



Paweł Polak

The Pontifical University of John Paul II, Krakow
Copernicus Center for Interdisciplinary Studies, Krakow

COMPUTING AS EMPIRICAL SCIENCE – EVOLUTION OF A CONCEPT¹

Abstract. This article presents the evolution of philosophical and methodological considerations concerning empiricism in computer/computing science. In this study, we trace the most important current events in the history of reflection on computing. The forerunners of Artificial Intelligence H.A. Simon and A. Newell in their paper *Computer Science As Empirical Inquiry* (1975) started these considerations. Later the concept of empirical computer science was developed by S.S. Shapiro, P. Wegner, A.H. Eden and P.J. Denning. They showed various empirical aspects of computing. This led to a view of the science of computing (or science of information processing) – the science of general scope. Some interesting contemporary ways towards a generalized perspective on computations were also shown (e.g. natural computing).

Keywords: empirical computing science, natural computing, philosophy of science, methodology of computer science.

1. Introduction

While looking for computing's place in today's world, it seems natural to ask the question whether this science could be a candidate for provider of the fundamental framework of knowledge guiding philosophical intuitions and assumptions. In other words, is it possible that computer science may play the role of primary world-view when forming knowledge in 21st century society? Interestingly, this role may be somewhat further clarified by reflections on the methods of computing, a field (known also as empirical computing) that has been under development for several decades and one that is only recently gaining wider recognition.

It would seem that, today, computer science is one of the fully formed disciplines; thus it should have a clearly crystallized methodology. However, despite decades of its dynamic development, the end to discussion about the methodological foundations of computing is nowhere in sight. Reflection on the methodological aspects of computer science could therefore provide

a beneficial perspective on understanding its evolution and role in modern society.

In this paper, computing is regarded, following the scientific praxis, as a field that is methodologically and epistemologically heterogenous.² Computing was to be the science not only of computability and algorithms but also concerned with the broad understanding of computational machinery and its construction. The basis of the classical understanding of this science has become three fundamental concepts: the concept of the Universal Turing Machine (UTM); the concept of the algorithm; and the stored-program paradigm (see Tedre, 2011).³ With the development of the research, it has become evident that such a standard approach requires extensions and changes.

This study will consider the methodology employed in the wider understanding of computing. Quite often, computing is perceived narrowly as the science of algorithms (e.g. Harel, 1987), which reduces this field only to the role of mathematics, and methodologically situates it partially in pure mathematics and partially in applied forms. Such a reduction, quoted here as an expression of certain extremes in thinking about computing, does not reflect fully its specificity (Knuth, 1974). The thesis about the existence of empirical aspects of computer technology (included here collectively under the rubric of empirical computing) is not entirely new, though it will aim to better reflect the importance of innovative concepts emerging from this science. These novel ideas merit special interest because many researchers indicate that methods of computing approach those of the natural sciences and their philosophy.

In this study, we trace the most important current events in the history of reflection on computing. We will start by showing how a research program aimed at creating artificial intelligence contributed to the understanding of computing as an empirical discipline. Then we will show the evolving debate around the empirical nature of this science that is attributable mainly to the past two decades. We will focus on methodological issues and set aside, due to the limitations of this work, some of its important practical aspects.⁴ At the end, we will reflect on the impact of the transformations discussed on the generalization of the concept of computing.

2. Forerunners of Artificial Intelligence about empiricism (H. A. Simon & A. Newell)

While searching for the sources of views on certain aspects of the empirical nature of computing, it is necessary to refer to the research program on

artificial intelligence (AI). This multi-threaded field of research is derived from 20th century cybernetics, and today uses the name that was coined only in 1956. Since the 1960s, AI research has been developing mostly as a part of computer science; thus it has been naturally treated as a part of this discipline. The very concept of artificial intelligence is ambiguous and vague because it is based on variously understood concepts of intelligence or intelligent behavior that are meant to describe systems implemented by artificial means. Despite these conceptual problems, several technologies and systems have been developed that perform certain behaviors considered to be intelligent.

Following Mariusz Flasinski, we can enumerate the main areas of this type of behavior: perception; pattern recognition; knowledge representation; learning; problem solving; deduction; decision making; planning; natural language processing; manipulation and locomotion; social and emotional intelligence; and creativity.⁵

The extreme position on the empirical nature of computing was presented in the mid-1970s by two prominent representatives of artificial intelligence programs: Allen Newell and Herbert Simon.⁶ They created the first working AI programs, Logic Theorist (1956) and General Problem Solver (1957). It is worth recalling that these programs have had a significant impact on the development of AI; the first of them carried automatic proofs of 38 out of 52 theorems in Russell and Whitehead's *Principia Mathematica* (i.e. the proof of one of the theorems was regarded as more elegant than the original), which significantly strengthened the faith of scientists in the possibility of the success of artificial intelligence. Simon and Newell were acknowledged not only as precursors of practical solutions, but also established the view that intelligence (i.e. the ability to solve problems) can be realized with the help of manipulating symbols, e.g. by means of algorithms and computations. In this perspective, intelligence is understood as symbol manipulation that would be completely independent from the implementation or the computing system. As a recognition of their achievements, in 1975, they were honored with the prestigious ACM Turing Award. During the award ceremony, they gave a memorable lecture in which they formulated the program of the symbolic AI. For us, of special interest is the context of this work, because the authors treated computing as an empirical science, as was evident in the title of their joint publication 'Computer Science as Empirical Inquiry' (Newell and Simon, 1976).

