Journal of Information Systems Education

Spring 1994

Object-Oriented Design: A New Approach To Curriculum Development

ABSTRACT: Business schools are being challenged by industry to produce students who possess a broad understanding of the interlocking functions of business. However, the traditional curriculum design, especially at the introductory level, has emphasized learning business functions in isolated, stand-alone courses. This paper presents a new approach to designing introductory courses, utilizing a model called object-oriented design. This approach consists of breaking introductory courses in each discipline into discrete objects or modules. These objects are then re-coupled across disciplines to create a series of more holistic business courses. Object-oriented design is explained in more detail and then applied to re-design an introductory information systems course. A typical introductory course in information systems is used as an example to demonstrate how objects are created.

Dr. David Van Over
Dr. Dana L. Stover
College of Business and Economics
University of Idaho
Moscow, ID 83843

KEYWORDS: *Education, Curriculum Development, Object-oriented Design*

INTRODUCTION

There is a growing awareness in business colleges that the traditional approach to teaching students is becoming outmoded (1). Such awareness is heightened by comments commonly received from industry about the failure of graduates to see beyond the boundaries of their narrow specialties to the broader interlocking picture of business (2). Students' allegiance to a specific major also reveals that indeed, the larger picture of business is being neglected for concentration on the separate disciplines. These types of concerns are indicative of an approach to business education that is no longer appropriate. Some of the problems with the current approach are especially evident in introductory courses.

Most business colleges have some form of common body of knowledge (CBK) that students in all majors must fulfill. Typically, this consists of large section introductory classes in accounting, economics, finance, management, and so on. The intent of these courses is to give students an overview of the various disciplines that comprise business as well as a moderate amount of detail about each specific discipline. However, the effectiveness of this traditional approach is increasingly being questioned. Over ten years ago the Institute for Higher Education issued a report condemning the isolated manner in which most business courses were designed and instructed (3). This criticism was more recently voiced in a major report sponsored by the American Association of Collegiate Schools of Business (AACSB):

> ...the typical business school curriculum-particularly at the undergraduate level (does not) provide sufficient attention to both the need to and the means to use specialized functional knowledge in an integrated approach to the increasingly complex, fast-changing, and multidimensional problems of contemporary business (4:65).

The lack of cross-functional integration is especially troublesome at the undergraduate level due to their relative lack of job experience. Work experience often helps students to make the necessary integrations and cross-discipline linkages themselves -- it provides a useful framework upon which to incorporate new functional information. To the degree students lack experience, the greater the need for instructors to provide such a framework within the course (2).

This paper describes some concerns with the traditional approach to business education and offers a new approach for teaching the CBK using the concepts of object-oriented design (OOD). OOD is based on objects that encapsulate all of the information and operations required to perform a specific task (5) . It offers a different approach to curriculum design by coupling and de-coupling objects to create specific courses.

First, a description of the current state of teaching introductory courses in a business college is presented. OOD is then presented in more detail followed by a description of how this approach can be used in course design. The paper concludes with an illustration of how an introductory IS course can be taught as modules in other introductory classes.

PROBLEMS WITH CURRENT INTRODUCTORY COURSE STRUCTURE

A primary concern with the traditional CBK format is the position given to introductory courses in the overall educational plan. Instructors either approach the course as the initial step in a sequence of specific major courses or as a service course to non-majors. In the former case, instructors often cover detailed topics beyond the needs of non-majors in the class. When approached as a service course, instructors are often tempted to view the course as their one chance to tell students absolutely everything about the discipline. In either case, the

number of topics and assigned chapters multiplies past what many students can grasp beyond a superficial level of understanding. Certainly, most attempts at providing a framework to integrate topics among various introductory courses are sacrificed in order to just keep up with the topic-of-the-day. Compounding this problem are the quality variances among both teachers and students which create gaps and redundancies in learning. Despite these inconsistencies, material of future courses are based on uniform coverage and retention of introductory material.

An additional problem with the current CBK approach is the large amounts of resources such courses consume, not only in terms of faculty hours, but also in student credit-hours. Currently, most colleges require at least 6 introductory courses. However, an impending regulation in the AACSB guidelines will require a 10% reduction in course work within the business college. While some upper division requirements may be lessened, at least part of the CBK obligation must be eliminated.

