Why did computer science make a hero out of Turing?

Maarten Bullynck, Edgar G. Daylight, Liesbeth de Mol

[1]

[1] This is a version of the paper published as: M. Bullynck, E. Daylight and L. De Mol, Why did computer science make a hero out of Turing, vol. 58, 2015, pp.37–39.

Why Did Computer Science Make a Hero out of Turing?

Maarten Bullynck[†] Edgar G. Daylight[‡] Liesbeth De Mol[§]

November 14, 2016

Every discipline that comes of age consecrates its own roots in the process. In footnotes, anecdotes and names of departmental buildings, occasions are found to remember and celebrate personalities and ideas that a discipline considers its own. A discipline needs heroes to help create a narrative which legitimizes and fortifies its own identity. Such a narrative hardly reflects the complexity of historical reality. Rather, it echoes the set of preferences and programmatic choices of those in charge of a discipline at a given moment in a given place. Each name that gets integrated into an officialized genealogy is the result of discussions and negotiations, of politics and propaganda.

To the general public, the genealogies of physics and mathematics are probably more familiar than that of computer science. For physics we go from Galileo via Newton to Einstein. For mathematics we begin with Euclid and progress over Descartes, Leibniz, Euler and Gauss up to Hilbert. Computer science by contrast is a relatively young discipline. Nevertheless, it is already building its own narrative in which Alan Turing plays a central role.

In the past decennia, and especially during the 2012 centenary celebration of Turing, his life and legacy received an increasing amount of attention. Recently, the *CACM* has published two papers in which Turing's legacy is put into a more historical context [9, 6]. We continue this line of research by focusing on how Turing functioned as a hero within the formation of computer science. We will do so here by comparing the consecration of Turing with that of Gauss in mathematics.

Making Gauss a hero

In the early 19th century, the Prussian minister Wilhelm von Humboldt sought to introduce mathematics as a discipline per se in higher education. To do so, he needed an icon to represent German mathematics. He turned to the one German who had been praised in a report on the progress of mathematics to

[†]Université Paris 8, email:maarten.bullynck@univ-paris8.fr
[‡]Utrecht university, email:egdaylight@dijkstrascry.com
[§]CNRS, UMR 8163 STL, email:liesbeth.demol@univ-lille3.fr

emperor Napoleon: Carl Friedrich Gauss (1777–1855). Also, the new generation of mathematicians favoured a conceptual approach over computations and saw Gauss as the herald of this new style of mathematics. As such, Gauss became synonymous with German mathematics for both political as well as more internal reasons.

Towards the end of the 19th century, the prominent mathematician Felix Klein developed this Gauss image into a programmatic vision. From 1886 onwards, he had started to actively transform Göttingen's mathematics department into the world's foremost mathematical center. He promoted a close alliance between pure and applied mathematics and got cooperations with the industry on the way. On a national scale, he worked for the professionalization of mathematics education. To shape this disciplinary empire, Klein, too, used Gauss.

In his 1893 address to the first International Congress for Mathematics in Chicago, Klein talked about the latest developments in mathematics and spoke of [10]:

> a return to the general Gaussian programme [but] what was formerly begun by a single master-mind [...] we must now seek to accomplish by united efforts and cooperation.

The edition of Gauss' collected works (1869–1929) provided an abundance of historical material which Klein used to build an image of Gauss supporting his personal vision on mathematics. Klein portrayed Gauss as the lofty German who was able to pursue practical studies *because* of his theoretical research, a portrayal that, although very influential, was biased nonetheless.

In the 20th century, Klein's interpretation of Gauss was picked up by the international mathematical community and was modified accordingly. In the U.S., following Klein's 1893-address, Gauss' fertile combination of pure and applied struck a note for a mathematical community that often worked closely in alliance with industry [11]. In France, after World War II, the Bourbaki-group emphasized the abstraction of Gauss' work that transcended national boundaries and had helped pave the way for their structural approach to mathematics. However, in contrast with the Kleinian 'pure mathematician', Gauss was also 'rediscovered' after the birth of the digital computer as a great calculator and explorer of the mathematical discourse [4].

