THE COBOL CRISIS POST Y2K – HOW DOES ACADEME' RESPOND?

Robert F. Roggio, Department of Computer and Information Sciences,
University of North Florida, 4567 St. Johns Bluff Road South, Jacksonville, FL 32224;
Ph: (904) 620-2985. email: broggio@unf.edu
James R. Comer, Department of Computer Science, Texas Christian University,
PO Box 298850, Fort Worth, TX 36129,Ph: (817) 257-7166, Email: j.comer@tcu.edu

ABSTRACT

Many IT project managers are seeking IS/CIS graduates / professionals with significant COBOL skills to support billions of lines of legacy applications worldwide. Yet in academe', programs are migrating away from COBOL. This trend may well produce graduates unable to maintain and enhance existing COBOL applications notwithstanding projected new development using COBOL. After examining industry's current and projected dependence on COBOL, concerns in COBOL instruction and its revitalization are documented. Recommendations for two IS/CIS programs - one oriented toward AACSB criteria; a second - outside a School of Business are presented but not included in this paper.

THE DOMINANCE OF COBOL IN THE CORPORATE IT WORLD

Now that the Y2K transition has concluded, the business community faces a reassessment (and perhaps retrenchment) of its COBOL dependency. While the corporate world may still be patting themselves on the back in justifiable pride for a successful transition of billions of lines of operational COBOL code into Y2K compliance, business IT managers and faculty curriculum planners must carefully reconcile the role that COBOL will play for the next ten to fifteen years in their respective worlds. Michael Wheatley [1] cites a Gartner Group estimate that somewhere between 150 and 175 billion lines of COBOL are currently in production worldwide and that this base is still growing. Stern and Stern [2], authors of one of the most widely used COBOL textbooks estimate the there are between 150 and 250 billion lines of production COBOL code. They further assert that additional COBOL code is being produced at a rate of around five billion more lines each year!

Let us consider a few more facts: the COBOL community invested somewhere between $300 billion and $600 billion dollars in the Y2K upgrading exercises. [3] These vast expenditures clearly revealed not only the worldwide dependence on COBOL applications for current operational considerations, but equally important (perhaps more important) indicated corporate IT needs in the foreseeable future. Wheatley [1] cites that the Y2K experiences not only brought into sharp focus that COBOL assets were much more complex than previously imagined, but that corporate dependency upon these assets is also much higher than previously conjectured. Bradley [4] offers that 75% of all production transaction on mainframes are done using COBOL programming support, over 60% of all web-access data resides on mainframes, that COBOL mainframes process more than 83% of all transactions worldwide, and that over 95% of finance-insurance data is processed with COBOL. Scott Ankrum [5] wrote that CICS transaction volume grew from 20 billion a day in 1998 to 30 billion a day in 2001. He further cites that this 50% growth has largely been driven by web-based applications that access back-end COBOL applications. "That philosophy of [COBOL] extinction has been replaced with one of extension and inclusion. COBOL applications are, by and large, too critical and too valuable to consider replacing en masse." [3] If there is but one thing alone that Y2K clearly pointed out, it is simply that COBOL is alive, well, and will persist well into the new millennium.

WHAT DO BUSINESS LEADERS PREDICT FOR COBOL'S FUTURE?

Carr and Kizior [6] undertook considerable research that provide very illuminating statistics regarding business and academic leader feelings regarding the role of COBOL now and in the future. In fact, they surveyed almost 3,000 CIS and IS programs and 5,000 businesses regarding the perceived future of COBOL. "In surveying business and academic leaders, the authors found that almost 95% of academic respondents and 90% of IT managers still want IS curricula to offer COBOL instruction. They further discovered that nearly 90% of IT managers want both object-oriented and web-based features integrated into COBOL instruction in college curricula." [6]

Another set of related questions can be collapsed into a single question: "Will COBOL will continue as a major implementation language for business solutions?" To discover the answer to these and related questions, Carr and Kizior's survey of business leaders discovered that "over 87% currently develop and maintain code written in COBOL – over 50% of effort solely in maintaining current legacy COBOL applications; 20% of effort developing new applications using COBOL; and 30% of the programming resources expended in a mixture of new application development and maintenance." [6] That maintenance is a large percentage of effort expended on legacy applications is not a great surprise. However, that 20% of efforts in COBOL applications are currently expended developing new applications is most eye-opening and should provide additional insight as to what skill sets IT managers will need in the future from IS graduates. Ulrich reports a Gartner comment "...that 15% of all new application functionality through 2005 will be in Cobol." [7]

When asked how long COBOL was expected to remain important in meeting functional business needs, of IT managers surveyed, almost half cited that COBOL would be used at the current level for about the next 10 years. While around one-third of those surveyed indicated that they expected to see COBOL's importance in applications development decline over the years, there was still about 5% who indicated that they actually expected the use of COBOL to actually increase over the same next 10 years.

