

Neuro Object Oriented Programming Approach and Design (R - Formula)

Prateek Chaudhary
¹, (Pursuing MCA) IP University

ABSTRACT : Our object oriented programming approach have great ability to improve the programming behavior for modern system and software engineering but it does not give the proper interaction of real world .In real world , programming required powerful interlinking among properties and characteristics towards the various objects. Basically this approach of programming gives the better presentation of object with real world and provide the better relationship among the objects. I have explained the new concept of my neuro object oriented approach .This approach contains many new features like originty , new concept of inheritance , object relation with dimensions , originty relation with dimensions and time , category of NOOPA like high order thinking object and low order thinking object , differentiation model for achieving the various requirements from the user and a formula through which we can calculate the requirements.

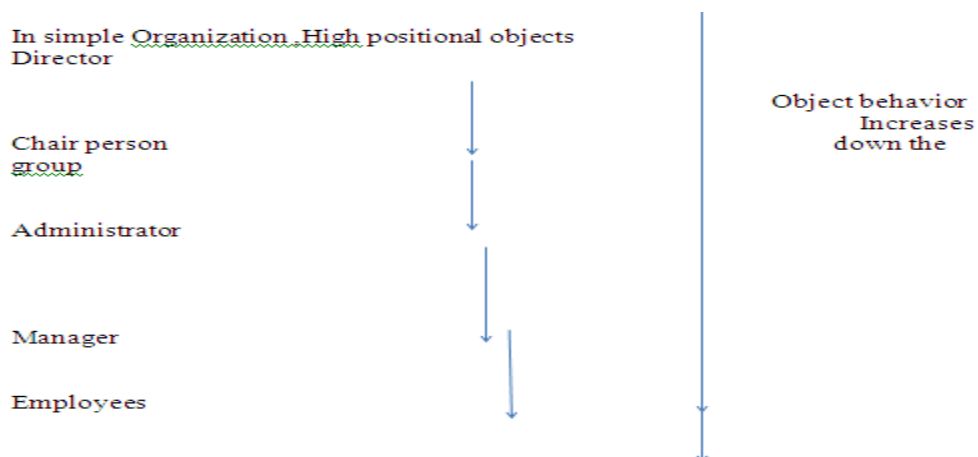
KEYWORDS: Homologous , Heterogenous , Dimension ,Impulsive object , Controlled object , Main object ,Interface object

I. INTRODUCTION

Most respectfully , I am Prateek Chaudhary from Delhi and Currently doing MCA from IP University .I have wrote some research papers on oops concept and I work as a trainer, consultant, writer and researcher in early childhood education. I have worked previously as a Mathematics and Science teacher in primary and early childhood teacher education and also doing Private teaching in various private institutes My professional interests are focused around research in science and also in mechanical engineering and also in OOPS concept . I have also made some design of single tyre vehicles and write some other object oriented approach for better system and software engineering .sir/mam this reference was given by my college faculty . and I have attached my research file . please sir , one time see my concept and new approach towards OOPS.

II. NEURO OBJECT ORIENTED PROGRAMMING APPROACH AND DESIGN

According to Neuro object oriented programming approach and design , “ originity of objects and its properties are play a main role in the system environment development On the basis of NOOPAD , we will explain the all positional behavior of object inside the various dimensions .Object position is also depends on the dimensions where we can define the coordinate of the objects .If we have to find the origin of object then we must follow the concept of originity to determining the object inside the dimension and time .This feature of object with originity explain the requirements of object with dimension and time .To determine the role of object inside the dimensions and time firstly we have to calculate the position of object inside the organization . the position of dimension means the level of object inside the organization .



2.1.Requirement Gathering procedures =According to the problem of getting the requirement , behavior , properties and information related to object , can be calculated using the mathematical calculation consider a object (O) have R (properties , behavior and information) , then according to object relation with their respective nature .