Distinctive and interesting in itself is the opening statement defining the subject of the research: 'Computer science is the study of the phenomena surrounding computers' (Newell and Simon, 1976, p. 113). The authors

proposed to look at computers as objects of the empirical world interacting with other empirical objects. Computing activities can be studied empirically, similarly to the interaction of computers with other objects of the real (empirical) world. According to the universal interpretation of the artificial intelligence research program, research in computing would be simply a variation of empirical science. The computer in this sense is not just a set of hardware and software; rather, along with the software the computer was treated as a living organism.⁷ It is worth noting that to describe the computer as a living organism in this way is not just a linguistic mannerism: computing, in this sense, would be methodologically approaching biology and other natural sciences. Clearly, this involves Simon and Newell's concept of strong artificial intelligence (strong AI).

2.1. Specificity of observation and experiment in computing

The methodology of computing, by Simon and Newell, is based on the concepts of observation and experimentation, but the understanding of it differs from the narrow conception of empiricism associated with physics. The concepts of observation and experimentation are here based on widely understood empiricism, as is the case of sciences such as astronomy, economy, or geology.

Actually constructing the machine poses a question to nature; and we listen for the answer by observing the machine in operation and analyzing it by all analytic and measurement means available. Each new program that is built is an experiment. It poses a question to nature, and its behavior offers clues to an answer. (Newell and Simon, 1976, p. 114)

The specificity of experimentation in computing lies in the fact that the same object (a computer program) can usually be examined by both empirical and analytical methods. A single experiment, then, would yield much more information than an experiment in other disciplines. Therefore, the analysis of computer objects only using formal (mathematical) methods will not provide a conclusive answer about whether the program actually works as expected. However, this claim is arguable. In many cases, the formal methods are sufficient to prove the correctness of a program. Moreover, the formal analysis of the computer software is even a subject of the academic science curriculum. In other words, you can entertain some doubts as to the validity of the presented approach.

The problem will become clearer when we look at very complex programs, especially those that include artificial intelligence; here, the formal means of analysis cannot provide a conclusive answer to the question of

whether a running program meets the requirements of intelligent behavior. We do not have, in fact, properly adequate formal recognition of what intelligence is. Moreover, the very purpose of the program seems to be fully elusive for formal methods, while empirically we can easily discern intelligent behavior (of course, this does not mean that there is unanimity on what is and what is not intelligent behavior). The precursor of understanding the need to apply empirical tests in evaluating intelligence in artificial systems started with Alan Turing. His famous test was based precisely on the operating procedure, so it had a strictly empirical nature.

According to Simon and Newell, an intelligent system can only be built using trial and error heuristics: it is necessary to create empirical theories about the necessary and required proprieties of such a system, and then to verify them in the specific case by constructing such a system and testing its properties. Thus, Simon and Newell are following, in principle, the Turing view on intelligent behavior in artificial systems. However, they draw far-reaching consequences from this position.

The main issue involved in how to understand an hypothesis or theory in the context of computing science remains. Simon and Newell claimed that in computing we deal with qualitative theories; they write about ‘Laws of qualitative structure’. They gave some examples of what they believe a qualitative theory would be. For them, the concept of a cell in biology is a case in point. The concept of a cell, which became the foundation of modern biology, is based on observation. Cells are described qualitatively, which is sufficient for formulating a variety of biological predictions about the functioning of a cell. The second example is the theory of plate tectonics in geology. The theory may be verified by observations; it can be used to draw interesting conclusions regarding the future and, to some extent, recreate the geological past; but it functions on the level of a qualitative description of geological phenomena. Similarly, several breakthrough ideas such as Pasteur’s germ theory or Dalton’s theory of the atomic structure of matter were all of a qualitative nature. Pasteur and Dalton always claimed that qualitative theories played a major role in the development of science, and many such breakthroughs in science have this type of character.

2.2. Hypotheses in computing: AI context

One may guess that qualitative theories were supposed to play the same role in computing as mentioned in earlier examples of qualitative laws. Such theories are present in all areas of computing, not only in AI, as mentioned by Simon and Newell. In this area, they pointed out two interesting empirical hypotheses: The Physical Symbol System Hypothesis and The Heuristic

Search Hypothesis. Let us look at the first one, as it had a profound impact on AI development.

Simon and Newell's Physical Symbol System Hypothesis is known primarily as the first formulation of an AI research program. It was formulated as follows: 'A physical symbol system has the necessary and sufficient means for general intelligent action' (Newell and Simon, 1976, p. 116). We are interested here only in the empirical nature of this hypothesis. The authors defined a certain class of systems (i.e. physical symbol systems)⁸ and posed the question whether such systems adequately represent some real processes. They assumed that this hypothesis may be true or false, but its verity should be decided not by philosophical argument but, rather, empirically.⁹ Such a hypothesis justifies the search for more and more refined information processing systems, and the authors tried to demonstrate that the history of computing is, in essence, a continuing search for the practical solutions that would confirm this hypothesis. The hypothesis would then play an important heuristic role, by defining the objectives to be achieved, and thus determining computing science's directions of development.

