On a purely taxonomic and descriptive meaning for classes

Jérôme Euzenat

INRIA Rhône-Alpes
LIFIA/IMAG,
46, avenue Félix Viallet,
F-38031 Grenoble Cedex 1
Jerome.Euzenat@imag.fr

Abstract

Three different aspects of classes in object-based systems are studied: the distinction between classes and instances, the separation of ontological from taxonomic function of classes and their descriptive or definitional meaning. The advantages of using a descriptive and taxonomic meaning for classes are advocated. One of the important reasons for separating ontology from taxonomy is the multiplicity of taxonomies over a same set of objects and the independence of objects from these taxonomies. These distinctions ground the semantics of the object-based representation system TROPES. The specialisation relation in TROPES is examined under this light and the classification mechanism is interpreted under the descriptive setting. It is shown that the use of a descriptive semantics of classes can support a semantics for the classification mechanism. In fact, there is no intrinsic superiority of definition over description: the precision of the former is balanced by the generality of the later.

Key words: Specialisation – Classification – Categorisation – Instantiation – Descriptive classes – Definitional classes.

The semantics of object-based representation formalisms (OBR) is far from explicit. Some insight is given here about the possibility of building a system in which the meaning of entities is grounded on (1) the separation of individuals from classes, (2) the separation of the ontological side of classes from their taxonomic side and (3) the descriptive interpretation of all these entities. These conditions do not prevent object-based representations from using powerful inference mechanisms such as classification, but it has to be given the adequate interpretation. This work is exemplified by the TROPES system [14], but results concerning the descriptive interpretation of objects are valid for any descriptive OBR system.

1 Separating ontology and taxonomy

Important work in knowledge representation showed the relevance of defining once for all the meaning of a knowledge representation system [6]. However, the use of a particular interpretation enables or disables the validity of some inference mechanisms.

Here, three distinctions in the interpretation of classes are presented. The rational for distinguishing individuals from classes (and definition from description) is first given (§1.1) before that of using descriptions instead of definitions (§1.2). Then, another separation is made between the ontological side of classes and their taxonomic side (§1.3).

1.1 Classes versus individuals

William Woods distinguished between structural and assertional links in semantic nets [22]. This distinction is the composition of the distinction between classes and individuals on one hand [3] and the fact that these links assert information about an entity or are used for defining the entity on the other hand. For example, the distinction between assertional and structural interpretation of the class described by "a cup is a container with ears" is given by the phrasing "a cup is a container which can have ears" (structural), "a cup is a container which must have ears" (assertional-universal) and "this container is called a cup and has ears" (assertional-individual).

These distinctions have to be made. However, the confusion between individuals and classes was only effective in semantic nets and early frame languages [16]. The class/individual distinction is already accounted for by the class-instance distinction of object-oriented languages (OOL) and post-frame languages (or OBR: artificial intelligence languages, such as KRL, SRL, and their commercial counterpart such as the ART and KEE object languages, based on the frame model but strongly inspired by object-oriented languages [15]). Moreover, the distinction in most existing OOL is sharp since the language for expressing instances is weaker than the one for expressing classes: instances can have values but cannot support added constraints. For instance, usual terminological languages allow to talk of "a man with less than three daughters" at the assertional level while an OOL can represent a man but cannot constrain the number of his daughters but by providing the list of the daughters. This is an acknowledged limit of this sharp distinction, however, the OOL world is changing (see [20]).

1.2 Definition versus description

Once classes and individuals are distinguished, the meaning of classes remains loose: they could be *definitional* if they provide necessary and sufficient conditions for being a member of the class, or *descriptive* if they only provide necessary conditions. This distinction led to terminological languages (TL, [18, 5]) which distinguish terminological and assertional components of TL: primitive concepts are descriptive classes, constructed concepts are definitional classes and individuals are descriptive individuals.

In contrast, the definitional or descriptive character of OBR had hardly been clearly stated (see for example the discussion in [10]): even if it is often considered that the interpretation of classes is descriptive, they are sometime used like if it were definitional (for example when using automatic instance classification mechanism). A descriptive semantics of classes is provided below for OBR classes and this enables a clear interpretation of classification results. However, it is noteworthy that this interpretation never suffers from exception [4]: classes are necessary conditions.

