BRINGING COBOL BACK INTO THE COLLEGE IT CURRICULUM

Ed Lindoo, Ph.D.
H. Wayne Huizenga School of Business and Entrepreneurship
Nova Southeastern University, 3301 College Avenue,
Fort Lauderdale-Davie, Florida 33314-7796 | 800-541-6682
772-260-1910 direct, edlind@nova.edu

ABSTRACT

As shocking as this may sound to some, there are many reasons beyond great need to bring COBOL back into the CIS curriculum. IBM estimates that there are more than 200 billion lines of COBOL code still in use across industries such as retail, insurance and banking [5]. How can this be? COBOL died years ago with the advent of C++, C#, .NET and others…..right? Wrong! Airey [1] found that COBOL still runs over 70% of the world's businesses and surprisingly there are over 200 times more transactions processed daily by COBOL business applications than there are Google and You Tube searches every day! Further, Airey [1] stated that the average individual interacts with a COBOL based business application at least seven times every day.

INTRODUCTION:

Today, when we withdraw money from an ATM, place an airline reservation, or order a product over the Internet, there is a very good chance that a COBOL application has been used to process the transaction. In fact, one estimate from Murach [7] states that COBOL applications account for 60% of all the applications that are currently in operation world-wide. Another estimate says that these applications process 85% of all computer transactions. Even the Social Security Administration still has 60 million lines of COBOL code in-use, according to a 2013 government report [5].

Why are all these applications still running COBOL? In the late 90s, billions of dollars were spent on COBOL applications to make them "year two thousand" (Y2K) compliant. As this problem was fixed, the code was also updated so that it could continue to run indefinitely. Because of the tremendous amounts spent on Y2K along with the economic down-turn, IT budgets were greatly reduced during the past ten plus years. As a result, there was no funding available to purchase or to rewrite programs to eliminate COBOL; we find ourselves today with the statistics previously mentioned.

With the advent of mini and micro-computers along with the development of languages such as C++, C#, Java, .NET and operating systems like Linux, a lot of old COBOL, Fortran, Pascal, RPG (and others) code has been re-written to these other languages. Yet, COBOL still dominates by handling 95% of all ATM transactions, 80% of all point-of-sale transactions, managing over 20,000 retirement plans, the transportation of 72,000 freight containers, 5.5 million homebuyers and over 500 million mobile users every day of the week! [1]. Most of these applications are running on big mainframe computers, many of which were developed by IBM. IBM realizes the importance of this market and is working to keep COBOL applications on mainframes by making it easier for customers to migrate COBOL applications from old to new

mainframe hardware. To do this, IBM provides software products that make it relatively easy to "web-enable" COBOL programs without even changing the COBOL code. IBM also now offers software that allows Java and COBOL to work with the same application, calling this "Java interoperability" [7].

For example, the latest mainframe COBOL compiler (called "Enterprise COBOL for z/OS") offers a number of features for Java interoperability. Specifically, it provides object-oriented COBOL syntax that allows the programmer to create object instances of Java classes, invoke methods on Java objects, define Java classes with methods implemented in COBOL, and access Enterprise Java Beans that run on a J2EE-compliant EJB server like Websphere. "Enterprise COBOL for z/OS" also provides XML capabilities which will make it easier to work with data across platforms [7]. Likewise, Micro Focus Visual COBOL enables COBOL teams to work on the same projects as the Java or C# developers. It also provides an immediately-available learning environment to enable other programmers to pick up sparse COBOL skills [1].

A question that comes to mind is why not rewrite COBOL code into a newer language? The answer is simple, if it's been working for 30+ years, where is the incentive to switch? After all, companies have gone out of business because they ignored the old adage of "if it isn't broken, don't fix it." Many have tried to rewrite the old tried and true code. This has proven to cost them a lot of time and money, often failing. Second attempts cost even more and few of the new & improved languages managed to replace the old COBOL. "Many of these initiatives failed because the systems were (and still are) too big, too complex, too integrated into critical business processes, and working too well to replace" [2].

A survey by Jones [4], found that Java was the number one code used in on-going projects in Japan, followed closely by COBOL! The following chart is a result of the Jones survey.

Language	**Projects**	
Java	822	28.20%
COBOL	464	15.90%
VB	371	12.70%
C	326	11.20%
Other languages	208	7.10%
C++	189	6.50%
Visual Basic.NET	136	4.70%
Visual C++	105	3.60%
C#	101	3.50%
PL/SQL	57	2.00%
Pro*C	23	0.80%
Excel(VBA)	18	0.60%
Developer2000	17	0.60%
ABAP	15	0.50%
HTML	14	0.50%

(based on 505 projects in 24 companies)

THE NEED FOR COBOL PROGRAMMERS.

