HYBRIS nr 35 (2016)
ISSN: 1689-4286

ALEKSANDER GEMEL
UNIVERSITY OF LODZ

TADAO ISHII
NIIGATA UNIVERSITY OF INFORMATION & INTERNATIONAL STUDIES

CONCEPTUAL SPACES IN OBJECT-ORIENTED FRAMEWORK

Despite an enormous progress in the field of cognitive and computer sciences in recent time, the problem of adequate and computationally most optimal representation of human cognitive processes in artificial systems is still a subject of numerous disputes. As Gärdenfors noted, due to proper methodological structure of cognitive science, the vast majority of cognitive representation designs is implemented in a symbolic or connectionist approach [Gärdenfors, 2000]. Of course, each of these approaches presents a different vision of modeling techniques of cognitive representations. According to symbolic approach, modeling of cognitive processes is based on describing them in terms of computational operations on abstract symbols [Gärdenfors 2000: §2.2]. While the second approach boils down to modeling cognitive processes using artificial neural networks [Gärdenfors 2000: §2.3]. Despite the fact that – according to Gärdenfors – both approaches are complementary and the main difference between them lies only in the matter of describing differing levels of mental phenomena, both have a common drawback. Namely, they are unfortunately not very sensitive to the level of conceptual representations, which after all is one of the key layers of the human cognitive system. Moreover, in the case of both models, it is extremely difficult to provide an explanation of the genesis of conceptual layers, which would be in accordance with the principle of cognitive economy. The conceptual spaces theory developed by

Gärdenfors confronts these difficulties, providing a model of adequate representation of a conceptual layer structure and its genesis.

However, this does not mean that Gärdenfors considers the symbolic and sub-symbolic approaches as deprived of advantages, and that the conceptual spaces model is kind of a theoretical competition for them. On the contrary, their unquestionable advantage is the ease with which their representations can be implemented in information systems demonstrating the considerable usefulness in designing cognitive architectures and learning systems. This feature is difficult to ignore since one of the main demands of cognitive science involves implementation of cognitive processes in a computational model of information processing [Gärdenfors, 2015]. Moreover, Gärdenfors' proposal is intended to be a mid-level complementation of cognitive representational structure of human (*i.e.* consistent with both paradigms, and fully translatable into each of them). It is also vital that the conceptual spaces model was open to a range of wide possibilities of implementation, and aligned with programming languages. This text is devoted to the issue of this coherence. We will attempt to show that the middle level of mental phenomena representation in a conceptual spaces framework is consistent with the object-oriented programming paradigm. The presented solution is also appropriate for the purpose of representing the conceptual structure of human categories, and very intuitive from the perspective of artificial systems programming.

The first part of the text is a short psychological description of the conceptual categorization process of humans and of the structure of their mental representation. The second part presents the conceptual spaces modeling strategy, which is in line with the psychological characteristics of the human cognitive apparatus. The third part contains a summary of the main principles of object-oriented programming, and aims to express the concept of cognitive domain in terms of classes of objects. Finally, the fourth and the fifth part are devoted to two implementational problems of the conceptual spaces framework in an object-oriented model, *i.e.* to the issue of the prototype category structure and to the problem of vagueness respectively.

1 Cognitive characteristics of category structure

It is a truism to say that cognitive architecture should provide the possibly simplest and most intuitive methods of cognitive processes implementation. An even greater truism is to say that it should appropriately describe features of human cognition, and not impose its own structure in the process of modeling. In practice, however, this compromise between transparency of cognitive architecture and its structural and implementational properties is often difficult to achieve. Achieving it must be based on the one hand on the research results concerning human cognitive system, and on the other, it must rely on choosing the most consistent modeling technique to the structure of that system. Certainly, this general outline of implementation of human cognitive capacities also applies to the specific analysis of cognitive processes. This means that research on a computer model of human category structure should therefore also come out strictly from psychological analysis; it is then necessary to choose adequate and possibly transparent modeling techniques to address the results of this analysis. Within the research approach that is most optimal from an implementational point of view, conceptual structure of human categories shouldn't be treated as a purely philosophical *a priori* construct, detached from human cognitive ability. Instead, as Rosch postulates, it should be constructed in accordance with the rules governing scientific methodology, *i.e.* on the basis of results of research and experimentation, and by taking into account the characteristics of the human cognitive apparatus [Rosch, 1975].