The defence of descriptive classes has two justifications: First, there are many descriptive concepts in the world to be modelled and this should be accounted for by knowledge representation (for instance, a researcher is an employee with a PhD but the converse is not true). Second, the descriptive interpretation is weaker than the definitional interpretation. Thus, more can be represented in the descriptive interpretation but less can be inferred. This means that any result on descriptive interpretation is also valid for the definitional interpretation and that it is possible to *explicitly* and locally strengthen the definitions in order to use the definitional interpretation. As a matter of fact, if a cup is characterised as a container with one ear, then from a definitional interpretation, it is possible to infer that an individual container with ears is a cup. From a descriptive interpretation, it can only be inferred that the individual container can be a cup.

1.3 Being versus being categorised

In classical OBR, there are strong implications of belonging to a class. But some of them are necessary while others are contingent (exemplified by the fact that an object can change class).

Classes, in many languages, are loaded with too much meaning. They merge the ontological aspects of instances (i.e. the conditions of their being: structure, integrity, identity) and the taxonomic aspects (i.e. the conditions for seeing them as a member of a class: domain restriction, relevant fields). There are two important problems here:

1. Ontological and taxonomic aspects are merged. One cannot merge both views together because this raises problems of integrity: does some object change class (breaking the integrity of the object) or is it replaced by another object with similar characteristics but belonging to another class (giving up identity of objects)? [9] Classes cannot play their role of categories in an identification system without breaking the ontological link with the objects.

2. There exist several different ways to classify objects in concurrent taxonomies (for example, employees in your firm are not categorised the same way by yourself or by your account agent). It is thus important to define the function of classes as ontological or taxonomic.

So far, three important distinctions for object-based knowledge representation have been made. They form the basis of the definition of Tropes. The remainder shows how these distinctions can be used in a coherent manner.

2 Tropes

Here is a short description of the Tropes system which distinguishes classes from instances (§2.2) and ontology from taxonomy (§2.3). The consequences of the descriptive character of classes are presented in the following sections.

2.1 Overview

Tropes [14] is an object-based knowledge representation model. This means that individuals are represented as objects. These objects are partitioned into concepts (an object is instance of one and only one concept). As an example, the `employee` concept concerns all the entities which are employees. The fields relevant to objects are described in the concepts. The structure of an object is uniquely determined by a set of fields and their basic domains independently of the classes to which the object can be attached. The instances of the `employee` concept have a `name`, a `salary` and a `publications` field. The basic domain of a field is either a primitive type (real, integer, string, boolean), a concept, or a type constructed from these one (and only them) with the help of set and list constructors.

Objects can be seen under several viewpoints. A viewpoint determines the set of fields which are visible from that viewpoint. The `employee` instances can be seen under several viewpoints: as `employees` from the `accounting` viewpoint, as `researchers` from the `functional` viewpoint, as `customers` from the `restaurant`... The `publications` are not relevant to the `restaurant` viewpoint as well as the `diet` to the `functional` one, and thus the corresponding fields are hidden in the respective viewpoints. Thus viewpoints do have more in common with database views than with KEE viewpoints [13] for instance. Viewpoints also determine a hierarchy of classes under which the instances of the concept can be classified. Classes introduce constraints on their member field values. They are related through the specialisation relation (whose interpretation is given at §3) and determine progressive subsets of the set of the instances of the concept. This progression is parallel to the reinforcement of the constraints added to sub-classes. Under the `functional` viewpoint, `employees` are divided into `engineers`, `researchers`, `clerks`, and other classes; `researchers` can be divided into `directors, assistants` and `juniors`. An object is attached to only one more specific subclass under a viewpoint, but is member of all the classes of which this class is a specialisation. The system is able, given a particular object, to find the classes under which it can be attached (this is called "classification" in knowledge representation and "identification" in data analysis).

