Edsger W. Dijkstra

Early Life

Edsger W. Dijkstra, third child of Douwe Wybe Dijkstra and Brechtje Cornelia Kluijver, was born in Rotterdam, Netherlands in 1930. His father was a chemistry teacher at a secondary school and was later promoted to the role of president of the Duch Chemical Society. His mother, also involved in academia, was a mathematician, however she never obtained a formal job. Dijkstra displayed a keen interest in the law sector during high school. He aspired to represent the Netherlands at United Nations conferences and felt a degree in law was a practical first step in achieving this. However, after his graduation in 1948 Dijkstra explored his options and decided to attend the University of Leiden majoring in Theoretical Physics. Dijkstra felt compelled to work in academia as a result of heavy influence from his parents.

While Dijkstra attended college, the concept of using an electronic computer was considered a novelty and it's popularity was gaining due to advanced discoveries made by Alan Turing numerous years earlier. He decided to attend a three week programming course in the University of Cambridge. Dijkstra commented on the experience saying "*It was a frightening experience: it was the first time that I left the Netherlands, the first time I ever had to understand people speaking English and immediately I was all by myself, trying to follow a course on a totally new topic. But I liked it very much.".* To his surprise, Dijkstra's supervisor also attended the programming course which eventually led to a job offer as a programmer in the Mathematics Centre.

Dawn of Software Engineering

Being one of the first software engineers, Dijkstra provided many nuanced theories about programming methodology and built the foundations for the development of software engineering enabling programmers to organize increasingly complex software projects.

In 1956, Dijkstra conceived the idea of a shortest path algorithm, and finally he published his work three years later with the algorithm being named Dijkstra's algorithm. The algorithm can be adapted to many forms however its main use is to find the shortest path between two nodes in a graph. This was a

groundbreaking discovery and its uses are not limited to software engineering but many other areas of technology also.
Millions of users around the world use Dijkstra's algorithm every day unconsciously while using applications and devices such as:

1. Google Maps: The digital mapping services provided by Google maps uses dijkstra's shortest path algorithm to calculate the best route from A to B which is provided by the user. However Google Maps has some limitations due to the shortest path algorithm. When travelling on roads with continuous bends eg. country roads, the shortest path algorithm calculates the route based on an unrealistic speed given the curvature of the road meaning the route and journey time are inconsistent.

2. Phones: The shortest path algorithm was a huge discovery for the telecommunications sector. It allowed telephone systems in a given city to be designed to an optimal rate. The desired route taken by a transmission is calculated via the shortest path algorithm with respect to the bandwidth being used.

Dijkstra's shortest path algorithm also led to the creation of OSPF (Open Shortest Path First). OSPF is a LSR (link-state routing) protocol that is used to find the best path between the source and the destination router using its own Shortest Path First.

Dijkstra was a member of the international team that was responsible for creating the computer language ALGOL-60. During this invention, Dijkstra introduced the notion of utilizing and handling recursion. This led to the creation of a new data structure called the stack. Stacks are still continuously being used by software engineers today. Speaking about the creation of ALGOL-60, dijkstra regarded the computer language as "*as the beginning of computing science; if we wish to mark a discontinuity in the way in which we thought about computing, then that is the emergence of ALGOL* 60*. ... it has made, for instance, the topic academically respectable.*" Dijkstra, along with the assistance of one of his colleagues, created the first compiler for the *ALGOL* 60 which was completed by August 1960.

In 1972, Dijkstra was awarded the ACM Turing award, the highest accolade achievable in computer science,

Program Methodology and Structure

In the 1950s - 1960s, programming was not considered an academic discipline. There was a lack of coherency surrounding the structure of programming due to no theoretical concepts or coding systems. While Dijkstra solely programmed in machine code, he concluded that high-level languages contained poor programming structure. Dijkstra subsequently released an article detailing a standard structure for programming. The beginning of the article contains a chapter titled “The inability to do much”. Here Dijkstra describes the issue of an increase in computer power accompanied by an increase in software complexity. This led to insufficient methods being used leading to a period known as the software crisis. He commented that *"The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem."*

Contributions to Concurrency

By introducing his 1965 seminal paper on the mutual exclusion problem, Dijkstra is believed to have spurred the study of concurrency. He wrote a paper (officially unpublished) on sequential process descriptions which solved the mutual exclusion problem. This was a crucial breakthrough in concurrency as it is the first known solution to the problem. He attributed the solution to the mathematician Theodorus Dekker with the algorithm eventually being known as Dekker's algorithm.

Concurrency has many uses in computer systems:

1. An increase in performance.
2. An increase in reliability.
3. Used as a control method in database management systems (DBMS).

Dijkstra's breakthrough in concurrency also reduces waiting\response time in a program. Stripe and Ebay are examples of companies today who utilize concurrency on a daily basis. The reduced waiting/response time in concurrency allows them to complete multiple transactions simultaneously. This wouldn't be possible without the foundation in concurrency Dijkstra provided us with and illustrates that many software engineering companies around the world employ his methodologies today.

Conclusion

After a long and decorated career, Dijkstra died in August 2002 in The Netherlands after a battle with cancer. In my opinion, Dijkstra is one of the influential software engineers in the world. He was a pioneer during the early days of software engineering and shaped the discipline of programming leading us away from the software crisis during the early days of software engineering.

Dijkstra's work can be seen across all sectors of computer science today. People not only study his work in secondary and third level education but utilize his algorithms when developing software throughout their careers. To conclude, without the work of Edsger W. Dijkstra, the complexity of software would be greatly reduced and our understanding of program design would be greatly hindered.

Sources:

https://en.wikipedia.org/wiki/Edsger_W._Dijkstra

https://mathshistory.st-andrews.ac.uk/Biographies/Dijkstra/

https://www.geeksforgeeks.org/applications-of-dijkstras-shortest-path-algorithm/

https://practice.geeksforgeeks.org/problems/what-are-the-advantages-of-concurrency