However, the more severe problem with the CBK approach stems from the stand-alone nature of the introductory courses. Students seldom see an integrated picture of business. Instructors tend to focus on the details of their own discipline and assume students can make the needed connections to other disciplines. The senior-level capstone strategy class is designed to bring all the functional pieces together, but by that time much learning has been lost. Without the larger framework of how the pieces fit together, it is often hard for students to retain understanding about the functional pieces themselves. This is especially true for undergraduate students who have little or no previous work experience.

The problem is not the result of unconcerned faculty, but rather an artifact of the traditional CBK approach. In this approach, instructors are expected to teach the basic concepts of an area and integrate those concepts across majors without sufficient resources. Faculty typically do not have the time nor necessarily the expertise to understand current trends across disciplines. Secondly, without a lock-step program in which all students take the same courses in the same order, an instructor cannot make assumptions about the knowledge of any student. When an instructor, in an attempt at cross-disciplinary integration, uses an example from an area which is unfamiliar to a student, not only is there no integration of the concept, there is probably more confusion. The instructor must choose between integration for some and confusion for other students, or no integration at all.

What is needed is not just a refinement in how introductory courses are taught, but a shift in the approach of CBK instruction. That is, instructors must begin teaching these courses as an introduction to business, not as standardized introductions to the individual disciplines. Perhaps if faculty do not label and segregate the various disciplines, students will approach courses more holistically. The remainder of this paper describes a new approach for teaching the CBK utilizing the concepts of object-oriented design (OOD)

OBJECT-ORIENTED DESIGN

The OOD approach is a way of thinking which decomposes a system into objects. An object consists of the data and all necessary operations that manipulate it. Each object is responsible for its operations and data and each has a specific role in the system (6) . The encapsulated data and procedures are separate from other objects in a system. The value of OOD is that objects remain independent and hence uncontaminated. Therefore, an object can be part of more than one system. OOD is founded on three basic concepts: encapsulation, messaging, and inheritance, which when combined lead to the creation of reusable objects.

Encapsulation organizes data and corresponding processes which manipulate that data into a single entity, called an object (7) The data and operations contained in an object are conceptually related to each other and distinct from all other objects in a system. Encapsulation adds two important features to a system. First, it affects system construction. Since objects are self-contained, designers need only define what they want an object to do, not how to do it. System design becomes a process of selecting objects and executing them in a logical order. Second, it affects system maintenance. The independence of the objects insulates changes in one object from affecting other objects (8) . Therefore, corrections and improvements can be confined to a single object without reconstructing an entire system.

Messaging is a request for an object to perform a procedure. A message consists of the name of an operation and any required arguments (9) . The message does not request how an operation should be performed. Messages and objects are independent. That is, the same message may be sent to more than one object, each performing its specific tasks, or an object may respond to more than one type of message (10). Objects can be chained together in that a message to one object may require that object to send a message to a third object, and it to a fourth object, and so on. Situations in which more than one object is required to complete a task are called collaborations. Changing the functionality of a system can therefore be accomplished by changing objects or by changing the messaging structure.

The third characteristic of OOD is inheritance. Inheritance allows objects to be arranged in hierarchies from the general to the specific with lower objects inheriting the attributes and behaviors from objects above. Objects are organized into classes so that each object in a class shares some conceptual similarity. Classes are then organized hierarchically where a class contains all of the features of a more general class plus some unique features. The value of inheritance is the elimination of redundancy or "re-inventing the wheel". When a new process is required, the designer determines the correct class for the object. Once the class is found, the designer will either find that the object already exists, or that a similar object exists which requires only minor modifications.

The primary benefit of OOD is reusability. Because each object is a self-contained, autonomous entity, they are reusable. Objects created with a high degree of self-sufficiency can be used as components in many kinds of systems. One goal of OOD is to create a library of reusable objects that have clearly defined behaviors and are fully documented.

OOD APPROACH TO COURSE DEVELOPMENT

Applying OOD to course development focuses on the concepts taught in a course and the procedures employed to teach them. Many OOD concepts are already used, at least in part, in course design. Encapsulation applied to traditional course development has all instructions, information, and exercises required to learn concepts kept together in an object (course) and jointly they are separated from other objects. Individual objects are bundled to create a major or a degree plan. The objects are somewhat autonomous in that people not directly involved with an object are not

typically concerned with how the concepts are taught, only that upon completion students have command of the concepts.