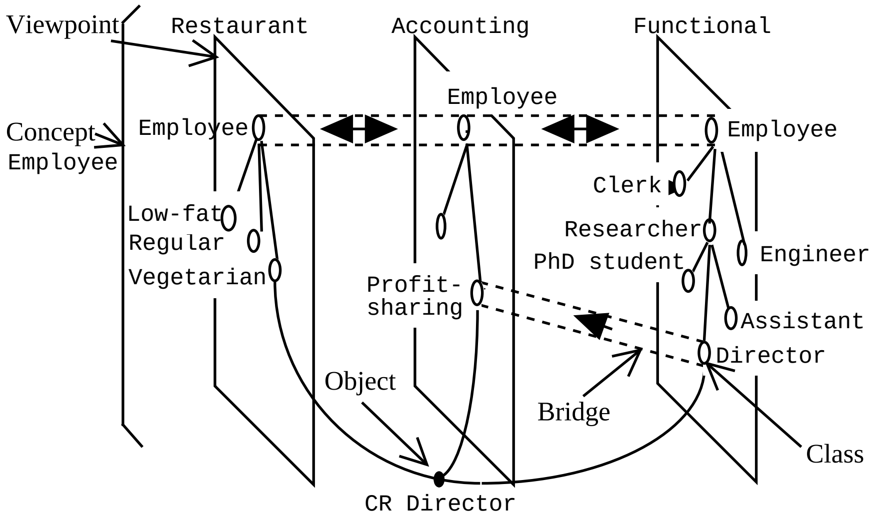

Figure 1: The `employee` concept

Bridges are knowledge components which link a set of classes under pairwise different viewpoints to a class in yet another viewpoint. The meaning of the bridge is that an object, which is member of each of the former classes, must be member of the later one. This allows, for instance, to infer that if an object is attached to the `director` class under the `functional` viewpoint, it is member of the `profit-sharing` employees under the `accounting` viewpoint.

In TROPES, the class is thus split into two separate units: the concept has in charge the ontological aspects of instances (i.e. the conditions of their being: structure, integrity, identity) and the class supports the taxonomic aspects (i.e. the conditions for seeing them as a member of a class: domain restriction, relevant fields). This allows to freely change the class of an object while its fundamental characteristics remain and to use several taxonomies simultaneously.

The usefulness of multiple taxonomies is twofold. First, one wants to use different related taxonomies (this is the case in biometrics when considering functional, chemical and evolutionary viewpoints on genes: there is a taxonomy for each viewpoint). Second, one wants to build a new taxonomy without discarding the current one (when testing a new hypothesis about the knowledge manipulated by the biometrician). For instance, from an already existing knowledge base containing an important set of objects, it is possible to run clustering algorithms for building another taxonomy on a new viewpoint and to compare it with other taxonomies.

Note that, in TROPES, the taxonomy is not part of the implementation level but of the knowledge level: it is aimed at representing man-made taxonomies and not these taxonomies that are the consequence of entity definitions (such as sub-typing, set-inclusion or structure subsumption in TL). They are compatible with them but not complete (§3).

Of course, the ontology is not unique and given once for all: it can be different with regards to the application. However, there already can be several taxonomies in the same application.

2.2 Ontological prerogatives

Ontological prerogatives are attached to the *concept*. They are those which warrant the integrity of an object (i.e. that it cannot be modified in a way which would lead it to cease being an instance of the concept). This also includes the management of objects identity. In TROPES, objects are identified with the help of a key: a list of field values that uniquely corresponds to one object. The usual consequences with keys are related to integrity: an object without a key cannot exist and there cannot exist two objects with the same key (thus accidental co-reference is avoided).

These prerogatives play an important role at instantiation when the object is created and registered. They have in charge the management of existential dependencies from objects to the components they cannot exist without. During its lifetime, an individual, coming under some concept, is ensured that none of the above postulates is violated.

Moreover, the field values are part of the objects and do not depend upon the classes to which they are attached. Thus inference methods which represent implicit knowledge (valid deduction or value computation which must return the true value of an object) are attached to the concept itself [21].

2.3 Contingency and taxonomy

As it has been said above the classification of objects is considered as contingent. Thus there can be several different classifications of the instances of the same concept: in a particular firm, the individuals are not considered the same way by the project management staff, the account office and the restaurant. Therefore the same individual is seen differently under several viewpoints and each viewpoint can have its classification of the same set of individuals.

Under a particular viewpoint, an object is attached to only one class (this was motivated by the argument that an object belonging to two unrelated classes is so under different viewpoints [14]). This organisation is very useful since, from a particular viewpoint, one can classify an object with regard only to the relevant fields of the object. As a consequence, the set of classes to be considered is small, while considering the whole lattice of possible structures (e.g. `Vegetarian-Profit-sharing-Clerk`) would confuse the user.

A *class* defines constraints that objects must satisfy in order to belong to the class. It is a projection of the structure of the concept retaining only relevant fields and it defines an elaborated domain for the field values by refining the basic domain of the fields. This is achieved with the help of:

- Primitive domain restriction provided by domain enumeration, exclusion or bounding.
- Attachment restriction for concepts: the field values must not only be instances of a particular concept, but also be attached to particular classes of that concept.
- Cardinality restrictions on sets (resp. lists) by bounding the cardinality. It is noteworthy that constructed field values are true sets (resp. lists) and not multi-values, i.e. the value of the field are given by the corresponding set (resp. list) and not by a subset (resp. sub-list) of it which could be completed later.