In the field of research on the process of categorization, this approach, however, is relatively new. The structure of the conceptual category of human has been treated, in fact, in accordance with the purely structuralist model for a long period of time. You can even say that for over two thousand years the reflection on the nature of human reason has been done entirely under the aegis of the so-called classical category theory[1]. The main assumption of the classical theory of categorization is that categories are defined by necessary and sufficient conditions; therefore, it is sometimes also referred to as the model of

[1] Aristotle is considered to be the father of this theory, hence it is sometimes called 'Aristotelian theory of categorization'.

necessary and sufficient conditions (NSC). This theory was considered to be so evident that even a possibility of a different perspective on how to categorize was difficult to imagine. The situation is aptly summarized by Lakoff:

> "From the time of Aristotle to the later work of Wittgenstein, categories were thought be well understood and unproblematic. [...] This classical theory was not the result of empirical study. It was not even a subject of major debate. It was a philosophical position arrived at on the basis of a priori speculation. Over the centuries it simply became part of the background assumptions taken for granted in most scholarly disciplines. In fact, until very recently, the classical theory of categories was not even thought of as a theory. It was taught in most disciplines not as an empirical hypothesis but as an unquestionable, definitional truth" [Lakoff, 1987].

In the twentieth century classical theory of categorization became subject to criticism. The impulse came from Wittgenstein's philosophy and the experiments on categorization in the field of cognitive psychology conducted by Rosch and colleagues [Rosch, 1975, Mervis & Rosch, 1981]. In a series of experiments Rosch showed that the process of categorization was based on human neurophysiology and sensorimotor activity, that it was deeply connected with the process of constructing mental images and methods of organization of a memory material, and even that it was associated with processes of communication. Moreover, these studies have significantly influenced the scientific perception of the shape of human structure of concepts and categories.

According to the structuralist theory, construction of categories is determined solely by the infra-linguistic system of oppositional mutual relations. This means that the meaning of a unit within a system is determined by the place which it occupies in the language system of differences. A structuralist categorical outline for a sixteen-element set would thus look as follows:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Schema 1: Structuralist categorical outline.

Given a set consisting of *n* elements, each element is defined as a conjunction of negations of other elements:

$$\exists x\ (x \in \{x,y,z,\ldots, n\}: x = \neg y \wedge \neg z \wedge \ldots \wedge \neg n), \text{ for } n \neq \infty$$

According to this outline, for example some specific color name in the color domain is therefore defined as the sum of negations of all other color names occurring in the system of language. Consequently, all of the elements of cognitive domains (*e.g.* colors) are equivalent (*i.e.* there are no highlighted points in the system) since only the relations to other components define each element. This means that the boundaries of every category are sharply defined as connectives of conjunction and negation defined in the classical two-argument logic provide for their demarcation. Sharp boundaries of categories reflect the unambiguity in defining the relationship of a given object's membership. Relation of membership is therefore a "zero-one issue" in structuralist terms and – as befits a typical representative of the classical theory of categorization – is determined by a set of necessary and sufficient conditions. In consequence, this means that exactly just like there were no highlighted categories within a domain, there are also no highlighted elements within categories. Therefore, speaking of a better or worse example of given color (for example, 'red') is completely meaningless from the perspective of structuralism since all elements of the category are in fact equivalent.

Criticism of the classical concept of categorization has led to the development of the so-called prototype theory of categorization. The main arguments in favor of the latter have come from the results of psychological experiments [Rosch, 1977; Berlin & Kay 1969] which undermined the main assumptions of the NSC model. The experiments were conducted by Rosch and produced results showing clearly that the respondents distinguished superior and inferior members within the same category. In the case of NSC classification model, such phenomenon should not take place since the classical categorization process is based on common characteristics of all elements within a category. This change in the categorization model has also significantly influenced the vision of the human conceptual structure. Experimental studies clearly show that for each category we can highlight central and peripheral elements, which differ in terms of the degree of categorical membership (*vel.* representativeness). The central element, also known as the prototype, is a kind of touchstone of categorical membership. Elements more similar to the prototype are situated closer to the center of the category, and characterized by a greater degree of membership, while those less similar to the prototype take a more peripheral location in the internal structure of the category. The similarity construed in line with the assumptions of psychology thus becomes one of the most important concepts in the process of constitution of the human conceptual structure. In short, the question of psychologically appropriate representation of similarity seems to be crucial not only from the point of view of methodology of cognitive science and procedures of prototype categorizing, but also from the perspective of cognition *tout court*. The only question is how to understand it.

Most approaches rely on cumulative model of similarity. According to this model, the number of common features shared with the prototype determines the degree of similarity to the prototype. This is a result of a prototype structure organization which is constituted with the involvement of quantitative effects of perception. As Evans puts it: "Prototype structure thus concerns the degree to which redundancy in the category members is employed in categorization, by virtue of providing a salient set of attributes that organize the category" [Evans, 2007, 176]. However, a quantitative model based on classification of attributes generates numerous problems. First of all, as it is evidenced by numerous empirical studies, a sheer amount of common features does not play as important role as co-occurrence of certain specific groups of features in the process of assessing the similarity. Second, the presence of such a bundle of qualities in all members of a particular category may be considered as a necessary and sufficient condition of category membership, which leads to transformation of the prototype theory back into the classical model. Thus Gärdenfors, unlike Evans, shifts towards the geometric model of similarity representation and conceptual structure of the mind. This model is an integral part of his theory of conceptual spaces, the main assumptions of which are discussed in the next section.