The concept of messaging is very simple in academics. Students enroll in a course with the intent that they will learn a set of concepts. The operation of the message is to teach and the argument is the student enrolled. Collaboration occurs when students must take more than one course to fulfill a conceptual requirement (e.g., macro and micro-economics).

The idea of inheritance is also part of course development. Conceptual similarity is maintained by identifying courses within areas and areas within colleges. A hierarchy is created by offering classes at different levels from freshman introductory courses to doctoral seminars. Inheritance is enforced with prerequisites. Most classes require an understanding of the concepts taught in other classes. That is, courses inherit the concepts of previously taken courses. For example, statistics inherits many concepts from calculus.

Because OOD concepts are only partially implemented in the current approach to teaching the CBK, their primary advantage, reusability, is seldom recognized. Reusability is difficult to attain in semester length objects because of their size. One restriction is a limited audience. Some objects presented in a course are valuable to students outside the college. However, students may not be willing to make a semester long commitment for a few objects or may lack the pre-requisites for objects in which they have no interest. A second example of non-reusability is the lack of transferability of the preparation of teaching a course. Despite the stated goal of standardization of learning among sections of a course, sections may be "custom" prepared. A change to any component of a section (e.g., the teacher or the exercises) requires a substantial amount of re-work. Furthermore, custom approaches to teaching a course may lead to gaps in coverage. The biggest problem with the loss of reusability is redundancy.

Redundancy is evident in the catalogs of courses. Colleges of business typically offer their own versions of statistics, communication, programming, and mathematics to name but a few. A new course is created because a similar course in another college is only a 70, 80, or 90% fit. In essence, this creates 90% redundancy to get 10% specialized coverage. The same phenomenon occurs at the department level with introductory courses.

Nearly all introductory courses in business colleges have a chapter or section which covers management, marketing, decision making, computer use, and/or business strategy. The arguments for the redundancy are that students need to hear some concepts more than once to fully understand them or that hearing the same concepts from different points of view further entrench the learning. While this view may have some merit, it is difficult to argue that unplanned and uncontrolled repetition in a curriculum provides value to students.

One solution to these problems is to more fully implement OOD by de-coupling the objects of a semester course. The problem with the current approach is that objects are simply too large. Nearly all academic objects are 48 hours in length (16 weeks, 3 hours per week). A semester course is actually several objects packaged together, but treated as an individual object. While a standardized package size makes scheduling easier, the benefits of tightly encapsulated objects are far greater.

The advantage of incorporating OOD in course design is to create objects that are tightly encapsulated and hence very reusable. Teaching reusable objects instead of semester long courses would eliminate a great deal of redundancy. Students could learn the same concepts with fewer courses and faculty could teach the same concepts with fewer class preparations. Scheduling could actually become simpler since changing the teacher would have little effect on an object. Each object could have a standard college-wide preparation and the instructor would become one of the interchangeable components. More importantly, the concepts would be better ingrained since they would be presented within the framework of business rather than isolated in a discipline. Coupling various objects together to create a course would greatly enhance cross-discipline integration. Additionally, instructors can become much more effective at evaluating student comprehension of each object.

The obvious difficulty in this approach is packaging the objects. One extreme is to offer each object as an individual course. Students would select forty or so objects varying in length from one to two weeks instead of eight to ten 16 week courses. While this may be the ultimate solution, implementation of such a system might be a bit overwhelming. A more moderate approach identifies the objects of value to all business majors in some traditional courses and places those objects within other courses. An example is the introduction to information systems (IS) course.

IS is a service course that is valuable to students of every major. Yet many IS introductory courses are taught as the initial course to IS majors, not necessarily as a useful tool for other students - two very different objectives. The goal of an introductory course should be to teach students the tools they will need, irrespective of their major. Therefore, IS objects should be taught within the environment of a particular subject matter (e.g., management, marketing, etc.). The next section of this paper shows how an introductory IS course can be broken down into objects and then those objects relevant to a management class can be taught as part of that class.

INTRODUCTORY IS COURSE OBJECTS

The first step in designing an OOD course is to identify object classes (11) . A typical set of objects for an introductory IS course is presented in Table 1. Table 1 shows the classes, objects, example topics within objects, and time required to teach the objects that are common to many introductory IS courses. One interesting note to Table 1 is that by eliminating some redundancy of topics covered in other courses, a 48 hour course is reduced to 33 hours. The next step is to assign responsibilities to the classes and objects.