As opposed to instantiation, objects can be attached to a class and can be detached from it at anytime. Classes also allow the expression of hypothetical knowledge in term of default values or default inference methods. The hypothetical knowledge is relevant here since default values (for example) are given with regard to the specificity of the object class. These values are not to be seen as the true value of a field but as hypothesis that can be made under a particular viewpoint. As a matter of fact, default values returned from different viewpoints do not have to be identical. For instance, it could be the case that research directors are usually fat while vegetarian are usually thin: what about a vegetarian director?

The former paragraphs exemplified the account of class/instance and ontology/taxonomy distinction by the TROPES system. The remainder is devoted to the descriptive meaning of classes. For instance, in French public laboratories, there exist research engineers and research assistant. The first ones belong to the engineers body while the second ones belong to the researcher body. However, the only objective difference between these people is the category under which they have been hired! And there is no classificatory relation between these categories. This classification is very important for its consequences: only researchers can become research directors, only engineers can earn particular subsidies. But, there is no ontological difference between the members of these categories (as a matter of fact, an engineer can later become a researcher; but this can only be achieved by passing the same procedures as other candidates). Hence there are two categories with the same characteristics and they belong to the same hierarchy. As a consequence, any man fulfilling the conditions can be suspected to be either a researcher or an engineer. This answers the critics made to TL that such classes must be primitive terms and are thus unclassifiable [11]. It will be shown that the descriptive meaning for classes can support a specialisation relation and a classification mechanism.

3 The semantics of specialisation

The discussion above states the basic characteristics of a TROPES concept apart from the interpretation of the classification relation. The specialisation relation is given an interpretation according to a descriptive interpretation of classes. Next section shows that this descriptive interpretation does allow a correct and fruitful exploitation of the knowledge.

3.1 Set-based interpretation

The set-based view of TROPES only considers individuals, also called objects, and sets of objects (there are two kinds of sets: concepts and classes). The set of all objects is partitioned into concepts (one object belongs to one and only one concept). In a concept, several viewpoints over the same set of objects exist. A viewpoint on a concept corresponds to a structure (and not a partition) of the concept into hierarchically organised classes the most general of which, or *root class*, represents the whole concept.

As a consequence, classes are subsets of concepts. A class corresponds to a set E of objects which is called the *extension* of that class. The root class of each viewpoint represents the set of all the instances of the concept. Thus, an object is member of only one concept but of several classes: since TROPES can only attach an object to one more specific class by viewpoint, each class is divided into disjoint direct sub-classes. Classes are organised by the specialisation relation which is interpreted as inclusion ($\subseteq$) over their extensions. Consequently, there cannot be disconnection between classes of the same viewpoint (for each class but the root class, there is a class which includes it).

In order to allow cooperation between viewpoints (mainly during classification) *bridges* are introduced. Their set-based interpretation is that if a bridge goes from classes $A_1 \dots A_n$ to class B, then, each object belonging to each A_i is also an object of B (thus $A_1 \cap \dots A_n \subseteq B$). For instance, a bridge exists from any root class of a concept to any other root class under another viewpoint of that concept.

The set-based interpretation is sound but not complete in the sense that not every inclusion based structure corresponds to a viewpoint.

3.2 Structure-based interpretation

Once again, in the structure-based interpretation, it must be distinguished between two kinds of entities: structures and values. An instance of a particular concept is characterised by a collection of fields. The structures corresponding to concepts or classes can be described by record types with typed fields [8][1].

The fields are defined for all the instances of a concept but viewpoints hide some fields and classes add constraints on others. Hidden fields are considered as existing but not relevant for specialisation. In other terms, there is no constraint on such a field in the classes in which it is hidden. The structure associated to a class thus denotes a *potential extension* P (the set of all expressible objects satisfying the record type corresponding to the class).

[1] Recursive types will have to be introduced in TROPES and this requires a more powerful type system [2, 19]. However, recursive types are not dealt with here and it can be considered that the structure of a class is the definition of a record type.

In the structure-based view, the specialisation relation is interpreted as sub-typing ($<:$) on record types (this also could have been the O-subsumption [17]). It is noteworthy that the sub-typing relation is not necessarily isomorphic to the specialisation relation. As a matter of fact, two classes can have the same definition (and thus the same type) without even being related by specialisation. This is the case of the above mentioned engineer/researcher example. However, the specialisation must respect the sub-typing relation: a class can be a specialisation of another class only if the type corresponding to the former is a sub-type of the class corresponding to the later.