2 The model of conceptual structure in Gärdenfors' conceptual spaces theory

The central idea behind conceptual spaces is the belief that meaning can be described in terms of organized abstract spatial structures [Gärdenfors 2000 & 2014]. Notions used to such modeled meanings are then terms borrowed from geometry and vector algebra. The basic building blocks of every conceptual space are the so-called qualitative dimensions, which form a kind of theoretical framework used to set properties of objects and to define the relationships occurring between them. Dimensional reconstruction of quality is performed using a technique of multidimensional scaling (MDS) [on a theoretical basis for MDS *cf*. Shepard, 1962; Coombs, 1964]. The fundamental role of dimension in a conceptual space is representation of a wide variety of modeled object qualities in different cognitive domains. Therefore the coordinates of points within conceptual spaces represent individual cases of a given dimension, *e.g.* a certain temperature, a certain weight, tone, *etc*. Such points represent objects with specific qualities defined by their coordinates in space. The quality dimensions are *infra*-linguistic, which means that the quality represented by them does not need to be expressible in the form of a language or a symbolic code. As Gärdenfors puts it:

> "Humans and other animals can represent the qualities of objects, for example when planning an action, without presuming an internal language or another symbolic system in which these qualities are expressed. As a consequence, I claim that the quality dimensions of conceptual spaces are *independent* of symbolic representations and more fundamental than these" [Gärdenfors 2000: 11].

Gärdenfors uses the concept of dimension construed in such way to introduce the notion of cognitive domain, which – according to its conceptualization – is an integrated bundle of dimensions. Dimension is integrated when it necessarily occurs in conjunction with another integrated dimension. Examples of integrated dimensions may be, for instance, hue, brightness, and intensity because every representative of the color domain that features a specific value assigned in one dimension (*e.g.* brightness) must also have a certain value in other dimensions (*i.e.* a certain shade and intensity). The color domain as a three-dimensional conceptual space can be thus presented in a graphic form like the one provided in *Figure 1*. It consists of three quality dimensions *i.e.* brightness (growing vertically bottom–up), intensity (growing horizontally from the center of the cone to its rims), and shade (changing clockwise on a circle). The most contrasting shades lie at opposite points of the circle.

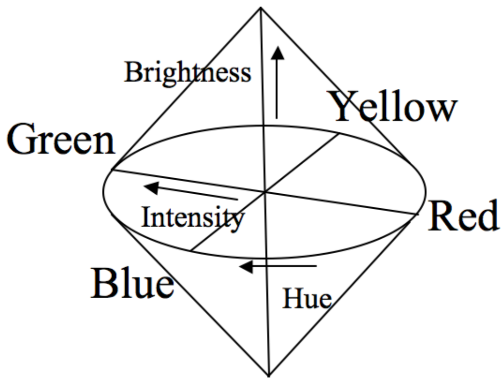

Figure 1: Representation of color domain in conceptual spaces model (Authors' work on the basis of [Gärdenfors, 2014, p. 23].

Both the properties and the concepts in the conceptual space are, according to Gärdenfors, represented by convex sets whose layout is determined by the location of the prototype of a given element (concept/property) and the prototypes of other surrounding elements in a particular cognitive domain. The solution proposed by Gärdenfors is in line with the findings of the prototype theory of categorization [Rosch, 1975 & 1978; Mervis & Rosch, 1981; Lakoff, 1987]. Representation of concepts as convex polygons in space allows for a more or less central location of a specific point within a given polygon. A central location of a point reflects prototypical characteristic of the object represented thereby. The prototype theory of categorization plays an essential part in Gärdenfors' model because it makes it possible to constitute the conceptual structure of the cognitive domain. Mapping of the category structure of a specific domain in the model proposed by Gärdenfors is carried out with the aid of a topological tool called Voronoi tessellation. It is a method of division of n-dimensional space into parts, using equal distance between the highlighted points, which, in accordance with the arguments proposed by Gärdenfors, create a collection of prototypes or objects with the highest degree of prominence in a given domain. Formally speaking, Voronoi tessellation proceeds according to the following formula:

$$VorP\ (p) = \{x \in E : \forall q \in P, d(x,p) \leq d(x,q)\}$$

In this formula, *d* is defined on the Euclidean space function of distance representing the degree of similarity between stimuli.

For an eight-element set of points $P = \{p_1, ..., p_8\}$ belonging to the Euclidean space *E*, the Voronoi area assigns a certain element p_1 from the set *P*, which is called the set of all points that are closer to point p_1 than to any other element from *P*. As a result, for an 8-element set *P* we will get the following division of space:

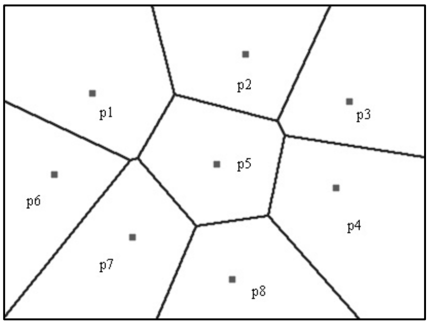

Figure 2: Voronoi tessellation for an 8-element set P (authors' work).

