Object-Oriented Systems approach to managing education in the 21st century

Author(s):

1. Rahul Vaid
(M.Engg. Mgmt. & M.S. program)
tinu@kosh.dartmouth.edu
Dartmouth College,
Hanover, New Hampshire, USA

2. Mukta Joshi,
(M.S. Program)
mukta@cis.wustl.edu
Washington University in St. Louis
St. Louis, Missouri,USA

1 Introduction:

The role of education is constantly changing and is becoming increasingly complex, because of which students have a new role to play in today's competitive world.

1.1 The Institution and Learning

As we approach the 21st century, strategies for applying the systems approach to learning are being proposed. These include re-engineering using team framework, forming what is popularly known today as virtual organizations,and total quality management of learning.

The institution or organization is structured according to the traditional top-down tree structure. We view it as a system and apply Object-Oriented Software (OOS) principles to convert it into a 'horizontal' or rather fashionably, a 'circular' institution. Learning is viewed as a process within this OOS that can be quantified to measure its efficacy according to specific organizational inputs.

1.2 Why Object Oriented?

We have come a long way from the structured, process-centric and data-centric software engineering design principles. We now visualize large and exceedingly complex systems as object smorgasbords. Object orientation is not only transforming the approach to software system representations but also giving a seasoned way to form organizations [4]. Object Oriented systems are built upon a sound engineering foundation, whose elements are collectively called the 'object model'. The object model encompasses the four basic principles of OOS, namely *Abstraction, Encapsulation, Inheritance, and Polymorphism*

Koshafian and Baker [6] define object orientation as :
$ObjectOrientation = AbstractDatatyping + Inheritance + ObjectIdentity$

The OOS methodology gives a rapid prototyping environment like the evolutionary model as opposed to the waterfall model in Computer Aided Software Engineering (CASE).

2 Organization and Process Development

As mentioned earlier, an organization may be an educational institution, the process may be the process of learning. The perfect organization can be visualized as a perpetual motion machine, running effortlessly, with perfect synchronization.

2.1 Measure of Quality

A good organization is one which has a structure that :

1. is flexible enough to be redesigned,
2. is portable enough to be modeled,
3. is scalable to the additions of sub modules and new processes,
4. lateral enough to provide a horizontal structure,
5. should have a bandwidth large enough so as not to sanitize and in effect filter information as it is delivered from one entity or unit to another.

We make an attempt in this paper to transform the orthodox organizational model to the ideal model (whose properties are stated above) using an OOS approach.

3 Transforming the structured system

To effectively utilize the OOS principles we need to fit the organization into the framework of OOS. In order to do this we define a so called transform. It is a virtual way of applying the concepts from OOS software domain to the organizational domain. A basic transform notation is shown in the fig 1.

$$\begin{tikzcd}\text{Software Object} \arrow[r, shift left, "\text{OOT}"] & \text{Organizational Principle} \arrow[l, shift left, "(\text{OOT})^{-1}"]\end{tikzcd}$$

Figure 1: Object Oriented Transform

3.1 Identifying the components of the Object Model:

The OOT of an *object* can be identified to be either the process of learning, a virtual team, or an organizational entity. Examples of possible OOTs of an object are shown in fig.2.

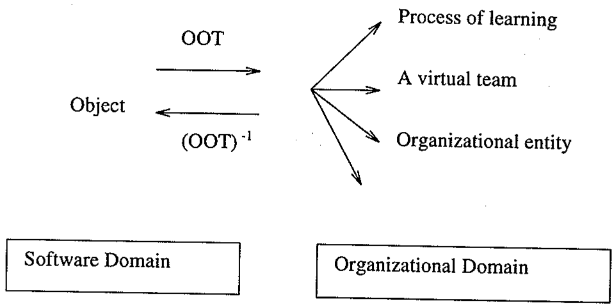


Figure 2: OOT of an object

3.2 Encapsulation and Abstraction:

Encapsulation is defined as the ability of an object to hide internal details that do not constitute to its essential characteristics. This implies that objects can establish boundaries around them.

Booch [2] defines abstraction as the process of focussing on the essential characteristics of an object distinguishing it from all other kinds of objects, thus providing crisply-defined conceptual boundaries relative to the user.