By the end of the 1990's, COBOL had all but disappeared from most college curriculums, replaced by C, C++, .NET and a host of other more "modern" languages. As demonstrated, COBOL hasn't gone away. What is going away is all of those retiring or soon to be retiring baby boomers who learned COBOL in college and wrote and supported all the programs that are still in use today. Merrett [6] cited a 2012 global survey by Vanson Bourne which found that CIOs in the UK believe 25% of their mainframe skilled staff will retire in the next five years leaving them with a serious need for staff with key skills in languages such as COBOL and PL/I.

As a result, we are starting to see a demand for COBOL programmers, a demand that will surely increase as more and more legacy programmers retire. As this happens there will be two distinct needs: Niche'….maintaining legacy code for years to come, and Development which will involve either interfacing COBOL code to newer languages or rewriting COBOL code into newer languages [7].

A search of several major job search websites conducted in January 2014 and again in May 2014 with keyword "COBOL" revealed the following:

Country	Site	Jobs 1/2014	Jobs 5/2014	% increase
U.S.	Monster	165	196	19
U.S.	Indeed	1960	2170	11
Europe	Eurojobs	11	18	64
England	Indeed	137	152	11
China	Sou.Zhaopin	95	101	06
Australia	Indeed	9	17	88
Canada	Indeed	163	176	08

While there is no scientific evidence here, there appears to be a significant increase in COBOL related jobs in this four month span. Considering that when searching these same sites for C++, .NET and Java the results were into the thousands, these numbers seem pale in comparison. However, Bloom [2], predicts that "over the next few years, new COBOL programmers are going to be in high demand." He goes on to say that in general, COBOL programming skill sets reside in baby boomers that have been programming in COBOL their entire career. Therefore, the real issue is that with baby boomers retiring and new college recruits lacking the skills to replace them, companies are realizing that if the software stops, so does their company.

As funny as it may sound, if you are looking for a job in IT, adding COBOL to your resume is going to be a plus that will most likely benefit you for a very long time. That is not to say that you want to become one of the COBOL "dinosaurs" and learn only COBOL, but rather learn COBOL to compliment the other languages you learn such as C++, C#, .NET and Java. Simply put, you need to learn COBOL so that when you're asked to re-write it into a newer language, you can go thru the COBOL program and understand exactly what's going on…..rewriting it into the new language. However, if you're interested in creating new solutions,

rather than slowly improving those that have been around for decades, COBOL probably isn't the language you want to go with.

Drilling deeper into the results from the sites searched, the majority of these listing are not looking specifically for COBOL only programmers, but rather C++, .NET and Java developers that “Understand other development platforms and tools including legacy COBOL in order to make recommendations on best approaches for integration and enhancement” [3]. A search of the American Express job site in January 2014 revealed that they are looking for a number of mainframe COBOL programmers. But they are also looking for system architects with the following skills as a requirement: “Proven experience with object-oriented design and coding skills across a variety of platforms (JAVA, HTML, BLAZE, DB2, XML, and Mainframe [JCL, COBOL, SQL, EASYTRIEVE]).”

Lowes, the home improvement company is looking for system architects with skills in “MQ, SOAP over HTTP, COBOL and Java/JEE.” There’s even a company in Naperville, Illinois, seeking an SQL programmer with the following statement in their job listing: “introductory knowledge of COBOL is a plus (to understand legacy applications)” [3]. The list goes on and on but it’s important to note that there are many jobs that are not specifically looking for COBOL programmers or developers. For example, the following are job titles found on “Indeed.com” that have a COBOL requirement within the job description. “Business Analyst”, “Java Developer”, “.NET Programmer”, “GIS Analyst/Developer”, “Web Business Analyst”, “Mainframe/Linux Conversion Specialist”, “Conversion Analyst”, “DataArchitect”, “Java - COBOL Programmer/Analyst”…..and the list goes on!

THE UNIVERSITY ISSUE:

At universities today, COBOL is rarely offered. In fact, a 2013 survey of 119 universities by Micro Focus [1] found that 73% of the universities do not offer COBOL as part of their curriculum, and of the remaining 27%, only 18% have it within the curriculum and 9% as an elective. For most students, this means that there is a very good chance they will attend a school that does not offer COBOL. Schools that offer COBOL tend to put it in classes that focus on business systems to give it context. At Syracuse, for instance, the first COBOL course is named "Enterprise Technologies." David Dischiave, an associate professor at the Syracuse University School of Information Studies, states that "Employers are knocking on our door trying to hire as many [COBOL-trained students] as they can." [8]. Syracuse also requires information technologies students to take Java courses. Other programming languages are offered as elective courses.

It's an entirely different situation at Carnegie Mellon University. At one time, the IS department was all COBOL and heavily involved in the IT curriculum. The one remaining hard-core COBOL class is an elective and has been renamed Introduction to Business Systems Programming. It includes topics related to enterprise IT systems to give students "a feel for what big business systems are like", said Ray Scott, director of systems and operations [8]. It’s hard to pin-point exactly why COBOL was removed from college curriculums, but the underlying

assumptions were that COBOL was going to be replaced by newer languages such as .NET and Java. No one realized the enormity of COBOL in the world, nor did they realize that most COBOL programs would be patched, not replaced for Y2K.