It is important to highlight that the model of the conceptual space shown in *Figure 2* leaves no room for existence of borderline cases; borders are, in fact, decidedly sharp since they are determined by the function of distance between two contiguous points belonging to different Voronoi areas. Of course, in a natural language, a large amount of concepts is vague as a matter of fact. However, it is possible to reconcile the representation of prototype conceptual structure in Gärdenfors' model with the phenomenon of vagueness, which was one of the major spots of concern in the classical theory of categorization. Let us recall that in the classical theory of categorization membership function is based on a set of necessary and sufficient conditions. This form of membership function doesn't make it possible to grade the degree of membership. In the prototype theory, however, by identifying the membership measure with the degree of prototype resemblance, one can blur boundary of categories thus creating a place for border cases.

Blurring the boundaries of categories in Gärdenfors' theory is possible by way of extension of its conceptual model. It involves multiple overlapping of tessellation (*i.e.* so-called cluster tessellation) made on the basis of vague prototypes [Douven & al. 2013; Decock & Douven, 2014]. In other words, in the case of vague terms, it is more adequate to consider a set P* of clusters of points $\{\{p1_1, \ldots, p1_n\}, \{p2_1, \ldots, p2_n\}, \ldots, \{pn_1, \ldots, pn_n\}\}$ instead of a set P of points. Decock and Douven [2014] use the standard example of colors as involving such cluster modeling: color names are vague, and modeling the conceptual space with clusters of prototypes is specific to vague concepts. It is traditionally assumed that when a partition of a space is generated by a set P* of clusters of points, these clusters form circles. That is, prototypical areas are circular, as shown in *Figure 3*.

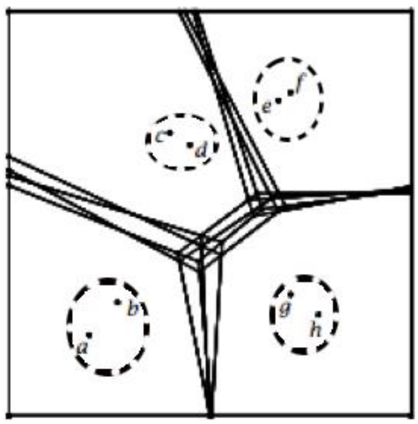

Figure 3: Voronoi Diagrams generated by P* (authors' work on the basis of [Douven & al. 2013])

A tessellation made by a cluster of points formed in a circle (so-called vague prototypes) allows blurring of boundaries between concepts in a given space. Every point located on the boundary of two or more terms is considered a borderline case of these predicates. The tessellation formed of vague prototypes is presented in *Figure 4*:

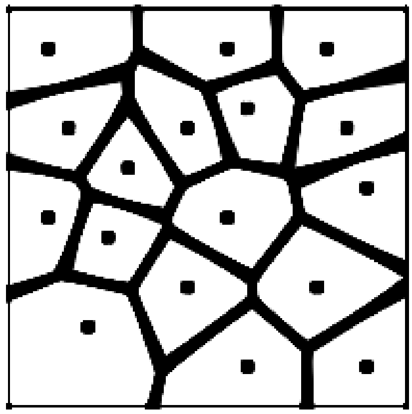

Figure 4: Use of *P** to model vague concepts [Decock & Douven, 2014]

3 Object-oriented data architecture

As we have argued in section 2 of this paper, cognitive architecture is supposed to be the most possibly intuitive method of cognitive processes implementation on the one hand, and should not impose its formal structure during the modeling process on the modeled cognitive system on the other. It seems that the key to achieve this goal is to select modeling techniques that will be most consistent with the structure of the modeled system. The approach that we propose in this paper, and which in our opinion is considered to be most consistent with Gärdenfors' implementational model, is object-oriented design. The greatest advantage of object-oriented programming, design and analysis is compatibility of such approach with the reality. Moreover, as research shows, human brain is naturally best-suited to such approach in information processing [Sheetz & Tegarden, 2001]. Second, object-oriented approach seems to be very easily reconciled with the conceptual spaces model, which allows very efficient and intuitive computer modeling of cognitive processes.

The essential concept of object-oriented paradigm is very simple. The system of knowledge, or the cognitive system of concepts in the real world is expressed as a formal model in terms of objects and connections between them. Then, a model representing objects and their connections is mapped within a computer system.

The first step of this process can be called 'modeling routine'; the second step is the process of simulation.