Let us redefine this thesis further in the following way: 'There is a physical symbol system that poses the sufficient and necessary means to realize general intelligence behavior.' Reasoning theoretically, only one observation is required to conclusively confirm (or falsify) the modified thesis. But such a position is methodologically naive (for both versions), because it does not take into account that it is impossible to obtain a conclusive confirmation or conclusive falsification of this thesis; the results of observation may always be reinterpreted.¹⁰

Artificial Intelligence systems would thus include questions posed to the reality, but the answers themselves would not be the most important. More importantly, the empirical hypothesis would be 'a source of ideas that would go into the construction of programs' (Newell and Simon, 1976, p. 119). Such hypotheses constitute, then, an important part of computing practices, even if they fall short of an epistemological ideal:

They have more flavor of geology or evolutionary biology than the flavor of theoretical physics. They are sufficiently strong to enable us today to design and build moderately intelligent systems for a considerable range of task domains of how human intelligence works in many situations. (Newell and Simon, 1976, p. 126)

If we perceive a radical Simon and Newell position as the reduction of computing to an empirical discipline, it is certainly an oversimplification: several research areas in computing, such as algorithmic theory,¹¹ are

very close to mathematics. But, if we assume the slightly moderate position that research in computing is based on some empirical qualitative principles (laws), it seems that this position is an interesting prelude to further consideration of the specifics of methodology in computing.

3. Computing as a natural science according to Stuart S. Shapiro

A more far-reaching approach to computing is presented by S. S. Shapiro in his work ‘Computer Sciences: The Study of Procedures’ (Shapiro, 2001). The central role in computing, in Shapiro’s view, is played by the procedure defined by Webster’s dictionary as ‘a particular way of doing or going about the accomplishment of something’. This concept includes both the concept of the algorithm and also the typical computing science objects (such as operating systems) that do not satisfy the strict definition of an algorithm (that the algorithm must be a terminating process). Shapiro argues that science dealing with procedures is a natural science because of its subject. This is justified as follows:

‘Procedures are not natural objects, but I claim that they are natural phenomena that may be, and are, objectively measured [...]’ (Shapiro, 2001).

A vital distinction here is between the object and a phenomenon. It is true that the object is an artefact, deliberately constructed by man, but its operations are based on natural processes, so it can be investigated like any other natural phenomena.

Such a perspective only narrows the research prospects, which is the main argument against Shapiro’s proposal. However, the author is not very clear about the scope of an empirical approach, which may be interpreted as an expression of caution. It is worth noting that Shapiro’s position presents significant advantages: it takes into account that if there are no natural processes behind computations, it would be impossible to create computer systems. Thus, it becomes evident that information systems fall within the typical causal relations with other natural objects, so these aspects can be tested using the methodology specific to natural science.

Shapiro’s concepts suggest that the boundary between computer science and the natural sphere, which is the domain of empirical research, is not as sharp as it might seem at first glance. Introducing computational processes similar to those in nature into computer science opened up a new perspective that allows one to extend the scope of computer science in the natural

world. Along with Simon and Newell's proposals, we are dealing here with interesting attempts to broaden the study of the nature of computing. These views, breaking away from mainstream thinking about computational sciences, paved the way for later discussion about the fundamental role of this discipline.

4. Computing paradigms of Peter Wegner and Ammon H. Eden

Topics related to the empirical nature of science also appeared in the context of the analysis of paradigms in the field. Results of this analysis pointed to the new ways computing could be generalized. Here we look at two more significant attempts at this type of analysis.

4.1. Peter Wegner's four paradigms (cultures) of computing

Peter Wegner presented these paradigms of computational methodology around the same time as Simon and Newell. In 1976, by analyzing the various definitions of computing, he recognized four computing paradigms and inscribed them in their historical development (Wegner, 1976).¹²

The empirical paradigm, which was inspired by the work of Simon and Newell, was, according to Wegner, historically the first paradigm of computing, and dominated computer science in the 1950s. This period is called 'data collection'. The name suggests associations with the concept of inductive data collection, aimed at creating a basis for the construction of the future theory. This interpretation seems to be correct because, in Wegner's view, empirical research had only a preliminary, preparatory character:

It is natural for disciplines to evolve from an initial empirical phase through a mathematical phase to a 'practical' engineering-oriented phase. (Wegner, 1976, p. 323)

So here we have a simplified outline of the development of research methodology: from data collection, through construction of mathematical theory, and, ultimately, ending with practical applications of that theory. Wegner saw, however, room for empiricism not only in the preliminary stage. In further stages of the development of computing, empirical methods would be, according to him, adequate for the tasks associated with the assessment and efficiency of programs (see Wegner, 1976, p. 324). Besides, empirical methods would be appropriate for testing highly complex programs, whose structure is too complicated for formal analysis. Such programs, according to Wegner, can be studied only empirically. It is worth noting that from

a methodological point of view, this proposal did not bring new and interesting elements; it is based on a rather simplistic and stereotypical image of science. However, the mere idea of applying the concept of paradigm to the analysis of computing was to prove fruitful years later.

4.2. Ammon Eden's three paradigms

The idea of using paradigms in computing was broadened and deepened by Ammon H. Eden (Eden, 2007). However, he took a different view from Wegner of the number of paradigms and, the point of interest to us, of the problem of empiricism. This author also showed precisely the philosophical basis of distinguished paradigms.

It should be noted that Wegner and Eden used Kuhn's concept of a paradigm. They wanted to show that accepting a collection of fundamental concepts determine that various conceptualizations concerning the scope of computer science and its methodology can coexist. This was not very insightful, because it does not take into account even the criticism of the concept of a paradigm, which was carried out by Kuhn. However, this concept worked quite effectively as an intellectual tool, ordering existing intuitions with regard as to what may be considered computer science.

In some aspects, Eden's research can be treated as an attempt to develop and clarify Simon and Newell's ideas. Firstly, Eden singled out the rational paradigm (Eden, 2007, p. 136n), which is shared widely by those theorists of computer science who treat it as a part of mathematics. From the point of view of epistemology, what is sought by means of deductive methods is a priori knowledge about 'correctness'. The research objects, for scientists sharing this paradigm, are mathematical objects.