2.2. According to the requirement law, “ It contains behaviors , information and presentation of an object and its is directly proportional to the amount of properties present in the various dimensions “.

$$nO = D * (n(n+1)/2)^n$$

where D is dimensional constant , which depends on the dimensional natures

$$D = m * (2^m - 1)$$

Note = every object contain at least two properties ,

R = properties = m - 1

R = requirement

Therefore ,

So the final formula is ,

$$n(O) = m * (2^m - 1) * (n(n+1)/2)$$

in one dimension , it is not possible to determine the requirement because , one dimension contain single length , breadth , or width or time , and object always created in two dimension (or minimum two dimensions) .

therefore ,

R (requirement) = m - 1

So , according to requirement need with dimensions ,

$$D = 2^m - 1 = 2^R$$

Where ,

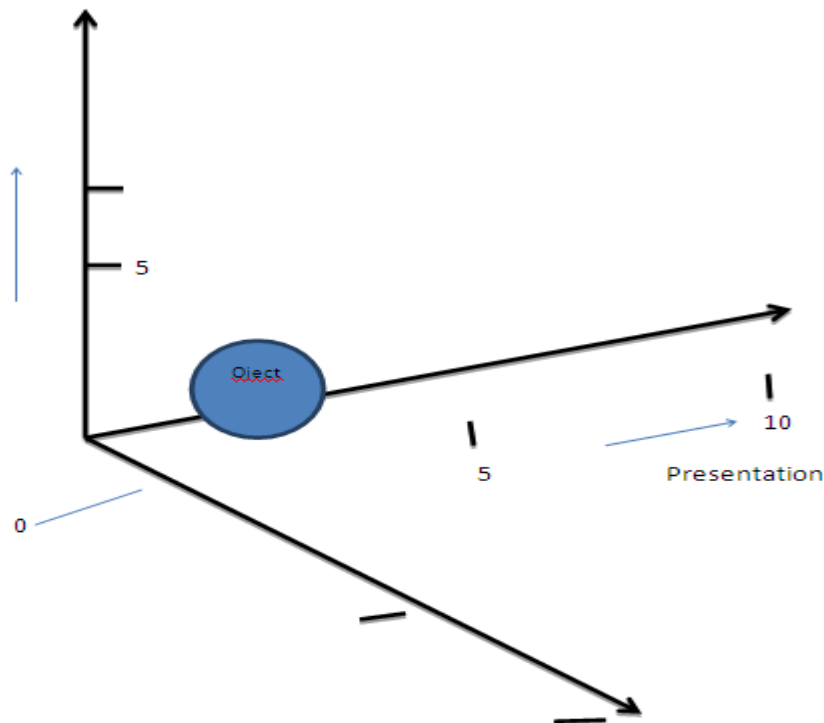
D = Dimensional constant

M - 1 = Maximun requirements

O = objects

Behavior

10



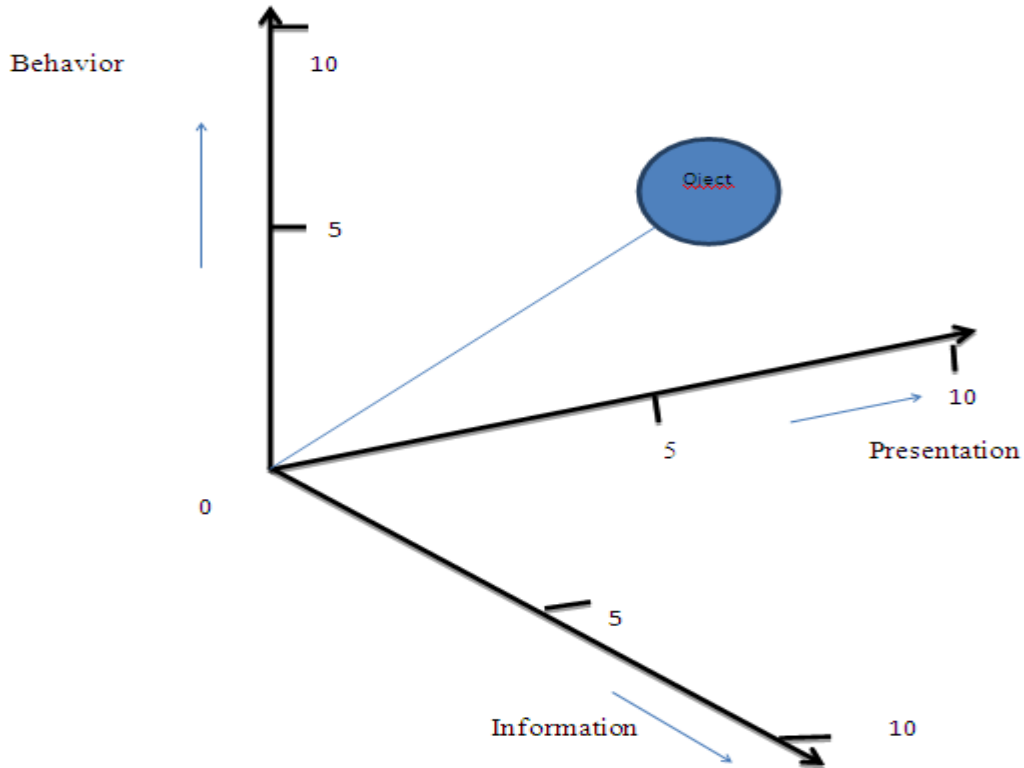
III. INFORMATION

3.1. According to the graph, object shows different characteristics in three given dimensions.

- [1] Information,
- [2] Presentation
- [3] Behavior.

From range 0 to 5,

Object complexity is minimum with limited suitable amount of requirements.



Object complexity is maximum from 5 to 10 range with large amount of requirements.

Prove = To calculate the object properties with their dimension always contain some functional requirement and this requirements effect the whole environment of system.

Solution = let O be the object,

$$O \propto 1 + 2 + 3 + \dots + n$$

According to the arithmetic progression (A.P) of sum of N – natural numbers,

$$\dots [S_n = n/2 (2a + (n-1) d)]$$

$$a=1, \quad d = 2 - 1 = 1$$

$$O \propto n (n + 1) / 2$$

Then, in normal life a simple entity (object) can perform $n (n + 1) / 2$ requirement with some properties,

If we consider n – object,

Then,

$$nO \propto (n (n + 1) / 2) ^n$$

so,

the main factor which affect the object relation with their requirements

$$nO = D * (n (n + 1) / 2) ^n$$

where D = proportionality constant (requirement constant)

so, its necessary to calculate the dimensional nature,

For every dimension contain at least two properties ,

$D = 1$, for one dimension

$D = 2$, for two dimension

$D = n$, For $m - 1$ dimension

$D = m * (2^{m-1})$

Note =every object contain at least two properties ,

For one dimension , $2^0 = 1 * 1$, object with zero property in 2's power

For two dimension , $2^1 = 2 * 2$, object with two properties in 2's power

For three dimension , $2^3 = 8 * 3$, object with three properties in 2's power

Properties according to dimensions ,

$R = \text{properties(Requirements)} = m - 1$

$R = \text{requirement}$

Therefore ,

So the final formula is ,

$n(O) = m * (2^{m-1}) * (n(n+1)/2)$