The structure-based interpretation of a bridge which relies classes $A_1 \ldots A_n$ to class B is that the record type associated with each of the A_i is a sub-type of the one associated with B (thus for each i, $A_i <: B$). This is a requirement of the TROPES model which is used in order to infer the necessary consequences on the record types.

3.3 Putting set and structure together

In fact the set-based and structure-based approaches reinforce each other and the relation made of set inclusion and record-type sub-typing is the *specialisation relation*. The next step after that dual explanation of the specialisation relation consists in determining if the interpretations are equivalent and, if not, which one prevails. In other terms: “Am I in such a set because of my structure” or “Do I have such a structure because I am in such a set?”.

Here comes the idea of necessary and sufficient conditions to belong to some class. When an object is voluntarily and successfully attached to a class, first, the structural conditions are met by the object and, second, the user decided to make of the object a member of the class. In this action, the type-checking which is necessary for successful attachment constitutes a necessary condition only grounded on the class structure, and the intent which is necessary for voluntary attachment constitutes a sufficient condition. There is no equivalence between E and P, but the general result is that $E \subseteq P$: if you got a PhD and are a member of the laboratory, you can be a researcher but maybe you are not, you are the cook. However, without a PhD, you cannot be a researcher.

Thus, the class definitions in themselves are (independently of their set of objects E) to be interpreted according to the descriptive interpretation of classes: they provide conditions that all attached objects must respect.

4 Classification and description

In order to show the usefulness of classification in a system whose semantics is given in terms of descriptions, a clear definition of classification is first given independently of any class system (§4.1). Then, the interpretation of classification and categorisation is given in the context of TROPES (§4.2).

4.1 General view of classification

Classification takes place over sets of abstract entities L_C and L_I the members of which are respectively named categories and individuals. An individual i can be classified under a particular category c. Here, these categories are just labels to put on entities and do not carry any meaning. They are assigned an interpretation for a particular system. The classification aims at determining the set $Cl(i)$ of categories in which an individual i is classified. The set of individuals i such that $c \in Cl(i)$ is called the extension of c (and noted $In(c) = \{i \in L_I; c \in Cl(i)\}$).

The set of categories is usually organised into a taxonomy (in the sense used in [23] for instance). A *taxonomy* is a structure made of a set C of categories and a relation $\leq$ on these categories which respects the *extension inclusion property*:

$$c' \leq c \Rightarrow In(c') \subseteq In(c).$$

As it is convenient to represent a taxonomy with the help of its transitive reduction [1], it is also convenient to represent the result of a classification with its minimisation. The *minimisation* of a set S with regard to a partial order R over its elements is defined by:

$$\mu_R S = \{x \in S; \forall y \in S, yRx \Rightarrow xRy\}.$$

It retains only minimal elements for R. Hence, the *minimal categories* of an instance i are $\mu_{\leq} Cl(i)$. *Terminal categories* of an instance i is the set

$$\nu Cl(i) = Cl(i) \cap \mu_{\leq} C.$$

It is the set of solutions which do not have sub-categories (or the set of leaves in which i is classified; the set $\mu_{\leq} C$ is called the terminal categories of the taxonomy).

For the purpose of describing classification algorithms, the following terms are defined: the *sub-classification* is the operation which classifies an object i in the sub-categories of a particular category c (which is called the *initial category*):

$$Cl_c(i) = Cl(i) \cap \{c' \in C; c \leq c'\}.$$

The *categorisation* (or class classification) supervises the construction of a taxonomy. It determines incrementally the relations between a new category and the already categorised ones. The categorisation helps constructing the $\leq$ relation according to a sub-categorisation criterion ($\ll$) external to the relation to be constructed but taking into account the already constructed part of this last relation. Intuitively, this criterion represents the loosest constraints that $\leq$ must respect in order to be a taxonomy. Thus, the property of $\ll$ is that

$$\forall c, c' \in L_C, c \ll c' \Rightarrow In(c) \subseteq In(c')$$

and any graph built with its help will respect the property

$$\forall c, c' \in L_C, c \leq c' \Rightarrow c \ll c'.$$

To that extent, the categorisation of a category c determines its more general categories (MGC) and more specific categories (MSC) with regard to the criterion:

$$MGC(c) = \{c' \in C; c \ll c'\} \text{ and } MSC(c) = \{c' \in C; c' \ll c\}.$$

If the newly introduced category is a sub-category of only one of its more general categories and all its sub-categories are sub-categories of its super-category, then the relation is a taxonomy.

