8-15-1997

Object-Oriented Trends:Information Systems Degree Programs

David E. Douglas
University of Arkansas

Bill C. Hardgrave
University of Arkansas

Object-Oriented Trends:
Information Systems Degree Programs

David E. Douglas Bill C. Hardgrave
Computer Information Systems and Quantitative Analysis
University of Arkansas

Abstract

Object-oriented (OO) application development continues to steadily move into the mainstream of systems development. Many information technologists forecast OO development will become the dominant application development strategy within a few years. Currently, object-oriented development tools and methodologies seem to be maturing enough to support this forecast. Key to supporting this shift toward OO development are a sufficient number of OO-based information systems professionals. Industry ultimately must seek their OO talent pool from information systems (IS) and computer science (CS) degree programs. This study looks at the change, over a four-year period (1992-1996), of OO technologies taught in IS programs. Specifically, changes in demographics of IS programs teaching OO, extent of OO courses taught, OO programming languages, and OO methodologies are presented.

Introduction

The fact that almost every major software vendor claims that their development methodology, tools, and databases are object-oriented confirms the impact OO has, and continues to have, on application development. Although OO may not be the only answer for eliminating poor quality, over-budget and late software development projects, many see OO as the best solution for quality software development (e.g., Martin and Odell, 1992).

The successful move of commercial application development utilizing OO technology into the mainstream requires OO developers. Organizations have relied on academic programs, particularly information systems (IS) degree programs, for infusion of new talent as commercial application developers. The question is: are IS degree programs providing enough OO course work to prepare graduates to assume, over time, roles as OO developers and support the increasing trend of OO development?

This study compares the extent of OO in IS curricula during two time frames, Spring 1992 and Spring 1996. Changes in the number of courses in which OO is taught and the content changes of those course are explored. Specifically, content changes in OO languages and methodologies are investigated. The magnitude of the changes over this time should provide guidance for both academia and industry as to the commitment of IS degree programs to provide a significant share of the talent pool to support the shift to OO development.

Methods and Results

Respondents

Questionnaires were sent to IS departments throughout the United States and Canada in Spring 1992 and again in Spring 1996. In 1992, 92 usable questionnaires were returned and in 1996, 101 usable questionnaires were returned. The sample for each of the two mailings represents both private and public and large and small colleges with IS degree programs of varying sizes. Table 1 provides the school profiles for 1992 and 1996 respondents. The majority of the respondents are from institutions which are public, AACSB accredited with less than 2000 students in the College of Business Administration and less than 200 majors.

In 1992, 35 of the 92 respondent schools (38%) were currently teaching OO, leaving 57 (62%) who were not teaching OO. Of the 57 not teaching OO, 30 planned to introduce OO in their course work in the near future. From the results of the 1996 survey, 72 of the respondent schools (71%) are currently teaching OO.

Of the 29 schools that are not teaching OO, 17 plan to teach OO, leaving only 12 schools (12%) that do not currently teach OO nor plan to teach OO. In the 1992 survey, the percentage of schools that were teaching OO plus those planning to teach OO was 71% [(35+30) / 92]. Remarkably, this percentage coincides with the percentage of schools currently teaching OO as indicated on the 1996 survey. The majority of IS degree programs are now teaching or plan to teach OO concepts (88%).

Programming Languages

Table 2 summarizes the OO programming languages (OOPLs) identified from the two surveys as being taught in IS degree programs. In both surveys, the language taught most often is C++. C++ also represents the largest increase in OOPLs being taught in IS degree programs from 1992 to 1996. There were small increases in the teaching of other OOPLs, but none represent changes large enough to identify a trend. Particularly, no notable increases occurred in the number of schools teaching the "pure" OO languages - Actor, Eiffel, Simula, Smalltalk. The "hybrid" language, C++, continues to be the overwhelming choice for teaching OO programming. The number of schools teaching Visual BASIC, an object-based language, was lower than expected given its market penetration in the client/server development arena.

The coverage of OO in IS curricula is also revealed in the number of programming languages taught. In 1992, most of the programs teaching OO relied on a single language (i.e., only 6 programs were teaching more than one language). The 1996 data reveals a large increase in the number of schools teaching more than one language in their curriculum (28 schools).

Development Methods

The top development methodologies identified in the two surveys are shown in Table 3. According to the 1996 survey, Booch, Coad & Yourdon, and OMT are the most popular methodologies.