in one dimension , it is not possible to determine the requirement because , one dimension contain single length breadth , or width or time , and object always created in two dimension (or minimum two dimensions) .

therefore ,

$R(\text{requirement}) = m - 1$

So , according to requirement need with dimensions ,

$$D = 2^{m-1} = 2^R$$

3.2.Global Object : These are those type of unique entity which change their properties on the basis of other object that why these are divided into four main category.

Four object categories

1. Main object

2. Interface object

3. Impulsive object

4. Controlled object

3.3.Main object = These are those type of object which directly interact with system properties and work behind the main system interface with important decision .

For example = Director , CEO and Administrator of organization act as a Main object

3.4.Interface object = These are those type of object which directly interact with the system environment properties and work with system interface directly .

For example = Employees of organization act as an interface object

3.5.Impulsive object = These are those type of object which directly interact with the system environment properties and work partially behind and partially interface of the system .

For example = Managers of organization act as an impulsive object

3.6.Controlled object = These are those type of object which does not interact with the system environment properties and work for instant of time of period .

For example = Watchman and security guard act as a controlled object

3.7.Class =class may be defined as the , “ Abstract template which consists a group of characteristics and properties used by various objects on the basis of their requirement .

class defines “what is exists outside the system modules and explain the Internal and External characteristics of requirements ” # A class is essentially a type definition , which defines the state space of objects of its type and the operations (and semantics) that can be applied to objects of that type . Formation of classes is also a general technique used by humans for understanding systems and differentiating between classes.For example = Car is the best example because almost all cars contain similar properties like four tyres and engine type and modal etc.Objects In car categories like Ford , TATA , Maruti etc.An object also provides some services or operations. These services are the only means by which the state of object can be modified or viewed from outside . for operating a service , a message is sent to the object for that service .Normally , these services are defined for a class and are provided for each object of that class. Encapsulating services and attributes together in an object is one of the main features that distinguishes an Neuro object oriented programming approach and design from rotational modeling technique.

