IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

Importance and Application of COBOL in Banking Sectors

Prem Mangesh Mundekar[1] and Prof. Divakar Jha[2]
Student, Department of MCA[1]
Mentor, Department of MCA[2]
Late Bhausaheb Hiray S. S. Trust's Institute of Computer Application, Mumbai, India

Abstract: *COBOL stands for COmmon Business Oriented Language. One of the first of the high-level languages, it was put together by a group sponsored by the Department of Defense to develop a common business language. That group came to be called CODASYL—the Committee on Data Systems Languages—and defined a "common business oriented language," Banks invested several millions in designing Mainframe systems architectures to respond to the business requirements. Whether that was a bank statement or an on-line money wiring transaction. COBOL became a reality. And became a liability. Moving from it meant to invest too much money in infrastructure and training and adding up the risk of fail on mission critical business functions. (I keep to my self that the most sensitive part of the human body is the pocket. And a Bank is nothing a huge pocket). This research paper will tell us about the importance and several applications of COBOL in banking sectors through several studies and their respective results.*

Keywords: COBOL, FS(Financial System), Banking.

I. INTRODUCTION

COBOL stands for Common Business-Oriented Language. Back the in late 1950s, the US Department of Defense sponsored its development, which included the support of the era's top tech companies like IBM, Sperry Rand, Boroughs and Honeywell. It is an imperative, procedural and, since 2002, object-oriented language. COBOL is primarily used in business, finance, and administrative systems for companies and governments. COBOL is still widely used in applications deployed on mainframe computers, such as large-scale batch and transaction processing jobs. However, due to its declining popularity and the retirement of experienced COBOL programmers, programs are being migrated to new platforms, rewritten in modern languages or replaced with software packages

II. RELATED WORKS

1. How COBOL is used in the 21st century?

Quite a number of companies, government organisations and other institutions still operate, maintain and update their COBOL applications from decades ago. Sometimes simply because they still work and it makes no economic sense to replace them with modern languages. Especially when the toolset of so many providers allows interfacing of COBOL applications with the modern languages.

2. Why is COBOL still important in 2021?

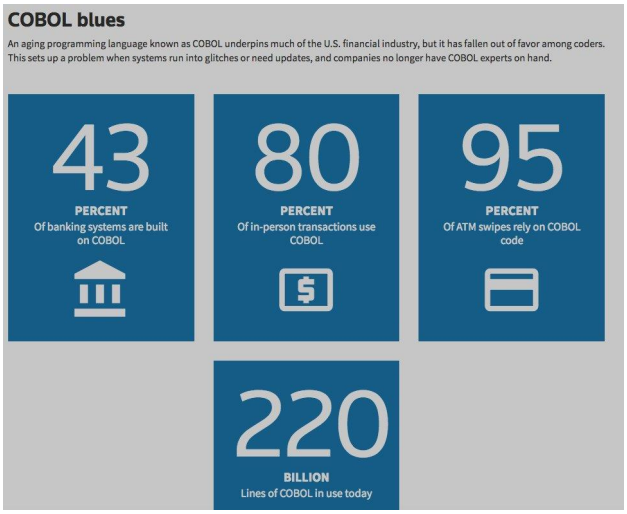
As clearly stated earlier. COBOL developers have inherited their codebase and also often their environment. Hence the newly built code is going into an existing environment. The reason COBOL is used by these businesses is simply because it was there in the first place. One could argue that if all those financial institutions successfully operate their businesses based on applications written in COBOL, then it must be a foundation for success. The reality is that many tasks that COBOL once was ideal for are now done by modern languages. Further the influence of the C language in modern programming languages have created a number of polyglot programmers that are capable of multiple languages and pick the one best suited for the job. So with an increased demand of a platform which can handle large volumes in the modern day and also with the utilization of next generation tools , languages and technology it makes sense that the harmonizing of "legacy" systems(IBM Mainframes + COBOL) and New Generation Technology will have your overall core systems

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

capable of scalability, reliability, security and the ability to offer stakeholders and customers relevant and intuitive systems to engage, transact and perform tasks which are needed depending on the nature of the business.

