Borderlines

Giovanna Cepparello

1 April 1999

Abstract

In this paper we try to describe how Dynamic Logic made it possible a fruitful interplay between Informatics and Philosophy. It did so by acting as a common language, spoken by both computer scientists and analytic philosophers. In this way, Dynamic Logic offered the humanistic culture a new possibility of digesting and elaborating the informatic revolution, alternative with respect to the AI-based approach.

Contents

1 Idola specus

Human beings usually like to belong to a group, being it a race, a club or more vaguely the extension of an ideology. Paradoxically, even the absolute outsider, the one who is always against, will be probably following a precise cliché, a stereotype, entering implicitly a social group. At the same time, we also like to arrange other human beings inside groups, and to decipher the behaviour of our fellow creatures using social groups as a key of lecture. In this sense, groups represent a kind of *categorization* at a social level, necessary in order to organize informations and to elaborate opinions about our environment and about ourselves. It is easy and comfortable to say things like "Of course he is so hard-working: he is german!" or "I'm a real philosopher: I forgot again the cake in the oven!". The social reality is so complex that we probably *need* a filter in order to handle it[1]. Obviously, this kind of categorization will be very useful in order to attribute a *value* to people and habits around us.

Now, the reason why we have recalled the great importance of social groups for men and women, is the following: we think that the same mechanism is in some sense at work within the ambit of the academic world, where the borders among disciplines, departments and consequently people involved is often thick and proudly cultivated. Many intellectuals have a strong sense of membership to a given cultural community, and this has effects at different levels: it means to dress in a certain way, to read certain books and maybe to patronize certain cafés or restaurants. It means to share the same commonplaces about the colleagues of the nearby department (one of the finest pleasures of life!). But, most importantly, it can also mean to cultivate and to follow a precise professional training, or in other words, to teach and to learn an official know-how, on which the future research must be based[2]. This official know-how will probably set also the viable links with other disciplines or research areas (and, consequently, with other human varieties), while discouraging the connection with others. As I said, social groups do carry a value. Before Galileo, nobody thought of pointing the telescope to the sky, although good lenses were around in Europe since the XIII century. The reason was that opticians were considered as toy-makers, and science did not take them too seriously for a long time.

Needless to say, breaking the borders among disciplines can be a real achievement. It can raise new questions and open new perspectives. It means to dissolve the baconian *idola specus*, namely the phantoms of our hovel. The point we want to make is that Dynamics (namely Dynamic Logic plus its manifold applications to linguistics, philosophy, economics etc..), thanks to the work of some open-minded researchers, had the very respectable role of breaking some cultural borders. Let us see how.

[1]This point of view of kantian flavour has been theorized for instance in [1].

[2]This recalls the Kuhn theory about scientific theories, that are to be conceived holistically as universes of belief, and consequently are very well 'fortified' against what is outside them (cf. [5]).

2 Something about the birth of computer science

Computer science was born under the patronage of cybernetics. According to his founding father, Norbert Wiener, cybernetics is the discipline studying control and communication in machines and animals. It deals with the question: how a given message can modify the behaviour of a living thing or of a machine. Kubernetiké is in fact the art of the helmsman, of the one who elaborates and organizes commands. Concretely, cybernetics aims to the study and the mathematical characterization of biological processes, in order to build machines, automata. From this short definition a parallel already arises, which will become a kind of guide-metaphor of the computer culture: the intriguing parallel between machines and living organisms. At a first level, machines can emulate the behaviour of living things, and give the appropriate answers to given stimuli. At a deeper level, machines can even simulate the processes going on within living things (cf. for instance [2], where the authors try to export from the animal world to the machines the mechanisms of recognitions of forms).

The second short remark about the origins of computer science is the following. Modern computers were born for meeting practical needs. If clocks were almost necessary in the commercially active society of the Renaissance, so were calculators in the industrialized society of the end of the last century, and even more in the dawning global village of the beginning of our century. We cite two significant historical facts. The first relevant use of punched cards machines dates backs to 1890, when Herman Hollerith applied them to the american census. In our century, the golden age of computers starts in the forties, during the second world war, and this circumstance will affect in part the empirical nature of informatics: Wiener develops some central ideas about calculators while attempting to solve the problem of predicting the route in laying anti-aircraft weapons.

3 The empirical way to the thinking machines

We think that computer science has been in some way marked by these aspects we have seen, namely:

1. the parallel between machines and living things, coming from the cybernetic culture;
2. the empirical research program that acted as an incubator for the first computers;

Concerning 1), it is interesting to see that from the beginning informatics appears before the public opinion as something very much linked with robots and similar. But also at a more serious level, the big challenge that informatics has issued to the culture of (part of) our century concerns the possibility of building an intelligent machine[3]. Point 2) has a lot to do with what we have said about

[3]Notice that while Artificial Intelligence was growing up, the couple machine-program turned out not to be so adequate in order to interpret the most complex phenomena of life and intentionality. The idea of a programmable machine is not able to account for the

idola specus: computers were born in a certain human environment, which has set the range of the admissible intellectual elaborations.