MICROFOCUS OFFERING

Believing that the industry needs its graduates to contribute where it matters most, the Micro Focus Company now offers a very attractive academic program for universities to re-adopt COBOL programming classes. The program is designed to specifically address the COBOL IT skill challenges facing many business organizations worldwide. It is designed "to promote and foster the education, enablement, and development of the COBOL language and business-based application development skills within today's Universities" [1]. Specifically, they offer to universities and students a free download of the Micro Focus Visual COBOL product for development. The license is good for one year. What this means for faculty is that they can bring COBOL back into their university at no cost. The software is free and can be downloaded and installed on lab computers, or alternately, students may install the software of their personal computers.

For Universities to get started to teach COBOL in the classroom, Micro Focus offers access to online software documentation and manuals, including a number of printed books and manuals, free access to Web based training courses, free online support through their community forums, and access to their contacts and opportunities forums. For administrators this means they just need to register into the program to receive and academic bundle of software. As a result, the only cost incurred is for the course to be developed……a real win-win for the school and its students. Student can get the free download at: https://www.microfocus.com/products/product-trials/index.aspx. Faculty and administrators can join the academic program at https://microfocus.com/forms/academicprogram.aspx

Many schools in a number of countries have signed on with Micro Focus to bring COBOL back into their curriculums. In the U.S., there are 117 schools in the alliance including Texas A&M University, The University of Texas, New York City College of Technology, Kansas State University, University of Wisconsin, University of Maryland, Illinois State University, Iowa State, and a host of community and technical colleges. In addition, colleges and universities around the world are also involved including schools in Argentina, Belgium, Brazil, Canada, China, Denmark, France, Italy, India, Philippines, Poland, Russia, Spain, UK and Venezuela [1].

CONCLUSION

As this research has found, COBOL is surprisingly still alive and well in society today, and it is not going away anytime soon. However, staff who have supported it since the 1950s are going away and the need for programmers who understand the language is increasing exponentially as the years go on. One of the basic needs of society is for people trained to the needs of society. Clearly the need to understand COBOL, from supporting, to interfacing, to completely re-writing to a newer language is there. As a result, the recommendation is that COBOL be taught in CIS college curriculums, at least as an elective because of the dominance it still has in the world. There is also a purely academic reason for offering COBOL courses.

COBOL provides a striking contrast to Java, C#, and Visual Basic. COBOL provides for structured programming versus object-oriented programming, self-documenting code versus cryptic code, one Read statement versus no easy way to read a record in a file and a few dozen COBOL statements versus thousands of classes, methods, and properties [1]. Giving students the opportunity to experience these differences allows the student to gain a better perspective on business applications, development methods, and programming languages. It is not suggested that COBOL take a priority over .NET or Java, but that all of these language be taught in the modern computing curriculum.

REFERENCES

[1] Airey, E. (2012). *Programming: Show Me The Money!* Business Computing World. Retrieved 1/3/14 at: http://www.businesscomputingworld.co.uk/programming-show-me-the-money/

[2] Bloom, E. (2013). *COBOL will outlive us all*. IT World. Retrieved 1/8/13 from: http://www.itworld.com/career/341879/cobol-will-outlive-us-all

[3] Dice.com (2014). Web search for COBOL programmers conducted 1/7/13

[4] Jones, D. (2012). Why is COBOL still popular in Japan? The Shape of Code. Retrieved 1/8/14 from: http://shape-of-code.coding-guidelines.com/2012/10/28/why-is-cobol-still-popular-in-japan/

[5] Kraft, D. (2013). COBOL: 10 Reasons the Old Language Is Still Kicking, eweek. Ziff Davis Enterprise, Foster City CA. Retrieved 1/3/14 at http://www.eweek.com/developer/slideshows/cobol-10-reasons-the-old-language-is-still-kicking#sthash.iePjWT4O.UW0Dj65f.dpuf

[6] Merrett, R. (2013) *CIOs struggle to deliver timely mobile business apps: survey*. CIO magazine. Retrieved 1/3/14 at: http://www.cio.com.au/article/462026/cios_struggle_deliver_timely_mobile_business_apps_survey/

[7] Murach, M. (2013) *The Future of COBOL*. The COBOL Center, infogal.com. Retrieved 1/3/14 from: http://www.infogoal.com/cbd/cbdz040.htm

[8] Thibodeau, P. (2013) *Should universities offer COBOL classes?* ComputerWorld. Retrieved 1/8/13 from http://www.computerworld.com/s/article/9238194/Should_universities_offer_COBOL_classes_