III. IMPORTANCE OF COBOL IN BANKING SECTORS

Despite being over 60 years old, COBOL remains the basis of the system code for the financial services (FS) industry. It was built in line with the development of mainframes during the 1950s and 1960s to meet shifting business needs. Back then, mainframes were designed for commercial use to process large amounts of data with a high computing power. Fast forward to 2020, many banks and insurance providers still run on mainframes and, in turn, modern COBOL. Undoubtedly, COBOL continues to play a central role in the FS sector, acting as the backbone to some of the world's most important organisations. Although COBOL has been constantly improved upon over its 60 years with millions invested annually, the language's enduring status ultimately stems back to its original design which has made it irreplaceable as the core language of business-critical computing. FS sector facilities operate with a large volume of transactions and users at any one time. Similar to when they were built 60 years ago, they need to be safe and robust, and have scalable data processing functionality. These systems need to match COBOL's capacity for accessing, processing, manipulating and reporting on huge amounts of data, significant arithmetic activity and working at speed. COBOL allows for constant modernisation of these core systems that would prove too hazardous to completely convert. If history has taught us anything, it's that introducing new functions to an IT system is rarely an error-free process. In the case of banking, any mistakes or downtime could potentially put data and information essential to transactions and securities trading at risk. In this sense, COBOL is key to helping IT departments prevent the risk of system failure and ensure customers' sensitive data is protected. Within banking, for instance, IT systems running on COBOL deliver too much value to be replaced.



- There are 220 billion lines of COBOL code still in use today.
- COBOL is the foundation of 43 percent of all banking systems.
- Systems powered by COBOL handle $3 trillion of daily commerce.
- COBOL handles 95 percent of all ATM card-swipes.
- COBOL makes 80 percent of all in-person credit card transactions possible.

As you can see, it's difficult to make it through a day without using a system that depends on COBOL. Bank accounts and check-clearing services, as well as public-facing infrastructures, like ATMs and traffic lights, still run on this code written decades ago.

"The second most valuable asset in the United States — after oil — is the 240 billion lines of COBOL," says Philip Teplitzky, who's slung COBOL for decades for banks across the U.S.

"The entire GDP of the world is in motion in the [banking] network at any moment in time," as Teplitzky notes. "A bank turns over twice its assets every day, out and in. A clearing bank in, say, New York, it could be more… So a huge amount

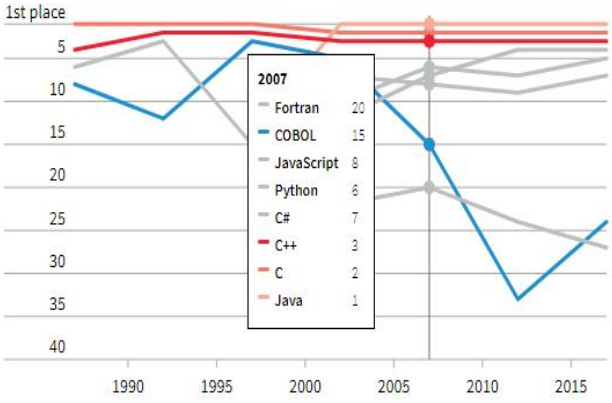
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

of money is in motion in the network and in big backend systems that do it. They can't fail! If they fail, the world ends. *The world ends.*

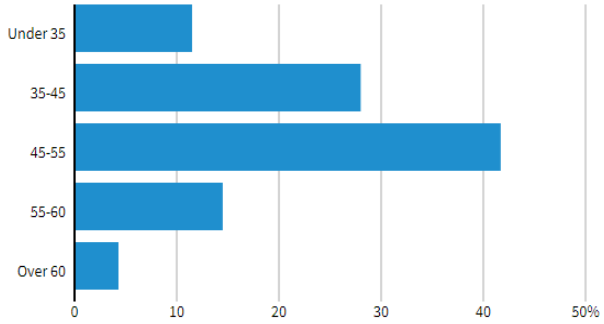
COBOL is not merely fast; it's also "stable, stable, stable", One of the processes that developed takes, every month, a file of about 2.4 million government pension and puts the proper amounts in people's bank accounts. "Bankers verify them and check them in 11 minutes. It hasn't failed in 20 years."

