# The Real Creators of Cobol

*The author presents a few of the highlights of early Cobol development, in particular dispelling the myth that Grace Hopper created it. The article outlines the creation and mission of the committee that developed the language in 1959 and a few of the major inputs and influences on Cobol's initial development.*

Jean E. Sammet

In the 40 years since the first Cobol specifications were publicly released, there have been a number of misunderstandings—and even some myths—about how Cobol was created and about Grace Hopper's actual role in that process. This article attempts to clarify and correct some of the false statements that are frequently issued.

I based this article on my personal participation in the original Cobol committee. More importantly, I still have all the crucial documents that were developed during the committee's deliberations. The proceedings of the ACM SIGPLAN History of Programming Languages Conference (HOPL), held in 1978, includes a detailed paper describing Cobol's creation.[1]

## The Very Early History

A committee known as the Short-Range Committee created Cobol in 1959. The committee contained representatives from six computer manufacturers—Burroughs, IBM, Minneapolis-Honeywell, RCA, Sperry Rand, and Sylvania Electric Products—and representatives from three government agencies—the US Air Force, David Taylor Model Basin, and National Bureau of Standards. The committee chair was from the NBS.

The Short-Range Committee came into existence as a result of a meeting held in the Pentagon in May 1959, organized by Charles Phillips of the US Department of Defense. At that meeting, three committees were defined: Short-Range, Intermediate-Range, and Long-Range. All three were under the overall guidance of a self-appointed Executive Committee. (The Long-Range Committee never came into existence.) The mission given to the Short-Range Committee was very long, but its most crucial portion said that the committee should "…recommend a short-range composite approach (good for at least the next year or two) to a common business language…." The full mission statement did *not* tell us to create a new language. The intent was that the Short-Range Committee would do a quick job that would last a short time while the Intermediate-Range Committee was to take its time and develop a business language that would be suitable for a long time; however, the latter group never did this because Cobol became entrenched. As I said in 1981, "I believe that most of us viewed our work as a

 0740-7459/00/$10.00 © 2000 IEEE

## Flow-Matic and Cobol

In my view, the most significant technical contribution Grace Hopper made was the concept of Flow-Matic (originally called B-0) and the leadership of its design and implementation. In attempting to develop a language suitable for business data processing, she realized that although mathematics had a relatively common vocabulary and abbreviations (for example, sin, cos, $x + y$), there was no similar common terminology for data processing. Thus, she said—in several informal papers and articles—that full English words should be used for data names (for example, `unit-price`, `discount`, `inventory`) and commands (for example, `count`, `divide`, `replace`). Furthermore, although mathematical problems could generally be stated and solved using only fixed and floating-point data representation, data-processing problems required a system that permitted the description of user-defined data types. Flow-Matic development started in 1955, and manuals and a system were generally available by 1958. It was used for practical work by several companies, including the Metropolitan Life Insurance Company. People from Met Life reported on the work at the Automatic Coding Symposium held January 1957 at the Franklin Institute in Philadelphia.[1]

Grace Hopper's role in Cobol has been generally misunderstood, and I would like to take this opportunity to correct the incorrect statements and impressions that have consistently been conveyed in almost all articles and books, and even by a misleading Navy commendation. These comments are based on original Cobol records from 1959, which I still have and are reported in detail in my paper, "The Early History of Cobol."[2] A draft of that paper was sent to many people—including Grace—for comments, and she generally agreed with what I said.

Grace was one of a group of six people who met in April 1959 and decided to suggest to Charles Phillips in the Department of Defense that he convene a meeting to consider the development of specifications for a common business language. She attended the meeting called by Phillips in May 1959, along with approximately 40 other people, including myself, from business, government, and academia. That meeting established the Codasyl Executive Committee and the Short-Range Committee, as well as other committees. Grace was one of two technical advisors to the self-appointed Executive Committee (the other being Robert Bemer from IBM.)

Under the aegis of and with minimal guidance from the Executive Committee, the Short-Range Committee defined the Cobol specifications by December 1959. There were initially nine members (including myself), and eventually over 25 people participated in some phase of the basic Cobol language design: this large group included two people who worked for Grace, but Grace herself was not a member of the committee that defined Cobol. She did not participate in its work except through the general guidance she gave to her staff who were direct committee members. Thus, while her indirect influence was very important, regrettably the frequent repeated statements that "Grace Hopper developed Cobol" or "Grace Hopper was a codeveloper of Cobol" or "Grace Hopper is the mother of Cobol" are just not correct.

Grace's primary contribution to Cobol was indirect, and via Flow-Matic. It was the only business-oriented programming language in use at the time the Cobol development started (aside from Aimaco, a dialect of Flow-Matic). Without the existing practical use of Flow-Matic, I doubt that we would have had the courage to develop a language such as Cobol. (The other significant input to the early Cobol work was Commercial Translator, a set of specifications from IBM, but it had not yet been implemented.) 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. This is a major contrast with the mathematical area, in which there had been many small attempts at a high-level language going back as early as 1952.

Grace spent a lot of time convincing managers in various companies of the feasibility of Math-Matic, Flow-Matic, Cobol, and other high-level languages at a time when this was a unique and generally uncomfortable concept. She led her own group in the very practical "race" with RCA to produce the first Cobol compiler and demonstrate machine independence. Both companies demonstrated their successful results in December 1960.