Secondly, Eden differentiated the technocratic paradigm, shared by software engineers, and the computing programs in the paradigm are treated as data and are investigated with empirical methods.

Thirdly, he distinguished the scientific paradigm shared by AI researchers, who perceive programs as the analogues to mental processes (or their equivalent). Researchers sharing this paradigm seek a priori as well as a posteriori knowledge by combining the use of formal deductive and empirical methods.

As we can see, the empirical threads appear immediately in the two last paradigms. We are specifically interested in the scientific paradigm, as one whose horizons extend further than the prospect of effective technical applications. The main issue is epistemological and was recognized by A. H. Eden in the following terms: Does the knowledge about programs constitute priori or posteriori knowledge, or does it come from pure reason

or from experience? In a case when we have formal program specifications, the analytic methods seem sufficient, while, for informal specifications, the analytic method proves insufficient. Moreover, although in principle these programs could be subjected to theoretical analysis, because of the complexity, in practice, the properties of these programs can only be learned through observation and experimentation.

4.3. Computing as natural science

According to Eden, in the scientific paradigm the researcher is faced with the complexity of programs¹³ and, sometimes, with their chaotic behavior these features make them resemble some natural objects described and studied by scientific methods. Thus, it appears that some scientific testing methods should be effective in the case of computer science. Some researchers doubt whether natural objects can be treated on a par with artefacts (as computers and programs are). Artefacts, however, interact with reality in the same way as natural objects do; this allows them to be considered a part of the same reality, and their interactions with other elements within this reality can be studied.

Eden notes that the notion of a scientific experiment must be clearly distinguished from the reliability tests in the technocratic paradigm: although the two concepts are connected with empirical research, the objective of the first is to carry out empirical hypothesis testing, while the second is intended to check whether the program meets the requirements set by users. Although the first approach investigates the general dependencies, rather than only the fulfillment of the specific application requirements, it is precisely the existence of the explicit testable hypothesis that decides the scientific character of this empirical method. It is worth mentioning that the concept of the hypothetico-deductive method is a scientific method Eden borrowed directly from Karl R. Popper.

Eden expanded Simon and Newell's concept, noting that a computer program modeling some theoretical concepts can successfully be used for testing. Instead of the typical natural sciences procedure of formulating theoretical deductions and comparing them with the results of experiments, the proposed process will realize the following sequence:

theory → model → computer model implementation → empirical testing of implemented model

Eden gave several examples that show the use of computer simulation in various applications in natural sciences: the theory of deterministic chaos;

cognitive psychology; evolutionary biology and genetics; and cosmology. He also undertook to justify the thesis that the objects of study of computer science are very close to natural objects. According to the author, computational objects are close to their natural counterparts to such an extent that the theory behind the objects approaches the reality.¹⁴ In the scientific paradigm, computer science would progressively become one of the natural sciences.

Recognition of computational objects as natural seems to be a risky move. Eden, however, gives a few examples that show that this approach may not be paradoxical. The most telling example is that of DNA, which is an information structure encoding information needed for the development of organisms. You can therefore claim that such a structure could be represented by computational objects:

It seems reasonable to view the DNA script in the genome as executable code that could have been specified by a set of commands in a procedural imperative [programming] language. (Brent and Bruck, 2006)

For Eden, identifying computing with the natural sciences plays an important role. He takes the view that the crisis in computer science is due to the dominance of the engineering paradigm; the position, moreover, is supported by references to the views of many well-known researchers. Thus, the scientific paradigm sets the direction in which the methodology of computer science should develop. Computing should become therefore one of the empirical sciences; in the long run, it should provide an influx of new theories promoting, in turn, the further development of new engineering concepts. Thus, the shift away from the technocratic paradigm may go on to serve the development of engineering software. This is Eden's main thesis, demonstrating the importance of the debate on empiricism for the future of the entire field of computing.

The identification of the subject matter of computing with the subject of natural sciences is, however, based on a very strong assumption that natural processes can be adequately reduced to something computational. Such a strong ontological assumption demands strong justification, especially given those arguments pointing out that it is by no means so obvious.

4.4. Empirical computing and other sciences

From our point of view, we can make some observations regarding the methodological scheme presented. Considering the empirical testing process shown above, it may be concluded that it encompasses all kinds of computer

simulations of real processes. Simulation may be treated as an implementation of the model of the theory, and empirical study of the properties of simulation gives indirect information about simulated theory. One may ask whether it is useful to have an intermediate element (i.e. simulation) between the theoretical structure and the empirical testing process; in principle, this fact seems superfluous. The answer lies in the complexity of the experimental situation: often, direct measurement exceeds time constraints and is technically or financially impossible. Simulation allows one to gain knowledge about a problem in a relatively easy and fast way. This allows exclusion of many wrong paths of exploration and identification of those that are promising. Modern science, faced with increasingly complex issues, commonly uses this methodology. In this sense computing has become one of the more important analytic tools of modern natural sciences and we can talk about the ‘informatization of natural sciences’.¹⁵

So here we are dealing with an interesting evolution of methods of empirical science. The extended scope of empirical investigations is achieved at the expense of mediated measurement processes. In natural sciences the treatment of simulation programs as empirical objects has become a norm; it is by far the most interesting plane of interaction between computer science and other fields. Scientific paradigm leads, ultimately, to merging computing within the structure of other empirical sciences, opening an extremely wide field of possibilities for it. The empirical threads in computing are well connected with the most important philosophical problems of modern science.¹⁶ Ideas discussed here are often expressed within the computing community, e.g. in the works of Peter J. Denning, discussed in the subsequent section.