Of course, the notion of "real world" requires some explanation in the context of this paper. In the case of cognitive architecture design, which is intended to be a reflection of the conceptual framework of human, the notion of "real world" applies to mental or conceptual reality of the mind. In fact, the concept of "real world" in an object-oriented methodology doesn't necessarily represent an external material world, but a reality modeled by an object-oriented language. Therefore, in the case of our endeavor, the mental reality of the human categorization system is initially a specific model of categories with a set of certain relations assigned to it. In some sense we can speak about a dual modeling process. On the one hand, it constitutes a description of the human cognitive system in a conceptual spaces framework, which is subsequently described as a system of knowledge in terms of an object-oriented methodology. Simply speaking, the goal of this text is to show, that an object-oriented model is in line with the conceptual space framework. In consequence, the latter can be therefore seen as a first step of object-oriented implementation. The second step consists in translating the CS framework into a model suitable for its simulation in the computer system. The process of simulation itself is not covered by the scope of this paper.

The basic programming unit in an object-oriented paradigm is not a *function* like in the case of procedural or functional paradigm (in which all computations are performed by applying functions), but a *class* representing a collection of similar real-world objects. In the procedural approach, the program is divided into self-contained and maintainable parts called 'modules'. Every module contains procedures, *i.e.* self-contained codes that carry out single tasks. A function in procedural programming is a self-contained code returning value to the calling procedure. In consequence, the distinction mark of a procedural paradigm is separation of program's data from instructions that manipulate it. On the contrary, an object-oriented paradigm binds together data and the instructions. The latter manipulate the former into the class. In OOP, class is an encapsulated definition of a collection of similar real-world objects. Therefore, in some way, class represents the conceptual category in human cognitive system. The application in OOP is modeled as a collection of related interacting and working *objects*. That's why OOP is considered to be a very intuitional programming approach – it describes informational system as it exists in real life based on interactions among real objects. Moreover, the programming process starts with a very common everyday human capacity, *i.e.* abstraction of useful real-world objects and classes.

The basic concepts of OOP are the notions of object and class. These two concepts are so closely connected that trying to explain them really seems like a chicken-and-egg dilemma. In real world it is impossible to define a class without using the term of object – and *vice versa.* However, in the case of OOP, the basic programming unit is the notion of class. Objects are always instantiations of a class. Due to the fact of objects being created from classes, they can be, as a matter of fact, distributed individually or as part of a library. However, it is classes that are actually pieces of a code, which means that they are the basic building blocks defining objects. In consequence, objects can be seen as merely conceptual units of both state and behavior. The state of an object is referred to through its attributes or properties, and the behavior of an object is defined by its set of methods. Compared to object, class can be thought of as a sort of higher-level data type.

Let us note that the concept of class is in line with the notion of domain in Gärdenfors' proposal. The conceptual space of a given cognitive domain is constructed from several quality dimensions connected to each other in a necessary relation. The concept of class in an object-oriented paradigm is easy to consider together with the notion of cognitive domain in a conceptual space. For instance, a color domain defined by three dimensions (*i.e.* hue, brightness, intensity) can be easy translated into a class of objects (colors) defined by these attributes.

All the elements of a cognitive domain (for instance, specific colors) represented by points in a conceptual space are instances of a given cognitive domain (*i.e.* color domain), just like objects that are instantiations of a given class in OO language. In other words, all the basic elements of both frameworks are mutually translatable. This applies also to values associated with objects and points. The values assigned to objects are in fact directly mapped to coordinates of points representing quality values relevant to a given class of attributes under consideration. However, these huge convergences are not sufficient to fully and naturally implement the cognitive process of shaping the structure of categories in an object-oriented language. The latter is more similar to the Aristotelian model of categorization rather than to the prototype one, which remains in accordance with Gärdenfors' proposal.

4 The problem of the prototype structure of an object-oriented language

As we said before, since in a traditional object-oriented paradigm class provides an abstract definition of objects, each object has to be defined as an instance of its class. In other words, class defines shared data structures and methods for an entire collection of objects. This means that most object-oriented programming languages are based on the classic categorization model in which objects of one class are defined in terms of the same shared properties. Therefore, the classification model of an object-oriented language is essentially different from the process of categorization performed by humans, and includes limitations when it comes to implementation of conceptual spaces in an object-oriented framework.

There are, however, some modifications to this classical approach, allowing for more flexibility in defining objects in a system. One of such modifications is prototype-based object-oriented programing, which in contrast to defining objects by class supports a more direct method of object creation. In prototypal object systems, objects are seen as a simple list of properties rather than merely instances of a class, with a special reference to the prototype from which they inherit their behavior. In other words, the fundamental difference between classical and prototypal object systems consists in the ontological status of a given object. While classical objects are defined abstractly in terms of the class from which they inherit their characteristics, prototypal objects are defined *concretely* as specific objects and inherit their characteristics from other specific objects. Moreover, prototype-based languages eliminate the dual nature of information system that requires at least two fundamental constructs – classes and objects, and facilitate direct creation and manipulation of objects. They are more elegant, and closer to the recent cognitive theory of categorization than class-based languages. As Taivalsaari puts it:

> "Prototype-based languages are conceptually elegant and possess many other characteristics that make them appealing. These languages are also seemingly closer to the prototype theory presented by cognitive psychologists and philosophers. For instance, the ability to modify and evolve objects at the level of individual objects reduces the need for a priori classification and encourages a more iterative programming and design style. In general, when working with prototypes, one typically chooses not to categorize, but to exploit alikeness. Rather than dealing with abstract descriptions of concepts (intensions), the designer is faced with concrete realizations of those concepts. Consequently, design is driven by evaluation in the context of examples: designers run their solutions to evaluate them in the context of some input to the program" [Taivalsaari, 1997].

