From Computing Machineries to Cloud Computing: The Minimal Levels of Abstraction of Inforgs through History

Extended abstract

The history of computing can be divided in two main periods: the ancient era and the modern era. Since ancient times, humankind succeeded to build methods and tools in order to help in calculation; in particular, in various parts of the world, in completely independent ways, different civilizations such as Roman and Chinese invented the abacus, which was still used by the Russian in 1957 for the necessary calculations to put Sputnik in space (Guedj 2005, 147). But the tool, i.e., the abacus, was not enough: various methods of representing numbers from 1 until 9,999 with the only help of the fingers were necessary to exploit the possibilities of the abacus, as reported by Leonardo Fibonacci da Pisa in his *Liber Abaci* (1202–1228)—'fingers' in Latin is *digita*, from which our use of 'digit' to indicate numerals derives (Boncompagni 1854).

Calculating machines were considered auxiliary tools for computation in the simple, non-abstract, sense—i.e., the user puts numbers in in order to have numbers out, and their meaning is in the eyes of the user himself—even when modern science and mathematics in parallel grew: Schickard's calculating clock (1623), as well as Pascal's *Pascaline* (1642) as well as Leibniz's Step Reckoner (1671–1673) are some notable examples. Their aim was to hide the calculation process to their users, as the idea of calculation as a tedious, low operation of the mind, good for slaves, not for men, as Napier first wrote in 1614 in publishing for the first time logarithms (Knuth 1973, 161) and Leibniz reprised at the end of the century (McCorduck 1979, 22). This divorce between intelligence and calculation, as put by Daston (1994), was also the philosophical basis of Babbage's Analytical Engine—his dream was the mechanical calculation and printing of all tables of ephemerides (Campbell-Kelly 1994).

The modern era of computing was born in 1936, when Church, Post and Turing put the foundations of general-purpose machines, while in 1941 Zuse built the first Turing-complete machine in the world (Rojas 1998). Unlike ancient times, modern computers were conceived to manipulate *symbols* in form of numbers: as Newell effectively recalled, "I've never used a computer to do any numerical processing in my life" (McCorduck 1979, 129). It is worth noticing that Turing (1950) still wrote explicitely 'computing machinery' in order to refer to machines, not human beings, when he proposed his famous test for Artificial Intelligence—term introduced at MIT by McCarthy in 1959 (Levy 1994). The idea behind modern comput-

ers is completely different from ancient times: calculation can represent—in digital form—intelligent behaviour or even mind *per se*. In other words, there is an *epistemological* level of abstraction in considering numbers as symbols, i.e., something that stands for something *aliquid stat pro aliquo*. In other words, the symbolization of numbers put in mechanical computation—which eventually constitutes software—is a collection of levels of abstraction (LoA), as "0's and 1's as such have no causal powers at all because they do not even exist except in the eyes of the beholder" (Searle 1990, 30). In fact, ontological forms of levellism—i.e., where LoA effectively exist, not only in the eyes of the observer—are hardly tenable if we analyse the generation of information after the Fourth Revolution (Floridi 2009, 37), especially if we do adopt a philosophical monism, i.e., that syntax is not intrinsic to physics (Searle 1990).

From the advent of general-purpose, Turing-complete machines, the relation between operators, programmers and users with computers, i.e., human-computer systems, or rather interconnected informational organisms or *inforgs*, in Floridi's terms, can be seen in terms of levels of abstraction (LoA), and henceforth analysed with the method of levels of abstraction (Floridi 2010, 2008).

In this paper an analysis of LoA throughout history of modern computing is proposed, in order to find the minimal number of LoA needed to explain the epistemology of inforgs—from early modern general-purpose operators of computing machineries until the final users of so-called 'cloud computing'.

This epistemological levellism uses Category Theory as the methodological reference, treating information as functions, i.e., a domain intensionally mapped into a codomain where the inner structure is preserved, instead of Cartesian products. Finally, a comparison with the method of LoA by Floridi (2008) is proposed, in order to find a categorial treatment of interconnected informational organisms.

References

Boncompagni, B. (1854), *Intorno ad alcune opere di Leonardo Pisano*, Tipografia delle belle arti, Roma.
URL: *http://gallica.bnf.fr/ark:/12148/bpt6k99398c*

Campbell-Kelly, M., ed. (1994), *Passages from the life of a philosopher: Work by Charles Babbage written c. 1812*, IEEE Press.

Daston, L. (1994), 'Enlightenment calculations', *Critical Inquiry* (21), 182–202.

Floridi, L. (2008), 'The method of levels of abstraction', *Minds Mach.* **18**, 303–329.

Floridi, L. (2009), Philosophical conception of information, *in* G. Sommaruga, ed., 'Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information', Springer, Frankfurt Am Main, pp. 13–151.

Floridi, L. (2010), *Information: A Very Short Introduction*, Oxford University Press, Oxford.

Guedj, D. (2005), *Numbers: the universal language*, Thames Hudson.

Knuth, D. E. (1973), *The art of Computer Programming: Seminumerical Algorithms*, Vol. 2, Addison Wesley, Reading (Mass.).

Levy, S., ed. (1994), *Hackers: Heroes of the Computer Revolution*, Anchor Press.

McCorduck, P. (1979), *Machines who think*, Freeman, San Francisco.

Rojas, R. (1998), 'How to make zuse's z3 a universal computer', *IEEE Annals of the History of Computing* **20**(3), 51–54.

Searle, J. R. (1990), 'Is the brain a digital computer?', *Proceedings and Addresses of the American Philosophical Association* **64**(3), 21–37.

Turing, A. M. (1950), 'Computing machinery and intelligence', *Mind* **59**, 433–460.