### References


1. *Automatic Coding,* Monograph No. 3, J. Franklin Inst., Philadelphia, Pa., Apr. 1957.
2. J.E. Sammet, "The Early History of Cobol," *History of Programming Languages*, R.L. Wexelblat, ed., Academic Press, ACM Monograph Series, New York, 1981, pp. 199–241.




stopgap measure—a very important stopgap indeed, but not something intended for longevity."[2]

The Short-Range Committee worked diligently from June 1959 on, but there were great difficulties in having a fairly large committee try to create a programming language. In November, the Short-Range Committee chair appointed six people to develop specifications for consideration:

- William Selden and Gertrude Tierney (IBM),
- Howard Bromberg and Norman Discount (RCA), and

- Vernon Reeves and Jean E. Sammet (Sylvania Electric Products).

We worked for two full weeks (including some round-the-clock sessions) in November 1959 and sent the proposed specifications to the full Short-Range Committee, which accepted almost all of them. After some editing (by the same six people), we turned in the specifications as a final report in December to the Executive Committee, which accepted them in January 1960. After some further editing, the Government Printing Office issued *Cobol 60*.[3]

## Grace Hopper's Role

For those who have read this far, some of you may be wondering about the role Grace Hopper played. I yield to no one in my admiration for all the work that Grace Hopper did over many years. However, she did *not* create or develop Cobol. Unfortunately, this has been one of the misunderstandings and myths created over the years. Hopper actually had two effects on Cobol. The first, and in my view the most important, was her leadership in developing the first high-level programming language for business data processing called Flow-Matic; this language was made available by Remington Rand Univac to its customers in 1957–58. Grace's second effect was her general guidance to the two members of her staff serving on the Short-Range Committee. Grace herself *never attended* any committee meetings.

To establish an accurate historical record, I included a lengthy description of Hopper's relationship to Cobol in the obituary I wrote when she died in 1992 (see the edited excerpt in the sidebar "Flow-Matic and Cobol"[4]). As I said there, "Without that existing practical use of Flow-Matic, I doubt that we would have had the courage to develop a language such as Cobol."

It continues to amaze me that the language we created in six months has continued to grow and evolve over a 40-year time period. Some people have said it is the most widely used programming language (at least in the US). I cannot verify that statement, but there are certainly many millions (and perhaps billions) of lines of Cobol code that have been written since its inception, and many of those programs are still in use. One reason for Cobol's longevity is the fact that the committees responsible for its development and standardization have added features over the years to keep up with newer technology and ideas.

It is worth noting that numerous reporters who tried to explain the Y2K problem to the general public somehow got the impression that the Y2K problem was caused by Cobol. This is absolutely false; Cobol allows the creation of a date in any way that the programmer wants, and the year can be designated using two digits, four digits, or even seconds if anybody was foolish enough to try that.

Our "short-range" language good "for a year or two" has certainly outlasted its original intent by a great many years, and probably will continue to be used well into the 21st century.

## About the Author

Jean E. Sammet is a programming language consultant. While working at Sylvania Electric Products, she was a key member of the Short-Range Committee that developed Cobol in 1959. She joined IBM in 1961, where she initiated the concept and directed the development of Formac (*For*mula *Ma*nipulation *C*ompiler), the first widely used general language and system for manipulating nonnumeric algebraic expressions. Among other technical and management assignments, she also coordinated the strategy and actions for IBM's Federal Systems Division's use of Ada and served on various DoD groups assisting the development of Ada. She took early retirement in 1988.

Sammet served as ACM president and vice president during the 1970s. She was editor-in-chief of *ACM Computing Reviews* and of the *ACM Guide to Computing Literature*, program chair for the first and second ACM SIGPLAN History of Programming Languages Conferences, and chair of the ACM Special Interest Groups on Symbolic and Algebraic Manipulation (SIGSAM) and on Programming Languages (SIGPLAN). She is the author of *Programming Languages: History and Fundamentals* (Prentice Hall, 1969).

She is a member of the National Academy of Engineering and was in the initial group of ACM fellows. She received an honorary ScD from Mount Holyoke College, the ACM Distinguished Service Award, and the Augusta Ada Lovelace Award from the Association for Women in Computing. She may be contacted at PO Box 30038, Bethesda, MD 20824; sammet@acm.org.

## References

1. J.E. Sammet, "The Early History of Cobol," *History of Programming Languages*, R.L. Wexelblat, ed., Academic Press, ACM Monograph Series, New York, 1981, pp. 199–241.
2. J.E. Sammet, "The Early History of Cobol," *History of Programming Languages*, R.L. Wexelblat, ed., Academic Press, ACM Monograph Series, New York, 1981, p. 203.
3. U.S. Dept. of Defense, *Cobol, Initial Specifications for a Common Business Oriented Language*, #1960 0-552133, US Government Printing Office, Washington, DC, Apr. 1960.
4. J.E. Sammet, "Farewell to Grace Hopper—End of an Era!" *Comm. ACM*, Vol. 35, No. 4, Apr. 1992, pp. 128–131.