What a course on philosophy of computing is not*

Vincent C. Müller

American College of Thessaloniki

1. Learning goals

This programmatic paper is trying to contribute to the development of an international course in the philosophy of computing, the main outlines of which were discussed at NA-CAP 2007 in Chicago (and at earlier CAP meetings). The chair of the 2007 panel, Piotr Bołtuć, invited the panelists saying: "The aim is to try to define some standards of what a good course in this would look like", so I will try to contribute to this aim.

A good course should include the interesting issues and anyone in the field can name many such issues, but why this thematic and pedagogical *unity*? Is there anything that holds the course together? This seems the crucial question. It appears that the course cannot be called just "Computing *and* Philosophy" as the CAP conference series and the associations because this title is chosen for its all-inclusiveness of "something to do with both computing and philosophy". Inclusiveness is useful for academic organizations and conferences, but in a course it needs to be avoided. We cannot present a ragbag of interesting issues, but neither can we go around and claim ownership of some established problems, perhaps after giving some of them new names. To put it positively, we must formulate the core of the course, the central learning goals, and then we can see what fits and what does not.

Such learning goals are not already specified by saying that the course is "about the philosophy of computing", not just because this is too vague, but primarily because it does not specify any philosophical problem. The idea that one can take any object, event or phenomenon x and then do a "philosophy of x" is one that gives philosophy a bad name. There is a good reason why there is a philosophy of physics, but hardly one of chemistry, why there is a philosophy of

* *APA Newsletter on Philosophy and Computers*, 8 (1/Fall), 36-38. http://www.apa.udel.edu/apa/publications/newsletters/computers.html

history, but hardly one of archeology. The former have interesting and basic philosophical problems, problems that are urgent for the practitioners of these disciplines, the latter do not. If we want to do and teach a philosophy of computing, we need to specify such basic problems that are intellectually challenging and practically relevant.

In the following, I shall take a look at some possible candidates that may provide unity – and essentially reject them all for being both too wide and too narrow. This should result in some controversy and thus in steps forward. The negative results also allow an indication where the positive result should be located.

2. Not philosophy of computer science

The obvious candidate for identification is the fairly well established field of philosophy of computer science – which depends on the large and wealthy mother discipline. This candidate is too narrow a specification for the course, however, because it is limited to computation in artifacts, in "computers". This ignores a purely formal theory of computing as well as the important possibility of computing in natural processes. It is now common to think of the human mind as operating mechanically, even as operating as a computational mechanism; the view called "computationalism" which was the original myth of the cognitive sciences. (The history of which can now be read in Margaret Boden's opus magnum (Boden 2006); see also my review (Müller 2008).) What is more, there are countless other natural (and even social) mechanisms that can usefully be regarded as computational.

The candidate is also too wide because computer science includes a few problems that are irrelevant to this course, namely those to do with technical problems of the hardware – problems in electrical engineering, robotics, etc.

Finally, it is far from clear that computer science really has interesting basic problems that are in need for philosophical analysis. In fact, "computer science" is a misnomer since the aim of the discipline is to *make* things, not to *find out* about the world. It is thus a proud member of the *engineering* disciplines, not of the sciences. This detail is often overlooked by all-encompassing representatives of the discipline, like Herb Simon, one of its founding fathers who used the characteristic title "Artificial Intelligence: an Empirical Science" (Simon 1995), arguing that the aim of AI is to find out what intelligence is.

Despite this rejection, I think we should keep in mind a few questions of this area that should feature in a course on Philosophy of Computing, for example: What is software? What is a computer program?

3. Not theoretical computer science

There is a theoretical area that may have been missed in the previous section: the purely theoretical study of computational structures and processes, often considered to be a branch of mathematics.

It should be obvious that this candidate is too narrow, given that there is more to computing than formal structure. There is also a certain danger that it may be too wide, in its discussion of some purely formal subjects and methods that lead in to the wider areas of logic and mathematics.

Having said that, this area is clearly rich in subjects that should feature in a course like ours, questions related to the formal description of computing and computing machines, the bounds of computability, the resources needed (complexity theory), program verification, data structures, etc.