Apart from the abovementioned unquestionably practical reasons, prototype object-oriented systems seem to be simply ideal for modeling conceptual structures of categories proposed by Gärdenfors. It becomes even more evident when we consider the process of category structure creation in a prototype-based object-oriented language. This process refers to methods of creation of new objects. While class-based languages parameterize objects using templates from which other instances of objects are generated, prototype-based languages adopt a different approach to object generation. Instead of using object descriptions beforehand, there are stock objects generated from prototypes, and customized later on. The crucial feature of prototype-based languages is the lack of an entity that represents a class of instances. It is the convention, or rather the resemblance, that plays the key part in this context.

The so-called "cloning" is the basic mechanism used for instantiating prototypes in prototype-based languages. The product of a cloning process is something we may call a 'field' or 'shallow' copy of an object, *i.e.* something of the same external structure of a given object. As a result, the clone shares attributes of the prototype, but with independent state values. This process is consistent with Gärdenfors' model of category structure creation. Let us recall that in the case of the conceptual spaces model, domain is constituted by a set of a given prototype's essential features, in which all particular objects participate, but the actual values of each object's features are specified by the degree of a given object's similarity to the prototype. Using the set of prototypes in a certain domain, for instance, a set of base colors in the domain of color [Berlin & Kay, 1966], we can easily generate the categorical structure of the domain similar to Voronoi division of space presented in Gärdenfors' model by simply cloning objects resembling a given prototype.

5 The problem of object graded membership

The problem of the most efficient strategy for adding vagueness in an object-oriented data architecture is an issue well-known in formal literature. As Marin, Pons, and Vila have pointed out:

> „In the last few years, the object-oriented database model has been modified in order to incorporate vagueness. Consequently, fuzzy object-oriented database models have appeared. Vagueness has been studied on different levels: considering fuzzy attribute domains in the database, softening the idea of membership of an object to a class, relaxing superclass subclass relationships, and even trying to reflect the fuzzification in the behavior of the objects" [Marin, et. al. 2001, 863].

In order to deal with the issue of graded categorical membership, Marin et al. [2000] have introduced a concept of fuzzy type, and use it as a new way of managing fuzzy structures. Fuzzy type allows incorporating attributes with membership degrees determined by layers, where higher layers represent a greater degree of membership. Type is defined as a pair (***S, B***), where ***S*** (*Structure*) is the set of attributes or instance variables (***S*** is the part of a set of all attributes ***A***, *i.e.* $\boldsymbol{S} \subseteq \boldsymbol{A}$), which characterizes the structure of the type, and **B** (*Behavior*) is the set of methods that define its behavior. Considering the previous definitions, a class **C** is a pair (***T*, *O***), where **T** is the type of the class and **O** (*Objects)* is the set of all objects that adopt the structure and the behavior of a given class. Finally, similarly to the notion of a structure, we can define *fuzzy structure* as a fuzzy set defined over a set of all attributes in a previous model. In consequence, *fuzzy type* can by defined as a special kind of type (***S, B***) whose structural part ***S*** is a fuzzy structure (*Fuzz* ***S***, ***B***). In other words, the key to identify fuzzy type is to find the membership function associated with the structural component of a given type. For ***T*** associated with a given class **C,** it adopts the following form:

$$\mu_S: A \rightarrow [0,1]$$

In order to make a fuzzy object-oriented model consistent with conceptual space and, in consequence, with cognitive capacities of the human mind, it is necessary to define membership function in a more cognitive manner, *i.e.* in terms of similarity to prototype. Similarity function can't be expressed simply as a monotonically decreasing function of distance from prototype, which takes values in the real number interval between 0 and 1 inclusive. Function of that sort, as Osherson and Smith [1981] pointed out, is cognitively inadequate and makes wrong predictions about the conceptual structure of human categories. As they have shown, function of that sort can by defined as function *c* that for any prototype *p*, distance function *d,* and domain *A* to which *p* belongs satisfies the following condition:

