Scientific Methods in Computer Science

Gordana DODIG-CRNKOVIC

Department of Computer Science Mälardalen University
Västerås, Sweden, gordana.dodig-crnkovic@mdh.se

Abstract. This paper analyzes scientific aspects of Computer Science. First it defines science and scientific method in general. It gives a discussion of relations between science, research, development and technology.

The existing theory of science (Popper, Carnap, Kuhn, Chalmers) has *physics* as an ideal. Not many sciences come close to that ideal. Philosophy of Science (Theory of Science) as it is today is not of much help when trying to analyze Computer Science.

Computer Science is a new field and its object of investigation (universe) is a computer, which is an ever-developing artifact, the materialization of the ideas that try to structure knowledge and the information about the world, including computers themselves. However different, Computer Science has its basis in logic and mathematics, and both theoretical and experimental research methods follow patterns of classical scientific fields. Computer modeling and simulation as a method is specific for the discipline, and it is going to develop even more in the future, not only applied to computers, but also to other scientific as well as commercial and artistic fields.

Introduction

It is not so obvious, as the name might suggest, that the Computer Science qualifies as "science" in a sense traditional theory of science [3-6] defines the term. Computer Science (CS) is a young discipline and necessarily starting from the outset very different from mathematics, physics and similar "classic" sciences, that all have their origins in the philosophy of ancient Greece.

Emerging in modern time (in 1940's the first electronic digital computer was built), CS has necessarily other already existing sciences in the background. Computer Science draws its foundations from a wide variety of disciplines [11], [14], [16]. Study of Computer Science consequently requires utilizing concepts from many different fields. Computer Science integrates theory and practice, abstraction (general) and design (specific).

The historical development has led to emergence of a big number of sciences that communicate more and more not only because the means of communication are getting very convenient and effective, but also because a need increases for getting a holistic view of our world, that is presently strongly dominated by reductionism.

1. What Is Science

> *The whole is more than the sum of its parts.*
> Aristotle, Metaphysica

Talking about "science" we actually mean plurality of different sciences. Different sciences differ very much from each other. The definition of science is therefore neither simple nor unambiguous. See [1] and [2] for several possible classifications. For example, history and linguistics are often but not always catalogued as sciences.
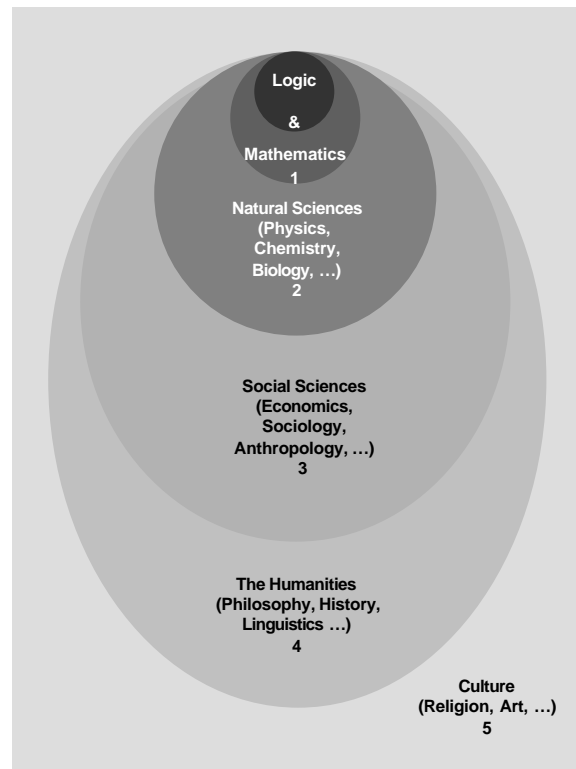
1.1 Classical Sciences

Figure 1. What is science? One possible view.

The figure above suggests that sciences have *specific areas of validity*. The logic and mathematics (the most abstract and at the same time the most exact sciences) are more or less important part of every other science. They are very essential for physics, less important for chemistry and biology[1], and their significance continues to decrease towards the outer regions of our scheme.

The logical reasoning as a basis of all human knowledge is of course present in every kind of science as well as in philosophy.

The structure of Figure 1 may be seen in analogy with looking into a microscope. With the highest resolution we can reach the innermost region. Inside the central region logic is not only the tool used to make conclusions. It is at the same time the object of investigation. Even though big parts of mathematics can be reduced to logic (Frege, Rusell and Whitehead) the complete reduction is impossible.

On every step of zooming out, the inner regions are given as prerequisites for the outer ones. Physics is using mathematics and logic as tools, without questioning their internal structure. In that way information about the deeper structure of mathematics and logic is hidden looking from the outside. In much the same way, physics is a prerequisite for chemistry that is a hidden level inside biology etc.

The basic idea of Figure 1 is to show in a schematic way the relation between the three main groups of sciences (logic & mathematics, Natural Sciences and Social Sciences) as well as the connections to thought systems represented by the Humanities.

Finally the whole body of human knowledge, scientific and speculative is immersed in and impregnated by the cultural environment.

Table 1. Sciences, objects and methods

SCIENCE	OBJECTS	DOMINATING METHOD
	Simple	*Reductionism (analysis)*
Logic & Mathematics	Abstract objects: propositions, numbers, ...	Deduction
Natural Sciences	Natural objects: physical bodies, fields and interactions, living organisms ...	Hypothetico-deductive method
Social Sciences	Social objects: human individuals, groups, society, ..	Hypothetico-deductive method + Hermeneutics
Humanities	Cultural objects: human ideas, actions and relationships, language, artefacts…	Hermeneutics
	Complex	*Holism (synthesis)*

1.2 Sciences Belonging to Several Fields

The development of human thought parallel to the development of human society has led to an emergence of sciences that do not belong to any of the classic types we have described earlier, but rather share common parts with several of these.