Making Turing a hero

Just like Gauss was instrumental to Humboldt and Klein to further the institutulization of mathematics, Turing played a similar role in the professionalization of the ACM in the 1960s. This goes back to the 50s, when some influential ACM actors, including John W. Carr III, Saul Gorn and Alan J. Perlis, wanted to connect their programming feats to modern logic. Stephen Kleene's *Introduction to Metamathematics* (1952) which contained a recast account of Turing's 1936 paper *On computable numbers* was an important source.

In 1954, Carr recommended programmers to deal with "*the generation of systems rather than the systems themselves*" and with "*the 'generation' of algorithms by other algorithms*", and hence with concepts akin to metamathematics [3, p.89]. Similarly, around 1955, Gorn became accustomed to viewing a universal Turing machine as a conceptual abstraction of the modern computer (see e.g., [8]). By the end of the 1950s, Carr and Gorn explicitly used Turing's universal machine to express the fundamental interchangeability of hardware and language implementations. Turing's 1936 theory thus helped ACM actors to articulate a theoretical framework that could accommodate for what programmers had been accomplishing independently of metamathematics [7].

In 1965, the vice President of the ACM, Anthony Oettinger (who had known Turing personally), and the rest of ACM's Program Committee proposed that an annual "National Lecture be called the Allen [sic] M. Turing Lecture" [1, p.5]. Lewis Clapp, the chairman of the ACM Awards Committee collected information on the award procedures "*in other professional societies*". In 1966 he wrote that

> [a]n awards program [...] would be a fitting activity for the Association as it enhances its own image as a professional society. [...] [I]t would serve to accentuate new software techniques and theoretical contributions. [...] The award itself might be named after one of the early great luminaries in the field (for example, "The Von Neuman [sic] Award" or "The Turing Award", etc.) [2].

ACM's first Turing Awardee in 1966 was Perlis, a well-established computer scientist, former president of the ACM, and close colleague of Carr and Gorn. Decorating Perlis is in hindsight thus rather unsurprising. Turing, by contrast, was not well known in computing at large, even though his 1936 universal machine had become a central concept for those few who wanted to give computer programming a theoretical impetus and also a professional status.[1]

The first wave of recognition that Turing received posthumously with the Turing award in 1966, is but a ripple when compared to the second wave. This started in the 1970s with the disclosure of some of Turing's war work for the Allies, followed by Andrew Hodges' authoritative 1983 biography, which also added a personal dimension to Turing's story, his life as a gay man in a homophobic world. This made Turing also known outside of computer science. The second wave culminated in the 2012 Turing centenary celebrations that nurtured the perception of Turing as the inventor of the modern computer and artificial intelligence. Some even claim that he also anticipated the internet and the iPhone.

The year 2012 was full of activities: there were over a 100 academic meetings, plaques, documentaries, exhibitions, performances, theatre shows and musical events. The celebrations also brought together a group of people with diverse

[1] We speculate that Turing was preferred over von Neumann, because the latter was associated with hardware engineering rather than with theoretical foundations of programming. Moreover, it might be that for the more liberally minded Carr, Gorn and Perlis, von Neumann was too strongly associated with conservative Cold War politics. There were other potential candidates as well, such as Emil Post. Historians are now starting to investigate these matters (see e.g. [6]).

backgrounds and promoted computer science to the general public, an achievement of which the longer-term impact has yet to be awaited [12]. A discipline has its heroes for good reasons.

As Hodges' biography shows, Turing's work was multifaceted. Not only did Turing contribute in 1936 to the foundations of mathematics, which later proved to be fundamental for theoretical computer science, he also worked at Bletchley park during World War II to help break the Enigma. He became an experienced programmer of the Ferranti Mark I for which he wrote a programmer's manual and even designed a computer, known as the ACE. He reflected on thinking machines and contributed to the field of morphogenesis.

It is therefore not surprising that for many today the multidisciplinary nature of computer science is personified in Turing who achieved all these different things in one short lifespan. Along these lines, Barry Cooper, the driving force behind the Turing centenary, said the following in 2012 (Quoted from [12]):

> The mission of [the Turing Centenary] was to address concerns about how science was fragmenting. We wanted to return to more joined-up thinking about computability and how it affects everyone's life. More generally, too, the Turing Year was important in highlighting the need for fundamental thinking.

