

Concerning Go To Considered Harmful

COMP110 - Research Journal

1604281

2017-11-13

1 Introduction

Edsger Dijkstra's 1968 letter, Go To Statement Considered Harmful [1], is a now famous argument for the removal of the GoTo statement from programming languages, with the exception of "perhaps, plain machine code" [1, p. 147]. Dijkstra's strong wording and bold opinions caused a stir in the academic community and led to many further debates on the validity of the use of GoTo; eventually culminating in a shift towards Structured Programming and the more familiar programming constructs we see today.

2 GoTo

The GoTo statement is a common feature of many programming languages even today when use of it is not widespread. Using GoTo will move the program execution to a new part of the program, designated by line-number or a reference to a labelled section. Navigating a program in this way does hold certain benefits: Use of GoTo statements can improve readability of code, particularly since it eliminates the need for extra variables. [2]

However, one of the primary reasons for the use of GoTo was a lack of block structures - self contained blocks of code that could contain both instructions and data. The absence of block structures necessitated the use of GoTo statements in early computing and the traditional reliance on it led to a reluctance to adopt other methods. (CITE) In addition, it has been argued that primitive control structures such as GoTo provide the building blocks for more complex functionality and allow inventive programmers to come up with new structures - thus progressing computing languages. [2]

3 Context

At the time of Dijkstra's letter, GoTo was widely used in programming in place of more modern structures such as Case statements and conditional if-else clauses. This was despite the existence of block structures and other "Structured Programming" constructs in ALGOL 58 and ALGOL 60. [3] However, Dijkstra was not the first to voice concerns over flagrant use of the GoTo statement. Already, it had been proven that any program processable by a Turin Machine could be constructed without the use of GoTo statements (albeit with new variables and rewriting necessary), [4] however this was not yet regarded as reason enough to abandon traditional programming techniques.

4 Influence

The publication of Dijkstra's letter marked the beginning of what would become Structured Programming, a philosophy advocating the use of block structures and subroutines over GoTo statements. However, it also sparked massive controversy in the world of academic computing that continued for years after, with advocates on both sides of the debate. [5] [6] [7] As Structured Programming became more popular, some critics pointed out that while it could be a useful tool, it did not miraculously fix all coding problems. Attention was drawn to Dijkstra's strong negative opinions of the GoTo statement, as well as the

narrow-minded beliefs of many Structured Programming followers regarding the use of GoTo. [7]

5 Conclusion

Overall, although his criticisms of GoTo were not necessarily unique or even incredibly revolutionary, the attention his argument gained brought the issue into the spotlight. Although his highly bold, passionate statements in the letter such as - “disastrous effects” and the need to “abolish” GoTo statements [1, p. 147] - made his argument infamous in academic circles, this ironically furthered his cause. The notoriety he gained and the publicity his argument was subject to, undoubtedly contributed to the fall of traditional programming. The resulting discussion and research into the topic were instrumental in the development of more sophisticated programming methods. (CITE)

In modern programming, the utility of GoTo in program structure is almost fully overshadowed by Object Oriented approaches which allow even complex programs to be abstracted into readable and easily-adapted forms. (CITE) However, there is still value in the availability of GoTo for specialist cases, such as fall-through in C case statements or error handling in nested loops. [8]

References

- [1] E. W. Dijkstra, “Letters to the editor: Go to statement considered harmful,” *Commun. ACM*, vol. 11, no. 3, pp. 147–148, Mar. 1968. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/362929.362947>
- [2] M. E. Hopkins, “A case for the goto,” *SIGPLAN Not.*, vol. 7, no. 11, pp. 59–62, Nov. 1972. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/987361.987371>

- [3] A. J. Perlis, “The american side of the development of algol,” *SIGPLAN Not.*, vol. 13, no. 8, pp. 3–14, Aug. 1978. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/960118.808369>
- [4] C. Böhm and G. Jacopini, “Flow diagrams, turing machines and languages with only two formation rules,” *Commun. ACM*, vol. 9, no. 5, pp. 366–371, May 1966. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/355592.365646>
- [5] R. J. DuWorks and S. W. Smoliar, “The arrogant programmer: Dijkstra and wegner considered harmful,” *SIGCSE Bull.*, vol. 4, no. 4, pp. 19–21, Dec. 1972. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/382224.382454>
- [6] D. E. Knuth, “Structured programming with go to statements,” *ACM Comput. Surv.*, vol. 6, no. 4, pp. 261–301, Dec. 1974. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/356635.356640>
- [7] P. Abrahams, “”structured programming” considered harmful,” *SIGPLAN Not.*, vol. 10, no. 4, pp. 13–24, Apr. 1975. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/987278.987280>
- [8] R. Doss, “Using goto statements,” *SIGSOFT Softw. Eng. Notes*, vol. 39, no. 5, pp. 1–3, Sep. 2014. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/2659118.2659129>