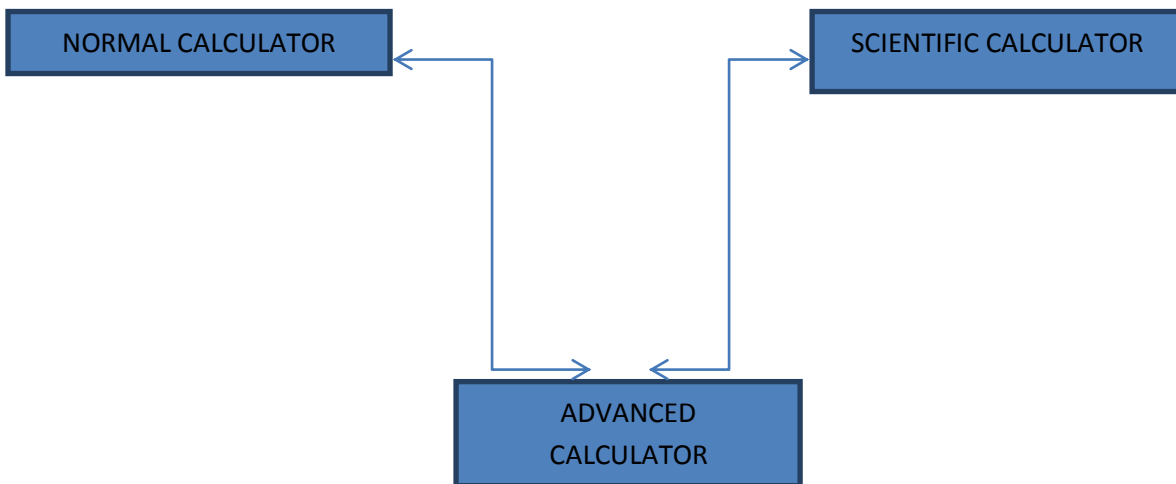
In Class division in Neuro object oriented programming approach and design

Class are divided into two category

- 1. Homologous class
- 2. Heterogeneous class

Homologous class =A group or series of n- number class that belong to the same system environment module are fall in this category and each containing the similar characteristics group forms Homologous classes .

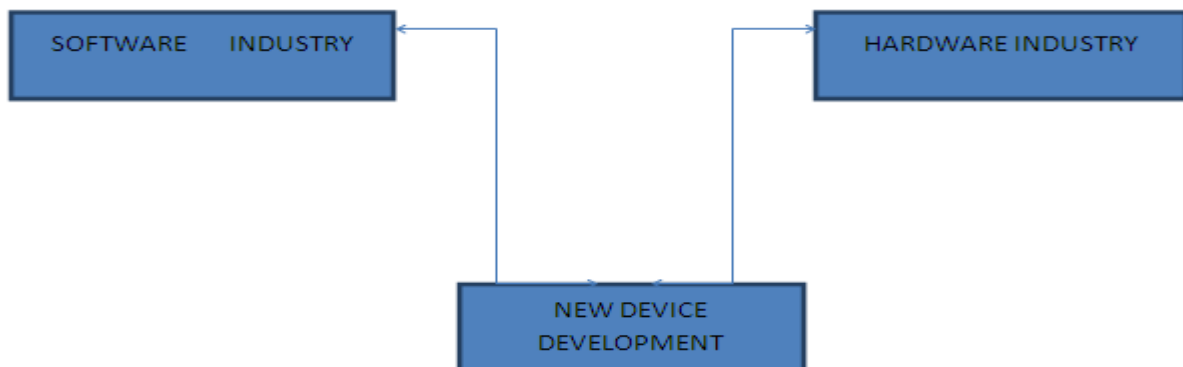
For example = calculator category



belong to the scientific calculator or normal calculator .

Heterogeneous class = A group or n – number class that belong to the different system environment module are fall in this category and each containing the dissimilar characteristics group forms Heterogeneous classes.

For example = combination of Hardware industry and software industry



Combination of software and hardware always help in development of new device.

REFERENCES

Help : Prof. Pallavee Joshi and Prof. Priti Khatri

Reference Paper : "Neuro Object Oriented Programming Approach and Design" / Author(s) Prateek Chaudhary / June 2013, Volume 3, and Issue 6

Reference book : Ivar Jacobson OOAD book.