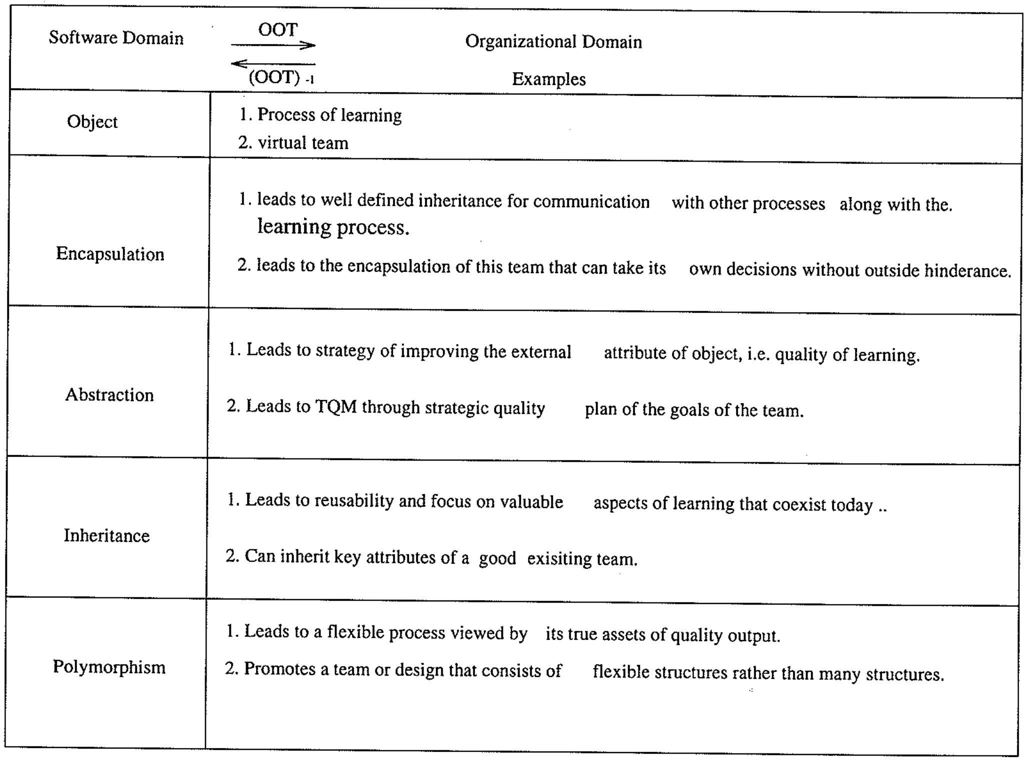
To this Christian [9] adds: 'Encapsulation and abstraction are opposite aspects of an objects treatment. Abstraction treats the part of an object that is visible externally, whereas encapsulation deals with the object internally, managing the private attributes of the object.

An analog of Abstraction and Encapsulation for the organizational domain by applying the OOT is shown in fig. 3. We see that a virtual team analogous to a group of mentors and students can be easily visualized into its object counterpart.

3.3 Inheritance and Polymorphism (Dependencies and hierarchies)

Meyer [1] defines inheritance as the relationship and dependency of one object on another. In a way inheritance addresses reusablity, reorganization and possibly scalability. Meyer defines polymorphism as the ability of an object to take several forms.

The figure shows examples of OOTs of Inheritance and Polymorphism. Once again using the OOT leads to possible results which form the basic attributes of an 'ideal' organization,



Software Domain $\xrightarrow{\text{OOT}}$ $\xleftarrow[\text{(OOT)}_{-1}]{}$	Organizational Domain Examples
Object	1. Process of learning 2. virtual team
Encapsulation	1. leads to well defined inheritance for communication with other processes along with the. learning process. 2. leads to the encapsulation of this team that can take its own decisions without outside hinderance.
Abstraction	1. Leads to strategy of improving the external attribute of object, i.e. quality of learning. 2. Leads to TQM through strategic quality plan of the goals of the team.
Inheritance	1. Leads to reusability and focus on valuable aspects of learning that coexist today .. 2. Can inherit key attributes of a good exisiting team.
Polymorphism	1. Leads to a flexible process viewed by its true assets of quality output. 2. Promotes a team or design that consists of flexible structures rather than many structures.

Figure 3: Encapsulation, Abstraction, Inheritance and Polymorphism

as discussed in section 3.