Responsibilities are the knowledge and actions contained in a class or object. In course development, they are what the students are supposed to learn and what instructors are supposed to teach. Responsibilities are similar to course descriptions. For example, Table 2 shows the responsibilities for a telecommunications class and objects. Either the entire class or the individual objects can be used as a vehicle to present the concepts.

Collaboration is the linking of objects or classes to perform more complex tasks. Collaboration represents the use of pre- and co-requisites in course development. The telecommunications example presented in Table 2 shows that the data network has two collaborations, computer technology and information systems. Since a class has all the collaborations of each of its objects, the class also has two collaborations. Students can take either the class as a whole or the data network object after taking the

other two classes, or they can take the video network or voice network modules without pre-or co-requisites. That is, they are stand-alone modules.

The final step is to place the objects within other classes. Teachers of other courses could adopt some of the objects to fit into their courses. For example, most introductory management courses include a chapter on information systems. This chapter typically covers such topics as why managers need information, where they get information, typical business software, and common applications of computers in business. Rather than simply reading the chapter and attending a lecture by the management instructor, students instead would sign up for specific IS modules taught by IS faculty. Management faculty would select appropriate modules based on what they believe all management students should learn about IS. The more pertinent modules would cover computer types, how computers work, transaction systems, management systems, decision support systems, office automation, data as a resource, IS planning, application packages, and IS efficiency. Approximately 7 classroom hours would be required to present these modules.

Three questions are likely to arise when applying OOD to course development. The first is the concern from teachers in introductory courses about adding objects to already full syllabi. The answer is that while students may lose three or four weeks of topic coverage in the introductory course, they are gaining an additional semester course. Topics which are important to students majoring in the area, but not necessarily to others, could be moved from the introductory course into another course offered by the discipline. Since the students will have extra time for a course, they may choose to take an additional course in the area.

The second concern is what to do with objects that are not adopted by other classes. There are two potential solutions to this question depending on why they were not selected. If they were not chosen because other majors are not interested in them, then they are probably specific to the IS major and should be moved to another course in the major. If however, they are valuable to students in other majors but do not fit into another course, then one of the other introductory courses should be restructured to accommodate them.

The final concern is in choosing which classes to apply OOD. The answer is it does not matter which CBK courses are selected, they are all good candidates. The short term solution is to pick those courses which provide the basic concepts or tools that are used in other courses. Good examples are IS, management, or marketing. The long term solution is to apply OOD to all introductory courses. Then, rather than adopting the objects in other courses, new generalized courses could be created from classes or super classes of objects. The result would be a generalized introduction to business approach as opposed to detailed introduction to business disciplines.

Table 1. INTRODUCTORY MIS COURSE OBJECTS

CLASSES	OBJECTS	EXAMPLES	DURATION (Hours)
Computer Technology	Types of computers	PC, Mini, Mainframe, super	1
	How computers work	Architecture	1
	Peripheral devices	Data input, storage, output	1
Information Systems	Transaction	Control	0.5
	Management	Reporting	0.5
	Decision support	Modeling	0.5
	Expert systems	Knowledge based	0.5
	Executive	Graphics	0.5
	Office automation	Multi-media	0.5
Databases/DBMS	Data as a resource	Data, information	1
	Organizing data	Logical vs. physical, relational, network, hierarchical	3
	Accessing data	sequential, direct	2
	Maintaining data	back-up, recovery	1
Software Development	IS planning	Business needs,	1
		Evaluating IS's	1
	System life cycle	Initiation, Development, Implementation, Operation & maintenance	3
	Computer aided Software Engineering (CASE)	DFD, ERD, Data dictionary, code generators	2
	Prototypes	Application packages	1
	End-user computing	4 GL's	1
	Programming	Languages, techniques	2
System Implementation	IS efficiency	Changing work processes, IS/Organization fit	1
	Security	Accidents,	1
		Computer crime	1
Telecommunications	Data Networks	Local area, Wide area, Value added	3
	Voice networks	Lease vs. buy	2
	Video Networks	Teleconferencing, Telecommuting	1

Objects and classes deleted because they are better taught in other domains.
Ethics, Competitive opportunities, Decision making, Management

CONCLUSION

Business schools must begin re-thinking how and what students are learning about business. Certainly, the basic functions of business must continue to be taught, but in a context that produces an understanding of the interdependencies of these functions. This paper has offered OOD as one approach to redesigning introductory curriculum which couples various objects across disciplines to create courses which approach business as an interlocking network of different disciplines rather than the traditional stand-alone functional courses.