IV. SOME STATISTICS AND GRAPHS OF COBOL

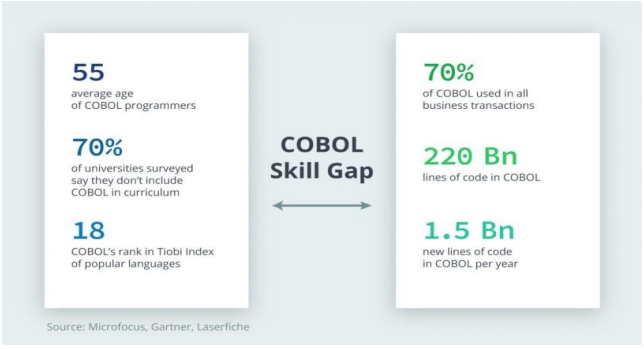
The popularity of COBOL has seen an uptick in the last five years, but has fallen dramatically overall in the last three decades



Average age of Developers On average, COBOL programmers are most likely to be between 45-55 years old.



Over 80% of in-person transactions at U.S. financial institutions use COBOL. Fully 95% of the time you swipe your bank card, there's COBOL running somewhere in the background. The Bank of New York Mellon in 2012 found it had 112,500 individual COBOL programs, constituting almost 350 million lines; that is probably typical for most big financial institutions. But we have were less COBOL young developers as it is an old language. From few years percentage of developers under 35 that is young developers are increasing due to need of many financial institutions and IT industry.



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

- Average age of COBOL developers are 55.
- 70% of universities surveyed say they don't include COBOL in curriculum.
- COBOL ranks 18th in Tiobi index of popular languages.
- 70% of COBOL used in all business transactions.
- 220 Billion lines of code are there in COBOL.
- 1.5 Billion new lines of code in COBOL per year.

There is a turnover rate of 20% or more among offshore COBOL outsourcers; resulting in a lack of continuity and knowledge. The people who become COBOL programmers don't have the same expertise as front-end developers. COBOL developers are trained in structured design and coding. In addition, they are used to working on large systems with millions of lines of compiled code. COBOL maintenance and programming are migrating to new locations: Poland, the Baltic countries, Russia, etc., which don't have the cultural familiarity with the business rules to be fully capable.

If the continued use of COBOL and mainframes is, as we believe, inevitable, or at least until someone produces an alternative that is as efficient and effective, the most appropriate solution is to train a new generation of COBOL programmers and help them understand these three fundamental skills:

- IBM Z operations—how to write, test and operate applications and data structures in an IBM Z environment
- Programming—the understanding and ability to design programs and understand the fundamental skills needed to be a programmer – this is a topic unto itself
- COBOL—the structures and nuances of the COBOL language.

V. APPLICATIONS OF COBOL

As you can see, it's difficult to make it through a day without using a system that depends on COBOL. Bank accounts and check-clearing services, as well as public-facing infrastructures, like ATMs and traffic lights, still run on this code written decades ago.

Every time you swap your ATM Card you are actually executing the COBOL program.

When your boss hands you your paycheck, odds are it was calculated using COBOL.

If you invest, your stock trades run on it too.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

Wonder why, when you're shopping at a retailer you will see a clerk typing into an old-style terminal, with green text on a black background? It's because the inventory system is using COBOL.

Several large enterprises, organizations, banks, financial firms, insurers, and industrial sectors such as healthcare, retail, automotive, shipping services, and others use COBOL for various reasons. For example, companies such as IBM, UPS, Fiserv, Bank of America, JPMorgan Chase, and Cigna still depend on COBOL. Thus, the language continues to play a crucial role in business computing that drives the global economy.

Despite the lesser developments in its versions, COBOL remains the language of choice across diverse markets and business lines. According to a Feb 2022 global survey by Micro Focus, around 92 percent of the survey-takers (software engineers, developers, IT executives, and architects from 49 countries) opined that COBOL applications are developed today to serve the strategic purpose of their respective organizations.

Our dependency on systems that still run on COBOL is astonishing. A report from Reuters in 2017 shared the following jaw-dropping statistics:

- There are 220 billion lines of COBOL code still in use today.
- COBOL is the foundation of 43 percent of all banking systems.
- Systems powered by COBOL handle $3 trillion of daily commerce.
- COBOL handles 95 percent of all ATM card-swipes.
- COBOL makes 80 percent of all in-person credit card transactions possible.

A good example of this could be witnessed during the pandemic. In the early days of Covid-19, businesses shut down en masse. Laid-off employees swarmed online to apply for unemployment benefits, and the websites for many state governments crashed under the load. In New Jersey, the governor told the press that their COBOL systems desperately needed help to deal with the new demands. "Literally, we have systems that are 40-plus-years-old," he noted.