From this perspective, Turing's theoretical work gives new impetus to the sciences as a whole, not just to computer science per se. The recent volume *Alan Turing – His Work and Impact* [5] — i.e., Turing's Collected Papers *cum* Essays from renowned scientists — also wants to bring this point home. It echoes even on the political level. The House of Commons has considered to name the new Technology and Innovation elite centers after Turing. According to the chairman of the Science and Technology Committee, "*There isn't a discipline in science that Turing has not had an impact upon.*" [?] As such, computer science, and especially theoretical computer science with its focus on computability, becomes the connecting discipline amongst the other sciences, and thereby turns into a fundamental science, not unlike mathematics.

The focus on computability and fundamental thinking is certainly not accidental. To a large extent the drive behind the Turing Year came from theoreticians. They do not ignore that Turing also worked in engineering. However, many of them argue that Turing must have invented the computer *because* of his theoretical 1936 paper. According to this view on science and technology, also present in Klein's interpretation of Gauss, theory precedes practice.

Looking backwards into the future

Over the past century, the one-dimensional image of Gauss has been replaced by a multitude of images. This shows that a discipline in constant evolution assesses its own identity through its heroes and allows for a *multiplicity* of readings. Certainly, each reading may further the agenda of a particular community, but the diversity of all images taken together, all grounded in some way in Gauss' legacy, positively stimulates the openness and generosity of a field.

Is Turing for computer science what Gauss is for mathematics? Computer science inherently involves both theory and practice. Turing embraced both sides in his work. However, if one celebrates Turing only *because* of his theoretical work, one runs the risk of increasing the already existing divide between theory and practice. Instead of favoring one reading of Turing and crowding out others, why not view Turing's own accomplishments as an invitation for a rapprochement between practice and theory? Integrating Turing's contributions into a more complex historical account may be to the benefit of computer science itself.

References

[1] ACM Council Meeting, 27 August 1965. Available from the "Saul Gorn Papers", the University of Pennsylvania Archives (unprocessed collection).

[2] ACM Council Meeting, 29 August 1966. Available from the "Saul Gorn Papers", the University of Pennsylvania Archives (unprocessed collection).

[3] J. Brown and J.W. Carr, III. Automatic programming and its development on the MIDAC. In *Symposium on Automatic Programming for Digital Computers*, pages 84–97, Washington D.C., May 1954. Office of Naval Research, Department of the Navy.

[4] Maarten Bullynck. Reading Gauss in the computer age: On the U.S. reception of Gauss's number theoretical work. *Archive for the History of the Exact Sciences*, 65(5):553–580, 2009.

[5] Barry S. Cooper and Jan van Leeuwen, editors. *Alan Turing – His Work and Impact*. Elsevier, 2013.

[6] Edgar Daylight. A Turing Tale. *To appear in: Communications of the ACM.*

[7] E.G. Daylight. Towards a Historical Notion of "Turing — the Father of Computer Science". *Accepted in the journal History and Philosophy of Logic.*

[8] S. Gorn. Real solutions of numerical equations by high speed machines. Technical Report 966, Ballistic Research Laboratories, October 1955. Available from the "Saul Gorn Papers" from the University of Pennsylvania Archives (unprocessed collection).

[9] Thomas Haigh. Actually, Turing did not invent the computer. *Communications of the ACM*, 57(1):46–51, 2014.

[10] Felix Klein. The present state of mathematics. In *Mathematical Papers read at the International Mathematical Congress, held in Connection with the World's Columbian Exposition Chicago*, pages 133–135, 1893.

[11] K.H. Parshall and D. E. Rowe, editors. *The Emergence ofthe American Mathematical Research Community, 1876–1900: J.J. Sylvester, Felix Klein, and E.H. Moore.* AMS, 1994.

[12] Sarah Underwood. The Alan Turing year leaves a rich legacy. *Communications of the ACM*, 56(10):24–25, 2013.