

Fabio A. Schreiber

The Cultural Roots of Computer Science

complete version of a letter published in

**Communications of the ACM – Forum
vol. 48, n. 11, Nov. 2005
pp. 11-12**

The Cultural Roots of Informatics

In the August 2000 *Viewpoint* column of CACM, Arora and Chazelle propose some remedy to an apparently paradoxical situation in which, while computers are pervading our everyday lives, Computer Science is facing a deep crisis in the U.S. [1]. They attribute it to “... *our collective failure as educators, researchers, and practitioners to articulate a cogent, compelling narrative about the science of computing (as opposed to just the technology)*” and blame the reduction of the field to programming and possibly its heterogeneous nature among the causes of the demotivation of young students in enrolling in Computer Sciences curricula and of a reduction in research funding.

This situation is not new and not peculiar to the U.S.. The large public - and the press articles are a testimony - tends on one side to look at computers as a mere mixture of technologies, obtained by wiring together quite a lot of circuits and writing silly (and mostly buggy) programs in some esoteric language, and on the other one to credit Informatics with the merit of inventing brand-new methodologies to solve any kind of problems, while the merit is often just to bring to the rationale and formalise some mechanisms which were already present in other disciplines, but in a latent, pragmatic form.

Therefore, while in [1] the author stress the benefits of integrating Computer Science with other disciplines (including the sociological fallout), I think that it should be also very important to go the other way round, i.e. to show by an epistemological approach how other disciplines lead to the development of a computer application. A first step toward this direction is constituted by R. Karp Turing Lecture [2]; the author, by examining the historical development of a part of Discrete Mathematics, shows how “Theoretical Informatics” results from a set of mosaic tesserae combining like in a puzzle.

I think that such an approach has even a further consequence. C.P. Snow brought to evidence, in a famous lecture [3], the divergence which developed between the humanistic and the scientific culture during the XIX and the XX centuries; since the end of the XVIII century, when the philosopher Kant and the mathematician Laplace joined their names in a cosmological theory, humanistic and scientific disciplines evolved in such a way that, while any cultured person knows who Shakespeare was and can mention a few of his works, very few humanists could even know the existence of the 2nd principle of Thermodynamics. Showing the cultural roots of Informatics, from Physics to Logic, from Linguistics to Mathematics, may implement the link between the two cultures and end the climate of diffidence and contempt that still exist among many of their scholars.

The attached figure is a modest contribution to showing how building a computer application requires knowledge coming from very different basic disciplines. Dotted lines represent a remote or indirect link toward topics of immediate interest to Computer Science; non boxed topics are chapters within each discipline which are relevant to a specific methodology or technology (boxed topics). Without further elaborating the rather intuitive notation, I want to comment on what this graph aims to be and what it is not: it should be a very coarse grain and incomplete representation of a network of more or less direct links among the disciplines contributing to the development of computer applications. One could immediately notice some remarkable omissions such as Graph Theory, which – as a branch of discrete mathematics - is pervasive of Computer Science, or Cryptography which is essential to modern distributed applications. Their links to the rest of the picture are so intertwined that I privileged clarity to completeness in this very high level representation which, in any case, should be further revised and refined. Certainly the picture *is not a workflow* of how to implement a computer application, *nor it is a semantic network* of concepts in the formal sense of this term.

I think that further work in this direction could be fruitful to attract brilliant young people to Computer Science; in the eighties, I used it with good results for introducing Computer Science and Engineering to teachers in Italian high schools (Liceo Classico).

Fabio A. Schreiber

REFERENCES

1. Arora, S. and Chazelle, B. Is the Thrill Gone? *Commun. ACM* 48, 8 (Aug. 2005).
2. Karp, R.M. Combinatorics, Complexity, and Randomness *Commun. ACM* 29, 2 (Feb. 1986).
3. Snow, C.P. *The Two Cultures and the Scientific Revolution* (1959).