ACADEMIC PROBLEMS AND TRENDS FOR COBOL INSTRUCTION

Let's first look at the numbers. The percentage of IS/CIS academic programs teaching COBOL is steadily decreasing, while instruction in languages such as Java and Visual Basic is increasing. According to Hardgrave and Doke [8], COBOL was offered in 90% of undergraduate IS programs in 1995 but has fallen to 53% in 1998, while instruction in C++ has fallen from 60% in 1995 to 53% in 1998. OO COBOL, very slow to get attention, has risen from 0% in 1995 to 7% in 1998; Java instruction has skyrocketed from 0% in 1995 to 43% in 1998 and instruction in Visual Basic has risen from 59% in 1995 to 70% in 1998.

There are many reasons for these figures. Faculty (as well as students) like to teach and learn the latest technologies. While COBOL 85 has been a significant improvement over earlier standards, COBOL itself is not a glamorous language and is often considered cumbersome. Some feel it lacks "elegance." The current standard fails to include features commonly available in more modern languages that can be readily used to satisfy current and future industry demands for multi-tiered applications.

GUI interfaces can be readily taught in Visual Basic and Java; networking considerations are facilitated nicely in Java; object-oriented programming with topics such as reusability, component-based software development and so much more, are provided in Visual Basic to a degree, more in C++, and in Java extensively. Many features forecast in the new COBOL standard are readily available in C++ and Java now. Thus there is little surprise that both faculty and students gravitate toward these technologies oftentimes at the direct expense of COBOL.

Attempts at IS/CIS program modernization present very serious challenges to educators. Often demand for courses such as Visual Basic, Internet programming, Java, e-commerce, client-server computing, and other "newer" courses are replacing COBOL instruction in some programs. In programs where the maximum number of hours is tightly constrained, the role of COBOL has been reduced in favor of Visual Basic, Java, Internet Programming, and similar courses that appear responsive to growing industry demands. As hard decisions are made in allocating language courses to the number of hours available for programming instruction, COBOL is continuing to lose favor. Faculty desire to teach COBOL continues to wane generally in favor of instruction in the exciting newer technologies.

A REVITALIZED COBOL

Let's look at the facts. All evidence appears to point to a definite place for COBOL-based applications in the foreseeable future. IT managers are insistent that COBOL continue to be an integral part of IS/CIS programs and nearly 90% of those interviewed [6] indicated the COBOL instruction should undergo modernization and that the instruction should include the object-oriented features and web-based features. But, these same individuals did not want the traditional topics removed from instruction; rather, they felt that instruction in the object-oriented capabilities and other modern features would complement (not replace) the standard structured COBOL typically taught.

Glass [9] sums up the criticality of COBOL in the text that follows. In comparing COBOL to Delphi, C++, Java, Visual Basic, and Power Builder, COBOL is still the "best language for developing information systems." Further, that while we continue to hear of the excitement of the web-based applications, "...the majority of leading IT systems continue along a moderate path while absorbing new technologies that contribute to practical, proven benefits." Glass goes on, "Discarding the novel and sensational, business, COBOL, and IT will continue to evolve together." [10] What, then, can the colleges and universities undertake to support the clear needs of the IT community?