While some faculty resistance is anticipated with this approach, academia must realize it is not exempt from a changing environment and customer demands. Over the past few decades schools of business have become very good at teaching disciplines in isolation but often have lost sight of the end product. Each major maintains a

Table 2. TELECOMMUNICATIONS

Responsibilities
Class
Topic coverage includes:
Basic technology of information transfer on both bounded and unbounded media.
Common configurations of data, voice, & video networks.
Lease versus buy decisions on switching equipment.
Common applications in business.
Effects of usage on organization structure and business practices.

Objects
Data networks topic coverage includes:
Basic technology of information transfer on both bounded and unbounded media.
Common configurations of data networks.
Transmission techniques in local area networks.
Uses and management of local area, wide area, and value added networks.
Security and responsibility for information in a distributed environment.

Voice networks topic coverage includes:
Basic technology of telephone systems.
Lease versus buy decisions on switching equipment.
Common applications; PBX, FAX, Cellular.
Effects of usage on organization structure and business practices.

Video networks topic coverage includes:
Basic technology of telephone systems.
Common applications; telecommuting, teleconferences.
Effects of usage on organization structure and business practices.

Collaborations
Class
Computer Technology
Information Systems

Objects
Data Networks
Computer Technology
Information Systems
Voice Networks
Video Networks

high degree of conformance, but the market quality of the end product has declined. By reorganizing our approach around the end product, a business education, we can help regain the market quality. Students can become valuable parts of businesses rather than possessors of isolated skill sets.

REFERENCES

1. Byrne, J. (1993). Harvard B-school: An American institution in need of reform. *Business Week*, July 19, 58-65.

2. *Harvard Business Review* (1992). MBA: Is the traditional model doomed? Nov-Dec, 128-140.

3. *Chronicle of Higher Education* (1984). Text of New Report on Excellence in Undergraduate Education, Oct. 24, 35-51.

4. Porter, L.W., & McKibbin, L.E. (1988). *Management Education and Development: Drift or Thrust into the 21st Century?*. New York: McGraw-Hill.

5. Howard, G.S. (1988). Object oriented programming explained. *Journal of Systems Management*, July, 13-19.

6. Ten Dyke, R.P., & Kunz, J.C. (1989). Object-oriented programming. *IBM Systems Journal, 28*(3), 465-478.

7. Freedman, D. (1992). The objective approach. *CIO*, May 15, 70-72.

8. Wirfs-Brock, R., Wilkerson, B., & Wiener, L. (1990). *Designing Object-Oriented Software*, Englewood Cliffs, N.J.: Prentice Hall.

9. Hartog, C., Mosely, D., & Haspiel, J. (1990). *Object-Oriented Technology: A Report to the Board.* Working Paper at Center for the Study of Data Processing, Washington University, St. Louis, 3(5), (WP 88-25).

10. Rubin, K.S., & Goldberg, A. (1992). Object behavior analysis. *Communications of the ACM, 35* (9), 48-62.

11. Monarchi, D.E., & Puhr, G.I. (1992). A research typology for object-oriented analysis and design. *Communications of the ACM, 35*(9), 35-47.

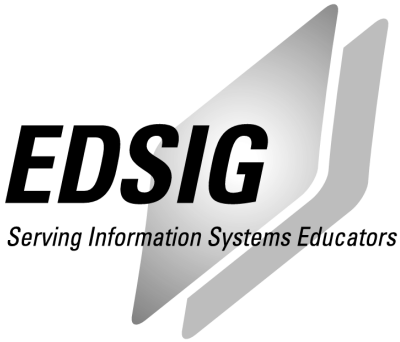
AUTHORS' BIOGRAPHIES

David Van Over received his doctorate from the University of Houston. He is an assistant professor and area coordinator of Information Systems at the University of Idaho. His current research interests include computer integrated manufacturing, curriculum design, and real time simulation.

Dana Stover received her doctorate from Washington State University. She is currently an assistant professor of Management. Her current research focus is intra-organizational segregation, sexual harassment, and curriculum development.

Professors Van Over and Stover are currently involved in a project to evaluate and restructure the curriculum for College of Business students.

Information Systems & Computing
Academic Professionals



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096