For summarising, a *classification scheme* is given by a set of categories (C), a sub-categorisation relation ($\leq$) and a sub-categorisation criterion ($\ll$). The first element allows classification, the second the expression of a taxonomy and the third the construction of that taxonomy. $\langle L_C \ll \rangle$ and $\langle C \leq \rangle$ defines two partial orders, the later being a subset of the former. This ensures that the taxonomy under which individuals are classified respects the extension inclusion.

From the properties of a given $\leq$ relation, it can be defined two important properties of the classification process [12]:

Univocity : a classification scheme is univocal iff

$$\forall i, \mid \mu_{\leq} Cl(i) \mid = 1.$$

This means that there is a unique minimal category for any individual. This property is important for representation languages in which an individual can only be attached under a unique category because it allows automatic classification (i.e. always attaching an individual under its minimal category).

Determination : a classification scheme is determinant iff

$$\forall i, \nu Cl(i) = \mu_{\leq} Cl(i).$$

This means that any individual is fully characterised by minimal classes. This is very important for biological taxonomies in which an individual is necessarily in a terminal taxon (there is no raw mammals — i.e. mammals which are not members of a sub-category — they are either primates, rodents...).

The two later properties can be affected with an objective or subjective character. This character is related with the incompleteness of a knowledge base. The *subjective* properties are in the subject (user)'s mind, the *objective* properties are in the object (category definitions). As a consequence, the system can check the objective properties but not the subjective ones (for example, OBR classically require that procedural attachments do not lead to side-effect but cannot check it: this is a subjective property). Knowing that a property must hold, even only subjectively, can be useful (for instance, when proving that the negation of that property objectively holds, the system can report the problem).

Moreover, the incompleteness of objects can lead to the incompleteness of the classification of that object (thus, even if the classification scheme is determinant, the classification process cannot reach the minimal class).

4.2 Consequences for classification in Tropes

Tropes taxonomies are hierarchies with no exhaustivity (the specialisations of a particular class are not required to cover the class) and subjective exclusivity (an object can only be classified in one direct specialisation of a class it is classified in; but the system cannot check it generally). Moreover, objects to be classified can be incomplete and, as a consequence of the descriptive character of classes, the fact that an object satisfies a class definition is not sufficient to attach it to the class. Hence, there is no determination and no univocity, so that automatic classification is impossible [12]. Such a result would have been dramatic for a purely definitional system, this is not the case for a descriptive one: the interpretation is given below.

4.2.1 Classification

Instance classification in Tropes classifies an object into a taxonomy. Thus, only C and $\leq$ are relevant. C is the set of classes in a particular viewpoint of a concept and $\leq$ is the specialisation relation. The object to be classified is necessarily an instance of the initial class c: an instance is attached to a class in each viewpoint (this could be the root class). Its field values are necessarily compliant with the class constraints (even if the value is unknown: this is an advantage of instantiation and attachment) and thus respect all the necessary and sufficient conditions over this class.

The sub-classification process goes down along the taxonomic tree by taking advantage of the sub-type relation between record types associated to classes in specialisation relation and the result given at §2.3: $E \subseteq P$. Thus it is able to classify the object in the more specialised classes in which the object respects the necessary conditions provided by the structure (and thus, are in P). So, the classification mechanism uses both sides of the specialisation relation: type checking for deciding if an object can belong to a class and set-based specialisation for deciding in which classes to sub-classify next.

The set $\mu Cl(i)$ is not necessarily a singleton, is not empty (since the initial class is in $Cl(i)$) and is not necessarily a set of classes with no further specialisation. This result is no more than formal and must be interpreted with regard to the semantics of Tropes. The interpretation of that result in a descriptive framework is that the object *can* belong to any class in which it has been classified. But, since the object only complies the structural constraints and that they are only necessary but not sufficient conditions, it is up to the user to decide in which class to put the object by an explicit statement. This is the only way to meet the set-based sufficient conditions.

4.2.2 Categorisation

For Tropes, the criterion relation ($\ll$) is the sub-type relation (<:) over the record-types associated to classes (note that this is different from the sub-categorisation relation which is specialisation). As for classification, sub-typing only determines necessary and not sufficient conditions for being a specialisation since the structure-based interpretation is not complete.