$$\forall x,y \in A: d(x,p) \leq d(y,p) \rightarrow c(y) \leq c(x)$$

According to Osherson and Smith, function *c* lacks some threshold value of similarity to prototype, and hence fails to specify the concept boundaries; this condition can be satisfied with the aid of similarity function *M* introduced by Hampton [2007, 365]. Hampton's formal representation of similarity is consistent with the concept of graded categorical membership. Using function *M,* one is able to generate the structure of concept with determinate boundary region for membership and its lower and upper bounds, represented respectively by values S_L and S_H. If $S(x)$ is the measure of similarity of *x* to prototype, and S_T is the value where *M* equals 0.5, then *M* is defined as follows:

$$M(x) = \begin{cases} 0 & \text{if} \quad S_L \geq S(x); \\ 2\left(\dfrac{S(x) - S_L}{S_H - S_L}\right)^2 & \text{if} \quad S_T \geq S(x) > S_L; \\ 1 - 2\left(\dfrac{S_H - S(x)}{S_H - S_L}\right)^2 & \text{if} \quad S_H \geq S(x) > S_T; \\ 1 & \text{if} \quad S(x) > S_H. \end{cases}$$

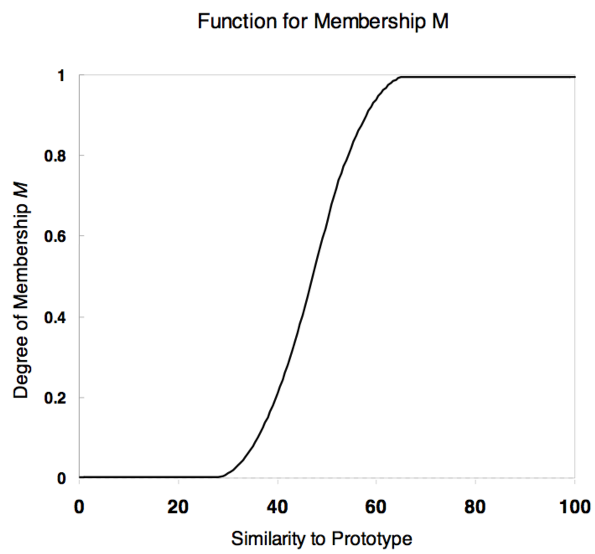

Figure 5: Function *M* with curved threshold between extensions and penumbra area [Hampton, 2007].

By applying Hampton's membership function to an object-oriented language with three boundary regions for similarity to prototype, every property in the model becomes affected by the membership degree. As a result of such operation, we obtain a type with a fuzzy structure set defined over the set of all possible attributes (*Figure 6*). The transition from a classical object-oriented language to a language enriched by the concept of fuzzy type introduced by Marin et al. is similar to the abovementioned transition from Voronoi tessellation of space to cluster tessellation proposed by Decock & Douven [Douven & *al.* 2013].

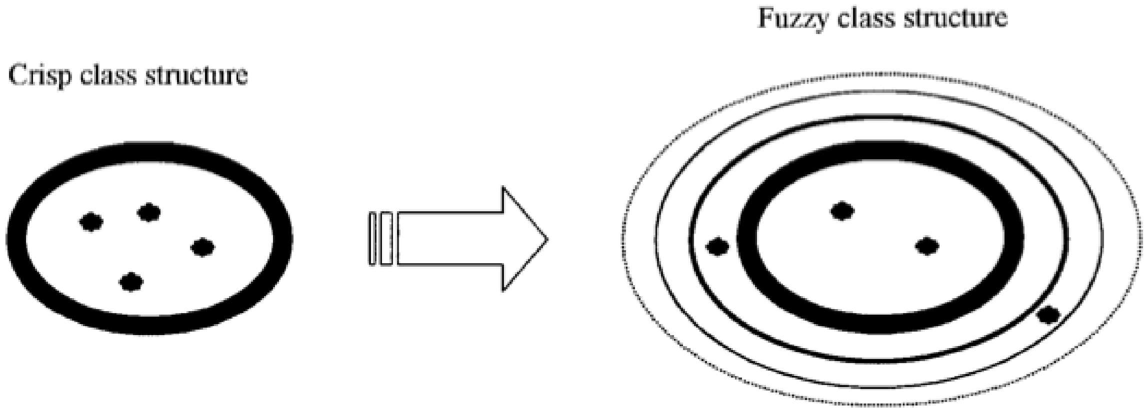

Figure 6: Applying fuzzy type to crisp class structure [Marin, et. al, 2000].

Finally, it is important to note that besides numerous structural convergences between object-oriented and conceptual spaces models, both approaches are also highly consistent with cognitive activities of humans. We don't claim that all cognitive activities are well-suited for both object-oriented and conceptual space approach. For instance, the famous Marr's model of vision is clearly inspired by procedural programming languages, as long as it is considered to be:

> "...a sequence of processes that are successively extracting visual information from one representation, organizing it, and making it explicit in another representation to be used by other processes. Viewed in this way it is conceptually convenient to treat vision as computationally modular and sequential" [Mayhew & Frisby, 1984].