5. Peter J. Denning – empirical methods are the future of computing

Of particular interest to us is the relevance of Denning’s comments to the debate on the future of computing. Denning thinks that up until the 1990s computer scientists focused on the design and development of technology with the objective of constructing a reliable computing and network environment.

Now that this has been accomplished, we are increasingly able to emphasize the experimental method and reinvigorate our image as a science. (Denning, 2009, p. 29)

According to Denning, computer science should be an important partner for interdisciplinary scientific research. He also believes that computa-

tional processes are carried out objectively in nature, and whether we know or observe such processes or not. Computing science would therefore be a study of information processes, including those processes occurring in nature. Computers in this sense are only a tool and not an object of study (Denning, 2009, p. 30).

Denning uses the concept of a paradigm in a very loose sense, sometimes talking directly about three founding traditions: mathematical, experimental, and engineering. In contrast to Kuhn's paradigms, boundaries between Denning's paradigms are quite liquid and sometimes one researcher can simultaneously appeal to different traditions, as he says about himself (Denning, 2005, p. 29).

A legitimate natural science must, according to Denning, satisfy the following six criteria:

- Has an organized body of knowledge.
- Results are reproducible.
- Has well-developed experimental methods.
- Enables predictions, including surprises.
- Offers hypotheses open to falsification.
- Deals with natural objects. (Denning, 2009, p. 29)

Denning claims that computer science meets five out of these six criteria; thus, it could be considered a true empirical science. The only problem for Denning is the last criterion. However, with the discovery in recent years of several information processing phenomena in nature, this objection turns out to be misplaced. What's more, processes associated with information processing tend to be very common in nature. Besides, the boundary between what is natural and artificial is beginning to blur in the natural sciences as well, e.g. in chemistry some molecules are designed so they may be regarded as artefacts rather than natural objects. Denning takes a slightly different view from Eden: he argues that computer science investigates natural phenomena, which objectively realizes some computational and information processes. He even supports the thesis about the mathematical nature of the universe, claiming that reality pursues its objective by computational processes.

The author admitted that 'the old definition of computer science as the study of phenomena surrounding computers is now obsolete' (Denning, 2007, p. 14), which, he went on, has precipitated a 'striking shift' in the meaning of computer science. In his view, the experimental methods went beyond AI problems and determined the specificity of the entire computing science, as a discipline devoted to the study of computational processes, both natural and artificial (Denning, 2007). This change may be exemplified by the new

term ‘computing science’ (science about computational processes) replacing the old one of ‘computer science’.¹⁷ Denning also addressed the question of the fundamental principles of computing and formulated seven of them on the basis of known systems. This approach was intended to expand the scope of computing to encompass natural systems.¹⁸

Referring to the well-known work by W. Tichy, the author points out that computing’s loss of credibility was the result of a high percentage of failure of hypotheses, as some 50% of them were formulated with little or no verification.¹⁹ In other sciences, this number is roughly 5%. It is, for him, an argument for the necessity of the widespread empirical testing of hypotheses posed in computer science. What’s more, Denning points out that the scientific approach in computing becomes more common as the number of tested hypotheses increases. In this vision computing will soon become a mature empirical science, but its role will be unique. Denning refers to the opinion of Ken Wilson, the Nobel laureate, who argued that computing constitutes the third foundation of empirical science, beside theory and experimentation. In this perspective the concept of generalized computing became independent of the program of artificial intelligence and has become the subject of independent empirical studies.

6. Towards radical empiricism in computation: a generalized perspective on computations

The shift in the meaning of computing, since the 1990s, as mentioned by Denning, has become the focus of many researchers who recognize the concept of computations as a central concept of computing. There is no consensus on what is generally understood as computation, but one can certainly say that this kind of approach has reversed the perspective on computer science and changed some of its objectives.

Samson Abramsky, from Oxford University Computing Laboratory, summed up this process: computing originated with attempts to automate computations, the original goal was to evaluate mathematical functions, but the road to this was led by the formalization and mechanization of logical and mathematical reasoning. The current goal of computational systems is the realization of a specific behavior that does not obtain the result of the function. Thus, these reciprocal interactions are the main objectives of modern computing systems (Abramsky, 2008). These interactions are an integral part of the wider understood empirical sciences.²⁰ Therefore, this direction leads to a deeper empiricization of computer science. It is also tied closely to recent research in complex systems.

The revised approach allows for a generalization of the concept of computations to include natural processes: physical, biological, and even social,²¹ which can now be treated as information processes. Of course, this requires new computation models, meaning new in comparison to the standard models. Interestingly, this approach finds acceptance not only among theorists but also practitioners in the field of robotics, as it facilitates the solution of complex problems (e.g. the Moravec paradox) and use of the concept of embodied computations, or computations exploiting physical processes, performed by robot components (Sarosiek, 2013). Adaptive computational techniques employing the physical interactions of the robot with the environment turn out to be, in practice, often much more effective than conventional computational techniques used in the control.

Similar proposals exist, such as that of Gordana Dodig-Crnkovic. She suggests that the perception of the computational nature of natural processes should produce a new type of philosophy, one based on the concepts of computing and information (Dodig-Crnkovic, 2012).²² Dodig-Crnkovic writes that it is to be ‘a new philosophy of nature providing the basis for the unification of knowledge from currently disparate fields of natural sciences, philosophy and computing’ (Dodig-Crnkovic, 2012, p. 1).