Thus, again, the interpretation of categorisation, according to the descriptive interpretation of classes, is that the categorised class *can* be a specialisation of all of its MGC and all of the MSC *can* be a specialisation of it. So, the user must choose the class of which c is a specialisation among $MGC(c)$, and the more specialised classes than c among those $MSC(c)$ which specialise the former one; but (s)he does not have to choose them all.

Hence, the resulting taxonomy is not necessary complete (w.r.t. the sub-type relation) but the categorisation process is (w.r.t. existing specialisation relation).

4.2.3 Extensions allowed by the weakness of descriptive semantics

Nothing requires that all viewpoints have the same interpretation. In a future development of Tropes, it is foreseen to add latitude: under some viewpoint, classification can be strictly based on sub-type (i.e. $P = E$). Moreover, if one retains this interpretation, (s)he can use specific classification algorithms which take advantage of this. As a consequence, this would allow automatic classification just like in TL. Moreover, if there were multi-specialisation in Tropes (the specialisation relation would be a direct acyclic graph and not a tree but its semantics would remain the same) it could be possible to build a complete taxonomy. This can be achieved by automatically making a class a specialisation of all of its MGC and all of its MSC a specialisation of it [7, 12]. As a consequence, the specialisation relation is immediately induced by the criterion (i.e. sub-typing) and the specialisation relation would be the sub-type relation. This would lead to the property of univocity.

As a conclusion, the use of classification and categorisation in Tropes is grounded on the semantics of the specialisation relation which, in turn, is grounded on the descriptive interpretation of classes. In fine, classification is still compatible with the descriptive interpretation of classes.

5 Related works

The three presented alternatives can be found in different representational systems:

Class/instance : semantic nets and early frame languages do not discriminate between classes and individuals. But in all but some OBR systems and OOL, the distinction is made. It is noteworthy that this confusion has been also made by early logicians and corrected later on. In TL, this distinction can be found in the difference between assertional and terminological components. However, this is changing due to the critics emitted in [11].

Ontology/taxonomy : there has not been a lot of works carried out on these aspects. OOL usually have purely ontological classes while TL have purely taxonomic ones. However, Rome [9] has been the first system to explicitly separate ontology from taxonomy. The system makes an object instance of an instantiation class borrowed from OOL and let it be classified under a representational class related to OBR. This system does not separate the base into different viewpoints and there is only one taxonomy. Not explicitly referring to the ontology/taxonomy distinction, Views [10] already offers the possibility to organise knowledge into multiple taxonomies. However, the semantics of relations is not given in a taxonomic setting (thus the user has to ground and implement classification) and the ontological apparatus is presented as useless platonism.

Definition/description : OBR usually consider (even if not explicitly) descriptive classes while TL consider definition (see above).

It is noteworthy that TROPES characteristics are a combination of those of OOL and post-frame languages. In fact, OOL classes have concept prerogatives (methods, instantiation, weak typing) with instances which cannot change class and OBR classes have class prerogatives (default inference, strong typing) with instances which can migrate. However, in the object-oriented view of TROPES the message-passing control mechanism and the specialisation of concepts are missing. It has been suggested that, in TROPES, concepts represent TL primitive descriptions and classes are constructed description. This is not adequate, since classes are interpreted descriptively.

6 Conclusion

A clear semantic account has been given of a system which separates the ontological aspects of classes from their taxonomic aspects. Moreover, the categorisation is grounded on the very general notion of description instead of definition. This generality prevents from using a classification system as straightforward as that of definitional systems but still allows a clear definition of classification which therefore can be used adequately. TROPES offers several other facilities, due to multi-viewpoints and the descriptive semantics, as for instance, the capability to build a new classification from objects with clustering or data analysis algorithms or that of using different classification algorithms on different viewpoints depending on particular reinforcement of the specialisation semantics.

The definition of TROPES semantics including inference mechanisms is not yet completed. The theoretical complexity of relevant algorithms promises to be very high but, from our experience, its practical impact is not critical, mainly because the complexity is on classes and not on objects (it remains high on recursive types but not practically). In fact, the problems are exactly the same as for definitional languages such as TL.

References

[1] Alfred Aho, Michael Garey, and Jeffrey Ullman. The transitive reduction of a directed graph. *SIAM journal of computing*, 1(2):131–137, 1972.

[2] Roberto Amadio and Luca Cardelli. Subtyping recursive types. In *Proc. 18th POPL, Orlando (FL US)*, pages 104–118. ACM, 1991.

[3] Ronald Brachman. On the epistemological status of semantic networks. In Nicholas Findler, editor, *Associatives networks: representation and use of knowledge by computers*, pages 3–50. Academic Press, New-York (NY US, 1979. (rep. in [6], pp191–215, 1985).