However, as Mather rightly points out, some aspects of cortical physiology and some cognitive activities are consistent with OOP models - like, for instance, object-specific priming, multiple object tracking, inattentional blindness, or attentional effects in motion perception [Mather 2001]. The model of categorical structure acquisition given by conceptual spaces framework seems to be also one of the cognitive activities well-suited for object-oriented design. We at least hope that we have been able to show it. Consistency of both models and their intuitiveness seems to create an extremely wide range of possibilities of implementation of Gärdenfors' theory.

REFERENCES

Berlin Brent, Kay Paul, 1969, *Basic color terms*, Berkeley: University of California Press.

Coombs Clyde Hamilton, 1964, *A theory of data*, New York: Wiley.

Decock Lieven, Douven Igor, 2014, *What Is Graded Membership?* "Noûs", 48, pp. 653–682.

Douven Igor, Decock Lieven, Dietz Richard, Égré Paul, 2013, *Vagueness: A Conceptual Spaces Approach*, "Journal of Philosophical Logic" 42(1), pp. 137–160.

Evans Vyvyan, *A Glossary of Cognitive Linguistics*, Edinburgh: Edinburgh University Press

Gärdenfors Peter, 2000, *Conceptual spaces*, Cambridge: MIT Press.

Gärdenfors Peter, 2014, *Geometry of meaning – Semantics based on conceptual spaces.* Cambridge: MIT Press.

Gärdenfors Peter, 2015, *Cognitive science: from computers to ant hills as models of human thought* [in:] Gemel A., Łukowski P., Żukowski B., (ed.), *Cognition, Meaning and Action. Lodz-Lund Studies in Cognitive Science,* Lodz-Cracow: Lodz – Jagiellonian University Press.

Hampton A. James, 2007, *Typicality, Graded Membership, and Vagueness*, "Cognitive Science" 31(3), pp. 355-384.

Marin Nicolas, Pons Olga, Vila A. Maria, 2000, *Fuzzy Types: A New Concept of Type for Managing Vague Structures,* "International Journal of Intelligent Systems", Vol. 15, pp. 1061-1085.

Marin Nicolas, Pons Olga, Vila A. Maria, 2001, *A Strategy for Adding Fuzzy Types to an Object-Oriented Database System,* International Journal Of Intelligent Systems, Vol. 16, pp. 863-880.

Mather George, 2001, *Object-oriented models of cognitive processing,* "Trends in Cognitive Sciences", 5(5), pp. 182-184.

Mayhew John, Frisby John, 1984, *Computer vision*, [in:] O'Shea T., Eisenstadt M., (eds.) *Artificial Intelligence: Tools, Techniques, and Applications*, New York: Harper & Row pp. 301–357.

Mervis Catlin, Rosch Eleanor, 1981, *Categorization of natural objects*, „Annual Review of Psychology" 32, pp. 89-115.

Lakoff George, 1987, *Women, Fire and Dangerous things: What Categories Reveal about the Mind*, Chicago: University of Chicago Press.

Osherson N. Daniel, Smith, E. Edward, 1981, *On the Adequacy of Prototype Theory as a Theory of Concepts*, "Cognition" 29, pp. 259–288.

Rosch, Eleanor, 1975, *Cognitive Representation of Semantic Categories,* w: „Journal of Experimental Psychology", 104, pp. 192-233.

Shepard N. Roger, 1962, *The analysis of proximities: Multidimensional scaling with an unknown distance function*. *Part l*, „Psychometrika" 27, pp. 125-140.

Sheetz, Steven, Tegarden David, 2001, *Illustrating the cognitive consequences of object-oriented systems development,* "The Journal of Systems and Software", 59, pp. 163-179.

Taivalsaari Antero, 1997, *Classes vs. Prototypes: Some Philosophical and Historical Observations,* "Journal of Object-Oriented Programming" 10(7), pp. 44-50.

ABSTRACT

CONCEPTUAL SPACES IN OBJECT-ORIENTED FRAMEWORK

The aim of this paper is to show that the middle level of mental representations in a conceptual spaces framework is consistent with the OOP paradigm. We argue that conceptual spaces framework together with vague prototype theory of categorization appears to be the most suitable solution for modeling the cognitive apparatus of humans, and that the OOP paradigm can be easily and intuitively reconciled with this framework. First, we show that the prototype-based OOP approach is consistent with Gärdenfors' model in terms of structural coherence. Second, we argue that the product of cloning process in a prototype-based model is in line with the structure of categories in Gärdenfors' proposal. Finally, in order to make the fuzzy object-oriented model consistent with conceptual space, we demonstrate how to define membership function in a more cognitive manner, *i.e.* in terms of similarity to prototype.

KEYWORDS: Conceptual spaces, Object-oriented framework, prototype theory of categorization, vagueness, cognitive architecture.

SŁOWA KLUCZOWE: Przestrzenie pojęciowe, programowanie zorientowane obiektowo, prototypowa teoria kategoryzacji, nieostrość, architektura poznawcza.