Banks need to hope that those old-timers hang on as long as possible**.** Because there aren't a lot of new young kids learning COBOL these days.

VI. WHY COBOL IS NOT GETTING REPLACED?

Innovation among the big banks is slow at best, and it's easy to see why – the technology that promised to liberate banking in the 1960s has today cemented the industry in analogue processes. But when asked the question "why do 80% of transactions and 95% of ATM swipes still rely on COBOL?", the answers from those in banking have ranged from "if it ain't broke, don't fix it" to "it's too big, too risky, and too expensive to change". Imagine you're the CEO of a major bank – would you bet on a migration project that could destroy your entire company if it went wrong?

The risk aside, the economic argument to migrate to modern systems is also a tough one. The money that's been pumped into keeping these mainframes and COBOL applications operational is astounding. Should institutions throw it all away and start again while that COBOL code is still running and functional? That's a hard pitch to a board that probably isn't particularly technically inclined. A *COBOL migration won't be cheap, nor fast.*

Upgrading these legacy systems isn't as simple as it sounds. The systems are vital, 24/7 fulcrums on which the financial, governmental, and business worlds pivot. The code is old, multilayered, and, often, poorly or completely undocumented. It also has to work, all the time. The prospect has been compared to taking the propellers off an aircraft and trying to fit it with jet engines—while airborne.

Things can't stay as they are, but the prospect of doing something about it is hardly appealing. Nevertheless, the only

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

way things are going to get better is to conduct controlled, careful migrations to modern soft- and hardware.

To achieve that without disruption, data loss, and downtime will require modern expertise and money, which is 50 percent of the equation. The other half is COBOL expertise and time. Unfortunately, those are the two ingredients we're almost out of.

The programmers who know COBOL are either retired, thinking about retiring, or dead. We're steadily losing the people who have the skills to keep these vital systems up and running.

VII. CONCLUSION

COBOL has been the bedrock of the FS industry since its inception and will no doubt hold this position well into the future as the language continues to evolve.

Collaborations between business and academia such as the COBOL Academic Program, alongside the 16,000 strong COBOL programmer Facebook community, growing range of user forums, training groups and other bodies are bringing the language to the next generation of IT talent.

Through this continual evaluation and renewal, COBOL will be a key part of the IT landscape for many years to come, within the FS sector and beyond.

The world runs on old technologies and old software. Being battle-tested and reliable will always trump the new shiny thing. On the other hand, old technologies inevitably go through stagnation and sooner or later something needs to change. And with change comes opportunity. **COBOL** is the workhorse of the bank industry, there is no denying that.

The mainframe modernisation business arena is worth hundreds of billions of dollars. Big firms are waking up to the fact that soon they will have a black box of code that no one could change or understand. But that could provide the opportunity for those with enough preparation.

VIII. LITERATURE REVIEW

With the help of this research paper we have highlighted some key features, importance and applications of COBOL programming language in Banking and Financial Institutions. We have gone through with several existing research papers and some financial news channels and articles and some of the experienced COBOL developers and Bank clerks to make this research paper fruitful. This is our small attempt to let people know the importance and applications of a programming language that controls their money and to clear the perception of many people that COBOL is ended.

IX. ACKNOWLEDGEMENT

We would like to acknowledge the University of Mumbai, Mumbai, India to give us the opportunity to do the research work under the title "Importance and Applications of COBOL in Banking Sectors". Also, we would like to acknowledge the college L.B.H.S.S. T's ICA Bandra East, Mumbai, India to support us during the research process.

REFERENCES

[1]. Dreamixblogs https://dreamix.eu/blog/dreamix
[2]. Global Banking and Finance review, https://www.globalbankingandfinance.com
[3]. Wealth Simple Magazine, https://www.wealthsimple.com/
[4]. The Micro Focus COBOL Academic Program, https://www.microfocus.com/cobol-academic-program/
[5]. Reuters graphics, http://fingfx.thomsonreuters.com/
[6]. Mainframe Banking Application Case Study by GenRocket
[7]. Problems with COBOL--Some Empirical Evidence D. M. Volpano Herbert E. Dunsmore Purdue University, dunsmore@cs.purdue.edu
[8]. Program Analysis Environment for Writing COBOL Aspects Hideaki Shinomi Hitachi, Ltd.