4. Not philosophy of information

One other candidate with reasonably serious aspirations of complete coverage is the philosophy of information – after all, what is called "computer science" in English is often called the equivalent of "informatics" in other languages.

Whether this proposal is too narrow is actually a difficult question because it would presume that we can say whether all computing deals with information. I tend to think that this is not the case, since some computing is purely syntactic; but clearly this is an open question. At any rate, we should not pretend that it has been answered – which is precisely what we would do if we were to limit our subject to the philosophy of information.

The candidate is clearly too wide since not all information is digital, but even if we include non-digital information, not all information is processed computationally, or processed at all, for that matter.

Again, this area is a rich source of relevant questions: Does computation compute over information or just over data? What is information and what is its dynamics? Is intelligence information processing, or could it be? Is nature somehow informational? – Of course, a rich source of programmatic material on these questions is (Floridi 2004).

5. Not philosophy of AI

The philosophy of artificial intelligence is clearly a rich source of basic philosophical problems, but equally clearly it is narrower in scope than our course: there is more to computing than intelligence. This area is also too wide because many of its problems are not specific to computational intelligence, particularly those traditionally discussed in the philosophy of mind and cognition.

The problems that we should take into account from this candidate are the classical ones concerning cognition, perception and intelligence in computational systems. Particularly important are, of course, the problems of representation or meaning in computers and perhaps the more recent questions whether an embodiment, a will and emotions are necessary for intelligence in computers.

6. Not philosophy of technology

One area that has been somewhat on the sidelines of philosophical debates within the strongly analytic traditions of the philosophy of computing is the embedding of computation in the wider context of human technologies. As in the philosophy of computer science, however, this area is too narrow, since it ignores computational systems that are not artifacts (e.g. formal, natural, social).

That this candidate is too wide is evident, since it covers any kind of human technology, not just the computational ones. (And if we shall discover that everything is computing, the philosophy of computing would not become all of philosophy.)

This area does, however, provide us with a very important set of questions, concerning human life with computers, especially human societies and computers. It also reminds us that there is the important issue of human ethical behavior with computers, what is known as "computing ethics." (The idea of ethics *for* computers or other technical artifacts is just a misunderstanding, in my opinion.)

7. Conclusion: The Core Curriculum

We have seen that each of the above proposals was both too narrow and too wide. In a sense, this is good news, since it shows that the philosophy of computing is neither a sub-division of a traditional field nor a superset of already existing fields. It does thus stand a chance to carve out a field of its own. It seems essentially situated within the philosophies of information and technology, while

incorporating most of the philosophy of computer science and of AI; plus an appendix of applied ethics.

On a positive note, I think that what we have seen so far can be captured fairly concisely to be about theoretical questions on the nature of computing, more practical questions of what that kind of mechanism is capable of and finally the ethical question. In his model course, Bill Rapaport suggested that "… an excellent course on the philosophy of computer science could consist solely of close readings of Turing's two major essays: his 1936 paper on computability and his 1950 paper on whether computers can think" (Rapaport 2005) I would agree, just that we would need to add a paper on ethics, something like Jim Moor's "What is Computer Ethics?" (Moor 1985).

Immanuel Kant famously defined philosophy to be about three questions: "What can I know? What should I do? What can I hope for?" (KrV, B833). I want to suggest that the three questions of our course on the philosophy of computing are: What is computing? What should we do with computing? What could computing do?

8. References

Boden, M. A. *Mind as Machine: A History of Cognitive Science*. 2 vols. Oxford: Oxford University Press. 2006.

Floridi, L. Open Problems in the Philosophy of Information. *Metaphilosophy* 35/4 (2004): 554-582.

Moor, J. What Is Computer Ethics? *Metaphilosophy* 16/4 (1985): 266-275.

Müller, V. C. Review of Margaret Boden 'Mind as Machine' (2 vols., Oxford University Press 2006). *Minds and Machines*, 18/1 (2008): 121-125.

Rapaport, W. *Philosophy of Computer Science: An Introductory Course* 2005 [accessed 12.3.2007 2007]. Available from http://www.cse.buffalo.edu/~rapaport/

Simon, H. Artificial Intelligence: An Empirical Science. *Artificial Intelligence* 77 (1995): 95-127.