The shift in computing objectives described needs to develop in-depth methodological analysis. So it is worth mentioning that this issue has already been treated in an extensive monograph,²³ and some of its aspects are covered in this publication. It should also be noted that methodological reflection on the nature of computation has already resulted in first classifications. Of particular interest is the proposal by Mark Burgin and Gordana Dodig-Crnkovic presented in their overview work on the concept of computations (Burgin and Dodig-Crnkovic, 2013). It is worth quoting the said attempt at classification to show the diversity of contexts in which the problems of computational processes are considered. It also shows how much the domain of interest of computing sciences has been extended: computing science understood as the science of computational (or information) processes.

According to these authors, the triad can be distinguished corresponding to Peirce’s triad of object-sign-interpretant:

- a) embodies computations (physical or embodied).
- b) symbolic or abstract computations.
- c) mental computations (mental).

The authors point out that mental and symbolic computations are based on computations that are always embodied. In the last class of computations, we can differentiate physical, chemical, and biological ones. The latter

are divided into the computation of physical, chemical and biological properties. Physical computations are also differentiated because of the scale to which they relate, from the quantum level, ending at cosmological scales.

Following the authors of *Handbook of Natural Computing*, this extension of the concept of computations brought Burgin and Dodig-Crnkovic acceptance of their thesis that the domain of natural computations is ‘the field of research that investigates both human-designed computing inspired by nature and computing taking place in nature’ (Burgin and Dodig-Crnkovic, 2013, p. 17). In this way, computing has become the science of computational (mathematical) nature. Thus, both simulations of natural processes used in the life sciences, as well as the use of natural processes in computations, are two sides of the same issue, which forces us to rethink the issue of the mathematical character of nature. However, one should agree with these authors that the conceptualization of computations as a natural information processing of information demands still better understanding. Currently, there are many reservations as to whether by extending its scope the concept has not lost its sense.

It seems that closely related with this trend is the concept of generalized computing presented by W. Marciszewski and P. Stacewicz (Stacewicz and Marciszewski, 2011, chapter 14). We are dealing here with a different starting point. For the authors, the primary concept is the generally understood Information Processing System (IPS). By analyzing this concept, one can notice a clear trend to enlarge the concept of information processes (and associated computation processes). What distinguishes the said concept from mainstream studies is its greater generality. Certainly, an interesting contribution to the methodological discussions is W. Marciszewski’s thesis to recognize social fabric as a full-fledged information processing (computational) system. The concept of computations has gained, in this way, a more interesting extension of meaning, and computing science shown another possibility of unification for knowledge of reality. Due to the nature of this publication, I direct the reader to the publication itself for a detailed explication of this proposal.

7. Conclusions

As shown in the above review, reflection regarding the empirical nature of computer science has accompanied this field for over four decades. In the ongoing discussion of the methodology of computing, there can be seen a growing awareness of the importance of the empirical aspects of

computer science. We can also see clearly that the current reflection is also deepening awareness of the fundamental nature of this science.

It is worth noting the continuing use of the already classic methodological concepts of Popper, Kuhn and Feyerabend, without taking into account the subsequent criticism of these positions. It is probably due to the fact that most of the contributors on these issues deal mainly with computer science and treat philosophy as marginal or, at most, secondary. As a result of this lack of in-depth discussion, the specificity of the methodology of computer science (or lack of specificity), is not recognized. This area, on the one hand, is one of the most important areas of modern knowledge, and, on the other, remains for philosophers a still poorly explored land.

It is also noteworthy that despite the lack of clarity regarding the fundamental issues of methodology of computing, the field has become, imperceptibly, an important component of the natural sciences (Winsberg, 2010; Leciejewski, 2013). Reflections on the role of computer simulations, and the role of computer technology to support experiments show not only a profound evolution of the methods of the natural sciences, but point to the deep links between computer science and the natural world. In this context, however, open questions remain about the future of the discipline. Can it disappear like the former microscopy, or retain its autonomy as a fundamental scientific discipline? Doubts that have been signaled as to whether computing science will survive as a separate study indicate that this area is so intimately integrated with other sciences that we lose sight of its individuality.

Slowly, we are also seeing that the majority of natural processes is computational in nature, which are the basic objects studied in computational sciences. The field is starting to appear as a unifying science, unifying domains as far apart as biology, physics, cognitive science, and social science. What are the limits of that unification? Will everything finally able to be computerized? It is probably too early to attempt to provide definitive answers. The differences in opinion about the nature of computing should not overshadow the fact of the deep and widespread informatization of sciences. Modern science without computing would be blind, and a scientific picture of the world can no longer survive without it.

N O T E S

¹ This publication was supported by the Faculty of Philosophy of the Pontifical University of John Paul II in Krakow.

² In this paper, I will not define the term “computing”; rather I will try to show how the scope of the meaning of this concept changed in recent times under certain pressures from

the empirical sciences. It is worth noting that today there is no prospect of consensus on the definition of computing.

About the different traditions of “doing” computing and definitional problems, one can read the work by Tedre (Tedre, 2011). The different attempts at formulating the definition of this field are discussed in the work by Rappaport (Rappaport, 2005, pp. 323–324). A somewhat modest list along with a brief discussion of the definitions is presented in the work of Dogig-Crnkovic (Dodig-Crnkovic, 2004). An interesting view upon the attempts to formulate the methodology of computing is provided by M. Tedre who also comments on the role of methodological anarchism in the initial stage of the development of science. He believes that this approach has managed to combine sometimes very distant epistemologically and methodologically fields, which has given rise to fundamental concepts such as the machine design paradigm to the stored-program computer paradigm. Computing is, in this perspective, the result of a fruitful methodological anarchy; however, it has a side effect of neglecting its methodological aspects. More on this topic can be found in (Tedre, 2006). (The work is available on the Internet at <http://www.cs.uku.fi/pub/dissertations/tedre.pdf>).