Interestingly, several non-OO methodologies - functional decomposition, enterprise modeling and waterfall - appear to be widely used to guide OO development, although the use of these methodologies has decreased in the past four years. The trend away from traditional (non-OO) methodologies towards OO methodologies provides some interesting observations. First, in 1992, OO methodologies were immature, thus forcing faculty to rely on the integration of OO techniques with existing non-OO methodologies. By 1996, OO methodologies matured enough to begin the move away from traditional development methodologies. Second, the use of non-OO methodologies could also indicate instructors' low level of comfort with using OO. As faculty continue to use OO, they are naturally becoming comfortable enough with OO development methodologies to move away from teaching the non-OO methodologies.

Of the OO methodologies, Coad & Yourdon and OMT realized the most change between 1992 and 1996. Coad & Yourdon decreased in the number of schools teaching it, while OMT increased. Booch's methodology demonstrated strong use in both years. Recently, three of the most common OO methodologies used in industry (Jones, 1995) - Booch, OMT, and Jacobson - have combined to define the new methodology called the Unified Modeling Language. Combining the top OO methodologies to share the same development approach and notation will result in tool vendors, such as CASE vendors, implementing the Unified Modeling Language. This will probably lead to the Unified Modeling Language becoming the dominate OO methodology being used in industry and consequently taught in IS degree programs.

Course Profiles

Both surveys collected information about the courses in which OO is offered; specifically, whether OO topics were taught in designated OO courses or integrated into existing courses. OO topics are categorized as: OO analysis (OOA), OO design (OOD), OO programming (OOP), and OO database management systems (OODBMS). For discussion purposes, courses which devote more than 50% of a course's time to

the above OO topics are considered designated courses. Table 4 shows the comparison of the designated versus integrated course offerings from the two surveys.

IS programs are electing to teach OO topics using the integrated approach more than the designated approach. Although the number of integrated and designated courses increased from 1992 to 1996, the ratio of integrated-to-designated did not change substantially. There are at least two reasons to explain this trend. First, it is naturally easier to incrementally introduce a new topic, such as OO, into an existing area. For example, OO data modeling can easily be introduced in a database management course. Second, the bureaucratic difficulty in introducing new courses into a curriculum may be forcing many programs to introduce OO into existing courses.

Conclusion

This article has investigated IS degree programs' current and planned teaching of OO concepts at two different points in time - 1992 and 1996. Based upon the trends identified herein, it appears that IS degree programs have recognized and responded to the need to be a supplier of OO developers for industry.

References

Corporate IS considers object plans. *Computerworld* 28, 4 (Jan 24, 1994), 61, 64.

Jones, N. Results of an interactive survey on object orientation. Gartner Group, Research Note: T-700-1192, July 12, 1995.

Martin, J., and Odell, J. *Object-Oriented Analysis and Design*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

Massey, P.D, and Douglas, D.E., "Object-Oriented Technologies in Information Systems Curricula," *Journal of Computer Information Systems*, 34 (1), Fall 1993, 3-7.

Radding, A. Smalltalk, C++ knock heads. *Computerworld* 29, 45 (Nov 6, 1995), 54.

Ruber, P., "Object Relief," *InfoWorld*, 19 (4), January 27, 1997, 85-86.

Institution characteristics	**Survey Year**	
	1992	**1996**
Public	71.7%	61.4%
AACSB Accredited	65.4%	61.1%
Number of Students < 1000	29.1%	29.7%
1000 - 1999	29.1%	34.1%
2000 - 2999	16.3%	15.4%
> 3000	25.6%	20.9%
Number of Majors < 100	50.0%	47.3%
100 - 199	28.4%	31.9%
200 - 299	13.5%	12.1%

> 300	8.1%	8.8%
Teaching OO	38%	71%
Planning to teach OO	33%	17%

Table 1: Demographics

	1992 (n = 92)	1996 (n = 101)
	Number	Number
Actor	2	0
Ada	2	7
C	4	7
C++	18	48
CLOS	0	0
Eiffel	0	2
Lisp	1	2
Object Pascal	3	5
Objective-C	0	1
Simula	0	0
Smalltalk	9	9
VisualAge	1	2
Visual Basic	2	8
>1 language	6	28

Table 2: OO Languages Taught

	1992 (n = 92)	1996 (n = 101)
	Number	Number
Booch	16	21
Coad & Yourdon	22	16
Enterprise Modeling*	11	8
Func. Decomposition*	14	8
Jacobson	0	3
OMT	2	10
Waterfall*	6	2

*non-OO methodologies

Table 3: OO Development Methodologies

	1992	1996

	Current	Planned	Current	Planned
Designated	33	23	64	27
Integrated	56	58	84	32
Total	89	81	148	59

Table 4: Designated versus Integrated Courses