[4] Ronald Brachman. " lied about the trees" or, defaults and definitions in knowledge representation. *AI magazine*, 6(3):80–93, 1985.

[5] Ronald Brachman, Richard Fikes, and Hector Levesque. KRYPTON: a functional approach to knowledge representation. *IEEE Computer*, 16(10):30–36, 1983. (rev. in [6], pp412–429, 1985).

[6] Ronald Brachman and Hector Levesque, editors. *Readings in knowledge representation.* Morgan Kaufmann, Los Altos (CA US), 1985.

[7] Cécile Capponi and Mathias Chaillot. Construction incrémentale d'une base de classes correcte du point de vue des types. In *Proc. journées acquisition-validation-apprentissage, Saint-Raphaël (FR)*, 1993.

[8] Luca Cardelli and John Mitchell. Operations on records. *Mathematical structures in computer science*, 1(1):3–48, 1991.

[9] Bernard Carré and Gérard Comyn. On multiple classification, points of view and object evolution. In Jacques Demongeot, Thierry Hervé, Vincent Rialle, and Christophe Roche, editors, *Artificial intelligence and cognitive sciences*, Non linear systems in Biomedicine, pages 49–62. Manchester University Press, Manchester (GB), 1988.

[10] Harley Davis. VIEWS: multiple perspectives and structured objects in a knowledge representation language. Master's thesis, MIT, Cambridge (MA US), 1987.

[11] Jon Doyle and Ramesh Patil. Two theses on knowledge representation: language restriction, taxonomic classification, and the utility of representation services. *Artificial intelligence*, 48(3):261–297, 1991.

[12] Jérôme Euzenat. Une définition abstraite de la classification et son application aux taxonomies d'objets. In *Proc. 2nd journées représentation par objets, La Grande Motte (FR)*, 1993. to appear.

[13] Robert Filman. Reasoning with worlds and truth maintenance in a knowledge-based programming environment. *Communication of the ACM*, 31(4):382–401, 1988.

[14] Olga Mariño, François Rechenmann, and Patrice Uvietta. Multiple perspectives and classification mechanism in object-oriented representation. In *Proc. 9th ECAI, Stockholm (SE)*, pages 425–430, 1990.

[15] Gérald Masini, Amedeo Napoli, Dominique Colnet, Daniel Léonard, and Karl Tombre. *Les langages à objets*. InterÉditions, Paris (FR), 1989. (tr. eng. Object-oriented languages, Academic press, London (GB), 1991).

[16] Marvin Minsky. A framework for representing knowledge. Technical Report AI-memo 306, MIT, Cambridge (MA US), 1974. (rep. in Patrick Winston (ed.), The psychology of computer vision, pp211–277, Mac Graw-Hill, New York (NY US), 1975; rep. in [6], pp245–262, 1985).

[17] Amedeo Napoli and Roland Ducournau. Subsumption in object-based representations. In *Proc. ERCIM workshop on theoretical and practical aspects of knowledge representation, Pisa (IT)*, pages 1–9, 1992. (ERCIM report 92-W001).

[18] Bernhard Nebel. Reasoning and revision in hybrid representation systems. *Lecture notes in computer science (lecture notes in artificial intelligence)*, 422, 1990.

[19] Bernhard Nebel. Terminological cycles: semantics and computational properties. In John Sowa, editor, *Principles of semantic networks: exploration in the representation of knowledge*, pages 45–94. Morgan-Kauffman, San-Mateo (CA US), 1991.

[20] Jean-François Puget and Patrick Albert. SOLVER: contraints + objects = descriptions. In *Proc. IJCAI93 workshop on object-based representation systems, Chambery (FR)*, 1993. this volume.

[21] François Rechenmann. Integrating procedural knowledge: procedural attachment revisited. In *Proc. 25th INRIA anniversary workshop on knowledge representation, Paris (FR)*, 1992. to appear.

[22] William Woods. What's in a link: foundations for semantic networks. In Albert Collins Daniel Bobrow, editor, *Representation and understanding: studies in cognitive science*, pages 35–82. New-York Academic Press, New-York (NY US), 1975. (rep. in [6], pp217–241, 1985).

[23] William Woods. Understanding subsumption and taxonomy: a framework for progress. In John Sowa, editor, *Principles of semantic networks: exploration in the representation of knowledge*, pages 45–94. Morgan-Kauffman, San-Mateo (CA US), 1991.