³ On a paradigm of machines with a stored program (stored-program paradigm) see (Tedre, 2006, pp. 439–440).

⁴ Certainly, from the point of view of the practice, the most important current issue of empirical computing is testing the reliability and efficiency of programs. Programs are treated as natural objects, which are examined with the classical methods of natural sciences. Problems of testing of correctness and reliability have become an integral part of the academic curricula and are subject to academic courses (see Noah Smith and David Smith, 2005). In his work, however, we will be interested in more general aspects of computing.

⁵ For a more detailed review of these applications of AI, see Flasiński (Flasiński, 2011, chap. 4).

⁶ This approach originated in the 1970s, and was shared by a group of researchers, including at least two more names: Alan Perlis and George Forsythe (See Denning, 2009, p. 29).

⁷ ‘The machine – not just the hardware, but the programmed, living machine – is the organism we study.’ (Newell and Simon, 1976, p. 113)

⁸ Following M. Flasiński, the physical symbolic system is “a set of elements, called symbols, out of which it constructs symbolic structures, called expressions and collections of processes (modifications, reproductions and destructions) operating on these expressions” (Flasiński, 2011, p. 6).

⁹ Most AI researchers agree that the devastating blow to the concept of the physical symbolic system as a paradigm of AI was dealt by J. Sarle’s Chinese Room Argument.

¹⁰ The position of the second thesis is analogous to naive falsificationism, with the only difference being that, in this case, a single case allows for the confirmation of the thesis, while in falsificationism, a single case allows for the falsification of it. Naivete of this position comes from fact that it does not consider many possible interpretations formulated only to prevent the confirmation of the thesis. To say it differently, the opponent of the thesis can always interpret the results as to prove that the hypothesis was not confirmed.

¹¹ We need to point out that even in this case there is a difference of opinions. The differences between computing and mathematics was postulated by Donald E. Knuth (see Knuth, 1974).

¹² Wegner differentiated scientific (empirical), mathematical, and technological paradigms and proposed the existence of the fourth one related to the control of the emergent complexity in information systems. The latter is associated with Wegner’s definition of

computing as the discipline of management of complexity. The issue of the philosophical foundations of computer science was also investigated two decades later by H. Fetzer (Fetzer, 1988), published also in (Colburn et al., pp. 321–358).

¹³ Eden brings in also a complexity argument; behavioral formal analysis cannot be realized in practice.

¹⁴ Eden is referring to Popper’s concept of verisimilitude.

¹⁵ More about the role of computer simulations can be found in (Winsberg, 2010) (Winsberg, 2014).

¹⁶ Eden’s position, as has been shown, is extreme. One should mention more moderate attempts trying to explain the changes in natural sciences under the wider acceptance of simulation. One such interesting work is a monograph by Winsberg (Winsberg, 2010). The author is cautious about the connections between the simulation and experience, but he admits that the introduction of simulation has brought an essential change in the methodology, posing several interesting philosophical problems. He also admits that so far philosophers do not perceive these changes. This is how he diagnoses the causes behind it: “philosophers of science have had a bias in favor of the proposition that the philosophically interesting action in the sciences occurs when new theories are proposed” (Winsberg, 2010, p. 135).

¹⁷ Denning believes that this term, while popular in Europe, better represents the specificity of discussed science than the term “computer science”, as it emphasizes the concept of information. Some of the discussed problems are, one has to observe, specific to the USA.

¹⁸ One can read more about this topic on the web page Great Principles of Computing <http://cs.gmu.edu/cne/pjd/GP/GP-site/>.

¹⁹ Denning believed that this is the main source of prophecies about the end of computer science.

²⁰ It is worth noting that these interactions may also form a part of simulated environments, extending in an interesting way the concept of empirical research.

²¹ Most often the literature considers physical, chemical, and biological processes only. Social processes are rarely mentioned in this context. It is worth noting that the interesting methodological analysis of the free market as a computational process was presented by Ludwig von Mises (Mises, 1920; for English translation see e.g. Mises, 2012) and Friedrich A. Hayek (Hayek, 1940). Interesting methodological analysis of this concept presented Witold Marciszewski (Marciszewski, 2005).

²² In another study, she claimed that “It is evident that natural computing/computing nature presents a new natural philosophy of generality and scope that largely exceed natural philosophy of Newton’s era, presented in his *Philosophiae Naturalis Principia Mathematica*. Natural computation bring us to the verge of true paradigm shift in modeling, simulation and controlling the physical world, and it remains to see how it will change our understanding of nature [...]” (Dodig-Crnkovic and Giovagnoli, 2012) Published also in: (Dodig-Crnkovic and Giovagnoli, 2013).

²³ Another current publication worth mentioning is the monograph by Zenil (Zenil, 2013).

REFERENCES

Abramsky, S., 2008. Information, Processes and Games, in: Adriaans, P., Ben-
them, J. van (Eds.), *Philosophy of Information*. North Holland, Amsterdam,
The Netherlands; Boston, pp. 483–549.

