-defined ontological semantics to UML constructs. In particular, we have focused on the UML metaconcepts of *classes and objects*, *powertypes*, *association* and *part-whole* relations (*aggregation/composition*).

Except from the distinction between shareable and non-shareable parts, the UML allows to express the distinctions between mandatory and optional aggregates and between mandatory and optional parts. However, in its current form (UML 1.4), it does not allow to express the distinctions between separable and inseparable parts, and between essential and nonessential parts. We therefore propose some extensions to version 1.4 of the UML in order to obtain a more satisfactory treatment of *part-whole* relations. Additionally, we define in terms of *GOL's contextual part-whole relation*: (i) a solution to the transitivity paradox of part-whole relations; (ii) a representation of contextual part-whole relations in UML.

Acknowledgements We are grateful to Barbara Heller and João Paulo Andrade Almeida for fruitful discussions and for providing valuable input to the issues of this article.

References

1. Degen, W., Heller B., Herre H., Smith, B.: GOL: Towards an axiomatized upper level ontology. In Barry Smith and Nicola Guarino, editors, Proceedings of FOIS'01, Ogunquit, Maine, USA, October 2001. ACM Press.
2. Cranefield, S., Purvis M.: UML as an ontology modelling language, Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99), Germany, University of Karlsruhe (1999) 46-53.
3. Baclawski K. et al.: Extending UML to Support Ontology Engineering for the Semantic Web, Fourth International Conference on the Unified Modeling Language: UML 2001, Toronto, Canada, 2001.
4. Overgaard G.: A formal approach to relationships in the Unified Modeling Language. In Manfred Broy, Derek Coleman, Tom S. E. Maibaum, and Bernhard Rumpe, editors, Proceedings PSMT'98 Workshop on Precise Semantics for Modeling Techniques. Technische Universität München, TUM-19803, 1998.
5. Evans A., France R., Lano K., Rumpe B.: Developing the UML as a formal modelling notation. In Pierre-Alain Muller and Jean Bezivin, editors, Proceedings of UML'98 International Workshop, Mulhouse, France, June 3 - 4, 1998, pages 297-307. ESSAIM, Mulhouse, France, 1998.
6. Guizzardi, G., Herre, H., Wagner G.: On the General Ontological Foundations of Conceptual Modeling. In Proceedings of 21th International Conference on Conceptual Modeling (ER 2002). Springer-Verlag, Berlin, Lecture Notes in Computer Science.

7. Henderson-Sellers, B., Barbier F.: What Is This Thing Called Aggregation, proceedings of Technology of Object-Oriented Languages and Systems Europe'99, Nancy, France, IEEE Computer Society Press, pp. 236-250, June 7-10, 1999 (ISBN: 0-7695-0275-X)
8. Bunge M.: Treatise on Basic Philosophy. Vol. 3. Ontology I. The Furniture of the World. D. Reidel Publishing, New York, 1977.
9. Varzi. A.C.: Parts, wholes, and part-whole relations: The prospects of mereotopology. J. of Data and Knowledge Engineering, 20:259–286, 1996.
10. Artale A., Franconi E., Guarino N., Pazzi L.: 1996. Part-Whole Relations in Object-Centered Systems: an Overview. Data and Knowledge Engineering, 20(3): 347-383.
11. Winston M.E.; Chaffin R.; Herrman D.: A taxonomy of part-whole relations, Cognitive Science 11 (1987), pp.417-444.
12. Iris M.; Litowitz B.; Evens M.: Problems of the part-whole relation. In M.Evens (Ed.) Relational Models of the Lexicon, Cambridge University Press (1988), pp.261-288.
13. Bunge M.: Treatise on Basic Philosophy, vol. 4: Ontology II: A World of Systems, Reidel Publishing Company, Dordrecht, Holland, 1979.
14. SUO: <http://suo.ieee.org>.
15. Genesereth, M. R., Fikes, R.E.: Knowledge Interchange Format, Version 3.0, Reference Manual, Logic Group Report Logic-92-1, Computer Science Department, Stanford University.
16. Sowa J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, 1999.
17. Russell, S., Norvig, P.: Artificial Intelligence, Prentice Hall, 1995.
18. Gangemi, A., Guarino, N., Masolo, C., Otramari, A.: Understanding Top-Level Ontological Distinctions. Technical Report 04/2001, LADSEB-CNR.
19. Guarino, N.: Formal Ontology and Information Systems, in: N. Guarino (ed.) Formal Ontology in Information Systems Proceedings of FOIS' 98, Trento, Italy, Amsterdam, IOS-Press.
20. Evermann J., Wand Y.: Towards ontologically based semantics for UML constructs. In H.S. Kunii, S. Jajodia, and A. Solvberg, editors, Proceedings of ER 2001, pages 354–367. Springer-Verlag, 2001.
21. Wand Y., Storey V.C., Weber R.: An ontological analysis of the relationship construct in conceptual modeling. ACM Transactions on Database Systems, 24(4):494–528, December 1999.
22. Opdahl A., Henderson-Sellers, B., Barbier F.: Ontological Analysis of whole-part relationships in OO-models, Information and Software Technology 43 (2001), pp. 387-399.