Review of Milestone-Proposal: The Compiler Milestone

Ragib Hasan, Ph.D., IEEE Member
Associate Professor, Department of Computer Science
The University of Alabama at Birmingham
Email: ragib@uab.edu

Abstract: *The Compiler is one of the fundamental technologies in the history of computing. The invention of the compiler made it possible to write high level computer programs, which in turn led to increased programmer productivity and reduced rates of error. This review examines the Milestone-Proposal for recognizing the invention of the compiler.*

1. Introduction

The "Milestone-Proposal: The Compiler, 1952" [5] is to recognize the invention of the Compiler in 1952. The plaque citation states the following:

> *"The compiler translated human-readable English keywords or commands into machine-readable instructions or code, thus creating well-defined communication between human programmers and computers. It made possible programs written for different computers rather than a single machine. Based on the compiler, the Flow-Matic was first used in the UNIVAC. This shaped the creation of COBOL, a programming language that facilitated the information highway and the beginning of the IT industry."*

In this review report, we examine the citation and supporting information as presented in the Milestone-Proposal, and evaluate this in terms of the following three criteria:

1. Is the suggested wording of the Plaque Citation accurate?
2. Is the evidence presented in the proposal of sufficient substance and accuracy to support the Citation?
3. Does the proposed milestone represent a significant technical achievement?

The rest of this review is organized as follows: in Section 2, we provide the historical background of the invention of the Compiler technology. In Section 3, we evaluate the milestone proposal based on IEEE's three milestone evaluation criteria. And finally, we conclude in Section 4 and provide our final evaluation of the milestone proposal.

2. Background

Early computers were essentially high-performance calculating devices, which allowed high-speed calculation based on a predefined sequence of operations [6]. However, programming such computers was a difficult task, requiring mastery and knowledge of the hardware operations [6,7]. The "programming" was performed directly in machine-readable binary or octal code which would be manually loaded into the computer. There were multiple limitations of this approach: writing binary or octal machine code sequence was a very complex and difficult task which was prone to errors. Only a few programmers were capable of programming in binary or octal code. Also, the dependence on machine language to write programs limited the use of the program to only that specific computer [6].

Subsequent development of assembly language concept improved the situation slightly, allowing programmers to first write computer operations in terms of hardware commands (e.g., ADD or LOAD), which will then be translated into machine language. This made the programs slightly better to understand. However, the assembly language instructions were still essentially machine instructions written using specific keywords. Converting the higher-level application to assembly language would still appear to be an art, requiring specialized skills about the particular computer hardware where the program will be run. Also, assembly language programs were specific to a given computer and could not be ported to other computers without starting over [6].

Real life applications required a way to easily translate a real-life problem, described in human language (e.g., English) to the mathematical logic and then assembly language code sequences. In the early part of the 1950s, this essential technology was missing, and early computers were all programmed with machine language or assembly language [6,7].

In this backdrop, researchers explored the creation of tools that would take a higher-level specification written in English or other easily human-understandable language and translate it into the corresponding assembly language and machine language instruction sequence. The advantage of this approach is that it made programs easier to write by a wide variety of programmers and did not require a detailed knowledge of the underlying hardware. Also, it made programs portable to many different computer architectures as the higher-level specifications could be machine-agnostic.

One of the earliest efforts in creating these tools was Grace M. Hopper's development of "The Compiler", a program that could convert higher level mathematical functions into lower-level assembly and machine language code [2,3,4]. While Corrado Bohm in his seminal 1951 PhD thesis [1] wrote about the concept of such a program, it was conceptualized for an abstract machine, rather than implemented on any real computers of the time. Therefore, in 1952, there were no implemented tools that could perform the unprecedented task of converting higher level specifications to a machine-readable code. This is where Hopper's invention of "The Compiler" comes in as the ground-breaking advance in the young field of computer science [4]. Hopper developed the A-0 system for the UNIVAC I computer. Technically, the A-0 system performed more like a linker or loader than the modern notion of compilers as used now. However, it allowed composing multiple previously written subroutines into a complete program which could then be automatically loaded into the computer [2]. In this sense, the program, for which Hopper coined the term "The Compiler", was a pioneering invention indeed as it automated the creation of computer programs from a given specification. Something previously only possible by manual stitching of machine or assembly language code could now be automated. This tool and the subsequent improvements ultimately allowed the development of the COBOL programming language, which was very close to English and was among the first mainstream high-level programming languages in history [4]. Considering this and the consequent revolution in programming language and software development, the invention of "The compiler" in 1952 is indeed a milestone in the history of computing.

Next, we take a look at each of the milestone criteria and examine whether the Milestone-proposal meets them.