¿From the combination of 1) and 2) we can draw a simple conclusion. Computer science has been aimed for a long time to the designing of machines more and more quick and competent, more 'intelligent' in a broad sense. In this dominant perspective, programs and programming languages have been conceived as instruments, totally functional to their task. Programs as instruments do not need to be based on a solid theory, they only need to work; and in order to check that they do work properly, it is enough to test them and kill the bugs.

4 The logical foundations of programming

Around the end of the sixties, a movement arises within the international informatics community that aims to provide programming with a logical foundation. Computers have already entered offices, factories, and their use is spreading rapidly. The *informatization* of society has started. In front of this growing importance of computers, some research groups raise the question of checking in some way the theoretical validity of programming. In particular, given a computer program π, they try to see if it is possible to prove that π satisfies the intentions of the programmer, namely that π is correct. The idea comes up to build formal languages that can talk about programs, and to create formal systems (axioms and rules) to prove facts about them. The importance of this question is not only theoretical: the consequences of a wrong program can be incredibly big. Quoting [3]:

> ...the cost of error in certain types of program may be almost incalculable - a lost spacecraft, a collapsed building, a crashed aeroplane, or a world war.

Given that, the idea of *founding* computer programs, proving their metaproperties, immediately captures the interest of many researchers. Therefore, the seminal paper [3] will become one of the most quoted papers in the history of computer science.

Easy to guess, this great research project started some 30 years ago represents from our point of view a first important step for breaking the original borders of computer science. It brings computer science very closed to the mitteleuropean mathematical tradition: the Hoare program is quite similar to the Hilbert program! Here is again an important - and explicit - quotation from [3]:

> In this paper an attempt is made to explore the logical foundations of computer programming by use of techniques which were first applied in the study of geometry and have later been extended to other branches of mathematics. This involves the elucidation of sets of

organization inside a living thing; DNA, for instance, cannot be explained without referring to the enzymatic proteins that rule its transcription and translation, but these are produced by the action of DNA itself.

> axioms and rules of inference which can be used in proofs of the properties of computer programs.

Accordingly, the Hoare program has had a vast cultural significance. It has transformed programming from a kind of handicraft into a science, into a branch of mathematics. The big moral of the story is that computer programs can be studied at a metalevel, as pure mathematical objects. They cease to be mere instruments to reach a task, or defective imitations of human inferential processes. Rather, they acquire an autonomous status, and they deserve the attention of intellectuals outside the informatic community.

5 The next step: Dynamic Logic

A few years after the publication of [3] a very nice truth comes to the light: that Hoare calculus can be seen as a multimodal logic. Here is a quotation I like ([4]):

> In the spring of 1974 I was teaching a class on the semantics and axiomatics of programming languages. At the suggestion of one of the students, R. Moore, I considered applying modal logic to a formal treatment of a construct due to C.A.R. Hoare, $p\{a\}q$, which expresses the notion that if p holds before executing program a then q holds afterwards. Although I was sceptical at first, a weekend with Hughes and Cresswell convinced me that a most harmonious union between modal logic and programs was possible.

In other words, the formalism of intensional logic turns out to be ideal in order to prove metaproperties of computer programs. And here the border really collapses: computer science shows a deep, essential link with that part of logic that most happily marries with analytic philosophy and with classical western philosophy. It is again the leibnizean possible worlds that we are talking about!

Hoare Calculus and Dynamic Logic opened a fruitful, interesting way from computer science to a certain area of humanities. They actually offered the humanistic culture a new possibility of digesting and elaborating the informatic revolution, alternative with respect to the Artificial Intelligence-based approach. In a slogan, AI promotes the idea that intelligent phenomena can be mimicked and reproduced within machines. On the other hand, the philosophical message behind the dynamic way is that the formalisms used for programming machines can be employed in their mathematical pureness as keys for interpreting intelligent phenomena. Quoting [6], here is the import of this research program:

> In recent years, there has been a growing interest in the logical structure of cognitive actions underlying human reasoning or natural language understanding. Traditional logic and philosophy have been mainly concerned with the products of these actions, such as thoughts, proofs, propositions. But in various disciplines, including philosophy, computer science and linguistics, the mechanisms of information flow themselves are becoming primary objects of study.

> This interest reflects a broader cultural influence of computational paradigms ...

We think that this 'cultural influence of computational paradigms' on the philosophical tradition has been made possible by the discovery, so to speak, of a common language, spoken at the same time by informatics and analytic philosophy, and this common language is Dynamic Logic. Dynamic Logic made it possible an intriguing cultural contamination. The dynamic international community is harvesting the appealing fruits...

References

[1] H. Tajfel, C. Flament, M. Billing, R.P. Bundy (1971), *Social Categorization and Intergroup Behaviour*, in: European Journal of Social Psychology, 1.

[2] H.R. Maturana, J.Y. Lettvin, W.S. McCulloch, W.H. Pitts (1960), *Anatomy and physiology of vision in the frog*, in: Journal of General Physiology, 43.

[3] C.A.R. Hoare (1969), *An Axiomatic Basis for Computer Programming*, in: Communications of the ACM, 12.

[4] V.R. Pratt (1980), *Application of Modal Logic to Programming*, in: Studia Logica, 39.

[5] T.S. Kuhn (1962), *The structure of Scientific Revolutions*, The University of Chicago.

[6] J. van Benthem (1996), *Exploring Logical Dynamics*, CSLI Publications and folli.