We have shown how the OOS methodologies can be used by institutions and organizations wherein the processes will lead to efficiency and success.

4 Semantic Data Models:

As a modelling technique for organizations and the porcesses within that are not currrently object oriented, we can use the Semantic data models to represent and organize the structure.

An example of a simple organization that is not initially object oriented is shown in the fig. Semantic data models for OO approach are used by Kashafiaut, Baker [4]. To start with, co-ordination and efficiency of the results through such organized and systematic change can be carried out using Petri Nets [5]. Petri Nets have been used to simulate processes for increasing the management's ability to understand and improve business processes [10] (Peters and Schultz, 1993)

In using this technique, an abstract level which is an intermediate level between the object and process can be captured as shown by Shams-Alice et.al [5].

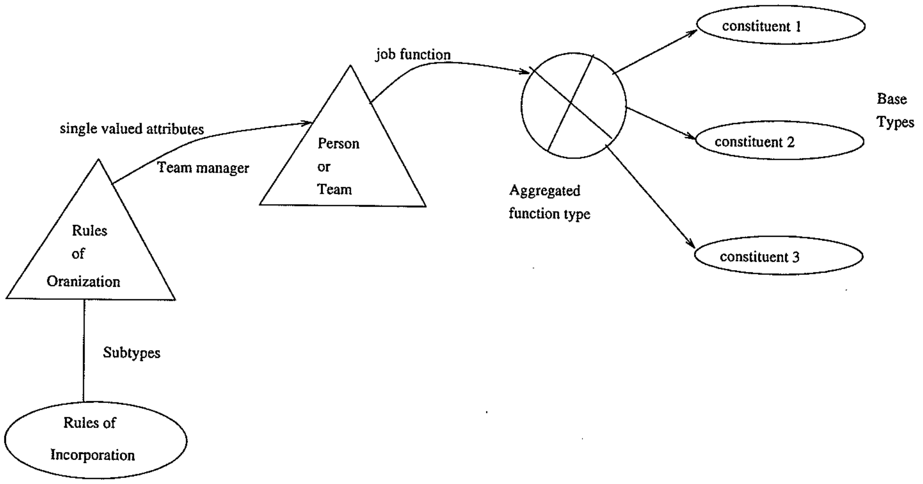


Figure 4: Schema of Semantic Data Model for Object Oriented modeling.

4.1 Conclusion:

We have tried to demonstrate that the object oriented approach to reorganizing an institution helps towards making the institution an ideal one. This paper basically shows how one can go about utilizing the OOS concepts in a business management context.

5 References:

[1] Meyer, B., *Object-Oriented Software Construction*, Prentice Hall, 1988.

[2] Booch,G., *Object Oriented Design with Applications*, The Benjamin/Cummings Publishing Company, 1991.

[3] Radia, S.R., Hamilton, G., Kessler, P. and Powell, M., *The Spung Object Model*, Proceedings of the VSENIX Conference on object-oriented Technologies (OOTs), June 26-29, 1995, Monterey, CA, USA. The VSENIX Association, 1995.

[4] Grayson, J.M., Runger, G.C., Ibrahim, N.A., *Object Oriented Oraganizations: Management Insights from software principles*, Quality Management Journal, vol. 3, No. 4, pp. 66-75, 1996.

[5] Shams-Alice, F. and Warboys, B.C., *Applying Object Oriented Modeling to support process Technology*, Department of Computer Science, University of Manchester, UK, 1996.

[6] Koshafiau and Baker, *Multimedia and Imaging databases*, Ch. 4, Object Orientation: Concepts, Analysis and design for Multimedia Databases, pp. 149-214, Prentice Hall, 1996.

[7] Basili, V.R., Briand, L.C., Melo, W.L., *A validation of object-oriented design metrics as quality indicators*, IEEE Transactions on Software Engineering, vol. 22, No. 10, October, 1996.

[8] Guest, R.H., *Organizational Change: The effect of Successful Leadership* , Invin, R.D., INC., 1962.

[9] Christian, K. *C++ programming*, Microsoft Press, Redmond, Washington, 1992.

[10] Peters, L. and Schultz, R.*Application of Petri-Nets in Object Oriented Enterprise Simulations.* The Proceedings of the 26th Hawaii International Conference on System Science, vol. 4, pp. 390-398, IEEE.