3. Review of the Milestone based on the 3 Criteria

3.1 Criteria 1: Is the suggested wording of the Plaque Citation accurate?

In the plaque citation provided in the Milestone-proposal [5], "The Compiler" developed in 1952 is credited as the tool that allowed higher level constructs or program structures to be converted into machine instructions in an automated manner. This is indeed correct. The latter part of the citation credits "The Compiler" as the basis of the "Flow-matic" system, which was an early English-like data processing system later developed by Hopper. In [2], Denise Gurer credits Hopper for creating the "*first compiler*". Gurer writes, "*Hopper supervised the department that developed the first compiler, A-0, and its successor, A-2. Hopper was also responsible for developing the FLOW-MATIC programming language, the only implemented business data processing language at the time*". This demonstrates the importance of "The Compiler" and supports the wording of the plaque citation. The view is also shared by Jean Sammet in [3], where she writes "*Thus, in my view, without **FLOW-MATIC** we probably never would have had a COBOL. The practical experience of implementing and using that type of language was priceless."*

Therefore, I conclude that the suggested wording of the Plaque Citation is accurate.

3.2 Criteria 2: Is the evidence presented in the proposal of sufficient substance and accuracy to support the Citation?

To support the milestone, the proposal presents several items as evidence. This includes an article published in Yale News commemorating the life and legacy of Grace M. Hopper [8], the IEEE Computer's web profile of Grace M. Hopper (stating her 1970 Harry Goode Memorial Award and 1979 W. Wallace McDowell Award) [9], the Wikipedia biography of Hopper [10], the Encyclopedia Britannica biography of Hopper [11], and a 2015 article published in the IT Professional magazine [4]. In my opinion, these citations strongly support "The compiler" by Hopper in 1952 to be the first instance of a compiler program. All the cited sources support the plaque citation statement. In particular, in the last citation [4], Strawn et al. laud Hopper's contribution and states that, "*Hopper was perhaps the first person to believe that computers should speak human-like languages, rather than requiring humans to speak computer languages. This belief began to be put into action in 1952 with the creation of the A-0 compiler*". In fact, Strawn et al. praises Hopper as the "*Mother of COBOL*" [4].

In review of the aforementioned evidence, I conclude that the evidence presented in the proposal are of sufficient substance and accuracy and the evidence supports the Plaque citation.

3.3 Criteria 3: Does the proposed milestone represent a significant technical achievement?

The proposed milestone of the invention of the Compiler in 1952 indeed represents a significant technical achievement. As stated in Section 2, Hopper's "Compiler" (the A-0 system) was the first automated program synthesis tool. Previous efforts such as that by Bohm et al. were based on abstract computers whereas Hopper's *Compiler* was the first such tool that was implemented in a real computer (the UNIVAC I).

The compiler paved the way for the creation of systems such as the Flow-matic, and subsequently, the COBOL language in the late 1950s. All of these allowed the ultimate computer and information revolution that has since then shaped the world for the better. K. W. Beyer in the book "Grace Hopper and the Invention of the Information Age" [12] credits Hopper as the pioneer whose inventions, including the "Compiler", led to the world-changing advances that has had a significant impact in the entire human race since then.

Therefore, I conclude that the proposed milestone indeed represents a significant technical achievement.

4. Conclusion

In this review, we analyzed the Milestone-proposal on the invention of the Compiler in 1952. By validating the presented evidence and cross-checking with independent references, we come to the conclusion that the milestone proposal commemorates a significant invention which is a true technical milestone in the history of computing. This invention serves as one of the foundations of the computing revolution and it should be celebrated as a notable achievement and as a milestone. Therefore, we support this proposal and look forward to the designation of the invention of "The Compiler" as an IEEE Milestone.

References

1. Böhm, Corrado (1954). Calculatrices digitales: Du déchiffrage de formules logicomathématiques par la machine même dans la conception du programme (PDF) (PhD) (in French). Zurich: ETH Zurich. Retrieved 27 September 2022.
2. Gurer, Denise (1995) "Pioneering Women in Computer Science" Communications of the ACM. 38(1), pp. 45-54.
3. Sammet, J. (1992). Farewell to Grace Hopper—end of an era!. *Communications of the ACM*, *35*(4), 128-131.
4. Strawn, G., & Strawn, C. (2015). Grace Hopper: Compilers and Cobol. *IT Professional*, *17*(1), 62-64.
5. Milestone-Proposal: The Compiler, 1952" at, https://ieeemilestones.ethw.org/Milestone-Proposal:The_Grace_M._Hopper_Milestone
6. Wilkes, M. V. (1968). Computers then and now. *Journal of the ACM (JACM)*, *15*(1), 1-7.
7. Davis, M. (1987). Mathematical logic and the origin of modern computers. In *Studies in the History of Mathematics* (pp. 137-167).

8. https://news.yale.edu/2017/02/10/grace-murray-hopper-1906-1992-legacy-innovation-and-service
9. https://www.computer.org/profiles/grace-hopper
10. https://en.wikipedia.org/wiki/Grace_Hopper
11. https://www.britannica.com/biography/Grace-Hopper
12. Beyer, K. W. (2012). *Grace Hopper and the invention of the information age*. MIT Press.