There is clear evidence legislates IS/CIS programs need to include at least a two-course sequence in COBOL (one course does not provide the needed expertise). These two courses must continue to provide instruction in the traditional topics including various file organizations and accessing techniques, as these topics are necessary to maintain the huge inventory of legacy applications. But COBOL instruction must be immediately upgraded to include the modern features that are available in the COBOL 2002 Standard [11][12], and these topics must be inserted into classroom instruction now. Topics such as recursion and dynamic storage allocation must be taught and reinforced with appropriate programming assignments. Perhaps the highlight of the new standard is certainly the support of *object-oriented programmin*g. But beyond this, the new COBOL standard included a large number of other enhancements, such as support of Boolean and bit data, native binary and floating point data, generalized exception handling, an enhanced CALL mechanism, improved table handling, automatic and based data, user-defined functions, standardized compiler directives and many, many more. [13] Web-based features using, or example, COBOL Net Express (by Merant) [14] where students may create cobol.cgi files on their own personal web server needs to be presented. Further, coursework must include the COBOL object-oriented exercises to communicate the principles of encapsulation, inheritance, and polymorphism. Reuse principles must be taught, assigned and exploited in practical programming assignments. Introducing these and related topics will enliven COBOL instruction dramatically for both faculty and students. When there is common knowledge that many features and language facilities found in Java Script, Java, Visual Basic, and other more modern technologies can also be done in COBOL – without the need to learn a new syntax – COBOL may once again achieve its place in being the best overall language that meets the great functional diversity of business applications. It will certainly render its last years more meaningful.

But this is still not enough. It is not sufficient to provide enhanced instruction in the language courses and let things go at that. Rather, IS/CIS curricula must ensure that traditional courses in Systems Analysis, Design and Implementation, generally a two-course capstone sequence in many IS/CIS programs, affords the students the opportunity, if desired, to develop an application using either the newer technologies by themselves (such as a Visual Basic front end with back end database) or a web-based application based on the COBOL language facilities and accessing a (mainframe) database. Thus, students need to be provided a "legacy path" as well as a more-modern "OO-path" through the IS/CIS programs.

In the next 10 to 15 years, we will see dramatic change. According to Hotle, a Gartner Group vice president [3], COBOL and Visual Basic "each [now] account for 35% of new business applications development, with the remainder divided among a handful of languages." Hotle states "COBOL is still in … development and extension" and believes "COBOL will be a strong, viable language for the next 15 years." [3]

CONCLUSIONS

COBOL will underpin much of functional business processing for a number of years to come. IT managers have pointed out the immediate and projected needs for professional COBOL programmers, as they realize that no language yet can match COBOL's file processing capabilities. However, these same managers feel that incorporation of the newer technologies, that is, OO programming, COBOL's web-based facilities, and other technologies must be married in order to provide a comprehensive technology base to support critical legacy systems for at least another ten years, while at the same time supporting infrastructure and modern development paradigms to meet business and consumer demands in areas such as web-based applications and e-commerce.

Revitalizing COBOL in our college curricula has been analyzed and proposed suggestions offered. These suggestions are designed to put the excitement back into COBOL instruction while recognizing that the new capabilities within the COBOL 2002 standard support the need for modern, state-of-the-art software development approaches.

REFERENCES

[1] Arranga, Ed in Roundtable Discussion, "In Cobol's Defense," *IEEE Software,* 17(2), March/April 2000, 70-72.

[2] Stern and Stern, *Structured Cobol Programming,* 9th Edition, John Wiley, 2000,

[3] Arranga, Ed and Price, "Fresh from Y2K: What's Next for COBOL?",*IEEE Software,* 17(2), March/April 2000, 16-20.

[4] Bradley, John in Roundtable Discussion, "In Cobol's Defense," *IEEE Software,* 17(2), March/April 2000, 70-72.

[5] Ankrum, Scott, *The Cobol Report*, January 2001.

[6] Carr, Donald and Ronald Kizior, "The Case for Continued Cobol Education," *IEEE Software,* 17(2), March/April 2000, 33-36.

[7] Ulrich, William M., "Remember Cobol? If You Don't, Get Reacquainted," *Computerworld News and Feature Story, "* May 2001 http://www.computerworld.com/cwi/story/0,1199,NAV47_STO60683.00.html)

[8] Hardgrave, Bill C. and E. Reed Doke, "COBOL in an Object-Oriented World," *IEEE Software,* 17(2), March/April 2000, 26-29.

[9] Glass, Robert, *The Software Practitioner,* Nov/Dec 1996, 8.

[10] Glass, Robert, "COBOL: Is it Dying or Thriving?", *J. Systems and Software,* 39(3), 1997, 197-199.

[11] COBOL WORLD 2001 – Abstracts Track 06: Cobol 2002, http://cobolworld.com/trk6_abstracts.html

[12] "Object Oriented Cobol" – tutorial, Gene Webb, http://objectz.com/oocobol/tutorial

[13] "Cobol 2000" , Fujitsu Software Corporation http://www.adgtools.com/info/whitepaper/coby2k.pdf

[14] Merant Micro Focus Net Express, http://www.merant.com/products/microfocus/net_express