- Brent, R., Bruck, J., 2006. 2020 Computing: Can computers help to explain biology? *Nature* 440, 416–417. doi:10.1038/440416a
- Burgin, M., Dodig-Crnkovic, G., 2013. Typologies of Computation and Computational Models. arXiv:1312.2447 [cs].
- Colburn, T.R., Fetzer, J.H., Terry L Rankin (Eds.), 1993. Program verification: fundamental issues in computer science. Kluwer Academic Publishers, Dordrecht; Boston.
- Denning, P.J., 2009. Beyond Computational Thinking. *Communications of the ACM* 52, 28–30. doi:10.1145/1516046.1516054
- Denning, P.J., 2007. Computing is a Natural Science. *Communications of the ACM* 50, 13–18. doi:10.1145/1272516.1272529
- Denning, P.J., 2005. Is Computer Science Science? *Communications of the ACM* 48, 27–31. doi:10.1145/1053291.1053309
- Dodig-Crnkovic, G., 2012. Alan Turing’s Legacy: Info-Computational Philosophy of Nature. arXiv:1207.1033 [cs].
- Dodig-Crnkovic, G., 2004. Scientific Methods in Computer Science [WWW Document]. Proceedings of the Conference for the Promotion of research in IT at New Universities and University Colleges in Sweden. URL <http://www.mrtc.mdh.se/publications/0446.pdf> (accessed 10.21.14).
- Dodig-Crnkovic, G., Giovagnoli, R., 2013. Computing nature: Turing centenary perspective. Springer, Heidelberg, New York.
- Dodig-Crnkovic, G., Giovagnoli, R., 2012. Computing Nature: A Network of Networks of Concurrent Information Processes. arXiv:1210.7784 [cs].
- Eden, A.H., 2007. Three Paradigms of Computer Science. *Minds and Machines* 17, 135–167. doi:10.1007/s11023-007-9060-8
- Fetzer, J.H., 1988. Program Verification: The Very Idea. *Communications of the ACM* 31, 1048–1063. doi:10.1145/48529.48530
- Flasiński, M., 2011. Wstęp do sztucznej inteligencji (Introduction to Artificial Intelligence). Wydawnictwo Naukowe PWN, Warszawa.
- Harel, D., 1987. Algorithmics: the spirit of computing. Addison-Wesley, Wokingham, England; Reading, Mass.
- Hayek, F.A. v., 1940. Socialist Calculation: The Competitive ‘Solution’. *Economica*, New Series 7, 125–149. doi:10.2307/2548692
- Knuth, D.E., 1974. Computer science and its relation to mathematics. *The American Mathematical Monthly* 81, 323–343.
- Leciejewski, S., 2013. Cyfrowa rewolucja w badaniach eksperymentalnych: studium metodologiczno-filozoficzne (The digital revolution in experimental research: methodological and philosophical study), Seria Filozofia i Logika. Wydawnictwo Naukowe Uniwersytetu im. Adama Mickiewicza, Poznań.

- Marciszewski, W., 2005. Wolny rynek jako system przetwarzania informacji (Free Market as Information-Processing System), in: Heller, M., Mączka, J. (Eds.), *Informacja a rozumienie*. Wydawnictwo Diecezji Tarnowskiej Biblos; Polska Akademia Umiejętności. Ośrodek Badań Interdyscyplinarnych, Tarnów; Kraków, pp. 235–267.
- Mises, L. von, 2012. Economic calculation in the socialist commonwealth. Ludwig Von Mises Institute, Auburn University, Auburn, Alabama.
- Mises, L. von, 1920. Die Wirtschaftsrechnung im sozialistischen Gemeinwesen. *Archiv für Sozialwissenschaft und Sozialpolitik* 47, 86–121.
- Newell, A., Simon, H.A., 1976. Computer Science As Empirical Inquiry: Symbols and Search. *Communications of the ACM* 19, 113–126. doi:10.1145/360018.360022
- Noah Smith, David Smith, 2005. Empirical Research Methods in Computer Science [WWW Document]. URL <http://www.cs.jhu.edu/~nasmith/erm/> (accessed 10.22.14).
- Rapaport, W., 2005. Philosophy of Computer Science: An Introductory Course. *Teaching Philosophy* 28, 319–341.
- Sarosiek, A., 2013. Próby aplikacji paradygmatu ucieleśnionego umysłu w tworzeniu sztucznej inteligencji (An attempt of application of the embodied mind paradigm to the development of artificial intelligence). *Semina Scientiarum* 12, 97–107.
- Shapiro, S.S., 2001. Computer Science: The Study of Procedures [WWW Document]. URL <http://www.cse.buffalo.edu/~shapiro/Papers/whatiscs.pdf> (accessed 10.21.14).
- Stacewicz, P., Marciszewski, W., 2011. Umysł – komputer – świat. O zagadce umysłu z informatycznego punktu widzenia (Mind – computer – world: On the Riddle of Mind from an Informational Point of View). Akademicka Oficyna Wydawnicza EXIT.
- Tedre, M., 2011. Computing As a Science: A Survey of Competing Viewpoints. *Minds and Machines* 21, 361–387. doi:10.1007/s11023-011-9240-4
- Tedre, M., 2006. The Development of Computer Science: A Sociocultural Perspective, University of Joensuu Computer Science Dissertations. University of Joensuu Department of Computer Science and Statistics, Joensuu.
- Wegner, P., 1976. Research Paradigms in Computer Science, in: *Proceedings of the 2nd International Conference on Software Engineering, ICSE '76*. IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 322–330.
- Winsberg, E.B., 2014. Computer Simulations in Science [WWW Document]. The Stanford Encyclopedia of Philosophy. URL <http://plato.stanford.edu/archives/fall2014/entries/simulations-science/> (accessed 10.21.14).
- Winsberg, E.B., 2010. *Science in the age of computer simulation*. University of Chicago Press, Chicago.
- Zenil, H. (Ed.), 2013. *A computable universe: understanding and exploring nature as computation*. World Scientific, Singapore.