Many of the modern sciences are of interdisciplinary, "eclectic" type. It is a trend for new sciences to search their methods and even questions in very broad areas. It can be seen as a result of the fact that the communications across the borders of different scientific fields are nowadays much easier and more intense than before.

Computer Science for example includes the field of artificial intelligence that has its roots in mathematical logic and mathematics but uses physics, chemistry and biology and even has parts where medicine and psychology are very important.

We seem to be witnessing an exciting paradigm shift:

We should, by the way, be prepared for some radical, and perhaps surprising, transformations of the disciplinary structure of science (technology included) as information processing pervades it. In particular, as we become more aware of the detailed information processes that go on in doing science, the sciences will find themselves increasingly taking a meta-position, in which doing science (observing, experimenting, theorizing, testing, archiving,) will involve understanding these information processes, and building systems that do the object-level science. Then the boundaries between the enterprise of science as a whole (the acquisition and organization of knowledge of the world) and AI (the understanding of how knowledge is acquired and organized) will become increasingly fuzzy.

Allen Newell, Artif. Intell. 25 (1985) 3.

Here we can find a potential of the new synthetic (holistic) worldview that is about to emerge in the future.

2. The Scientific Method

The scientific method is the logical scheme used by scientists searching for answers to the questions posed within science. Scientific method is used to produce scientific theories, including both scientific meta-theories (theories about theories) as well as the theories used to design the tools for producing theories (instruments, algorithms, etc). The simple version looks something like this (see also Figure 2):

1. Pose the question in the context of existing knowledge (theory & observations). It can be a new question that old theories are capable of answering (usually the case), or the question that calls for formulation of a new theory.
2. Formulate a hypothesis as a tentative answer.
3. Deduce consequences and make predictions.
4. Test the hypothesis in a specific experiment/theory field. The new hypothesis must prove to fit in the existing world-view (1, "normal science", according to Kuhn). In case the hypothesis leads to contradictions and demands a radical change in the existing theoretical background, it has to be tested particularly carefully. The new hypothesis has to prove fruitful and offer considerable advantages, in order to replace the existing scientific paradigm. This is called "scientific revolution" (Kuhn) and it happens very rarely. As a rule, the loop 2-3-4 is repeated with modifications of the hypothesis until the agreement is obtained, which leads to 5. If major discrepancies are found the process must start from the beginning, 1.
5. When consistency is obtained the hypothesis becomes a theory and provides a coherent set of propositions that define a new class of phenomena or a new theoretical concept. The results have to be published. Theory at that stage is subject of process of "natural selection" among competing theories (6). A theory is then becoming a framework within which observations/theoretical facts are explained and predictions are made. The process can start from the beginning, but the state 1 has changed to include the new theory/improvements of old theory.

Figure 2 describes very generally *the logical structure of scientific method used in developing new theories*. As the flow diagram suggests, *science is in a state of permanent change and development*.

The one of the most important qualities of science is its provisional character: it is subject to continuous re-examination and self-correction.
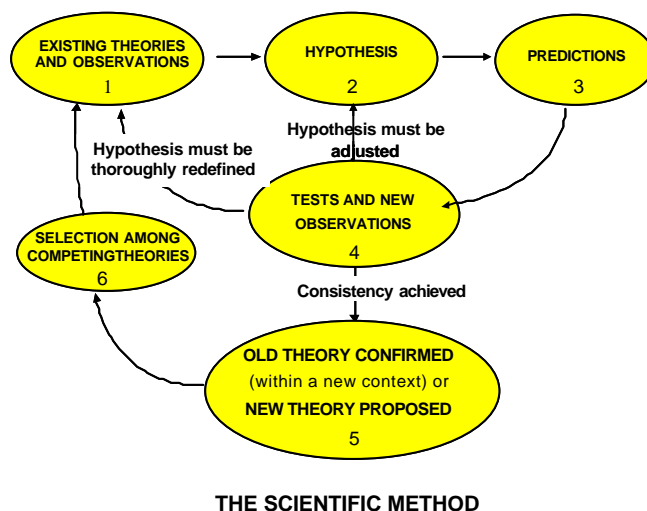

THE SCIENTIFIC METHOD

Figure 2. Diagram describing iterative nature of the hypothetico-deductive method

It is crucial to understand that the *logic of science is recursive.* Prior to every observation/experiment/theoretical test there is a hypothesis (2) that has its origins in the pre-existing body of knowledge (1). Every experimental/observational result has a certain world-view built-in. Or, to say it by Feyerabend [7], every experimental data is "theory-contaminated".

Here it is also interesting to mention that designing new experimental equipment or procedure match the same scheme:

(1) Start from existing theoretical/experimental framework; (2) Formulate the problem; (3) Infer consequences; (4) Test if it works as expected; (5-6) Accept.

As soon as a piece of equipment or method is designed and used as a tool for testing new hypotheses, it is supposed that it works according to the design specification. The detailed information about its internal structure is therefore hidden. *The same is true for the existing theoretical context of a theory under development - it is taken for granted.*

The scheme of the scientific method in Figure 2 is without a doubt an abstraction and simplification. Critics of the hypothetico-deductive method would argue that there is in fact no such thing as "the scientific method". By the term "the scientific method" they actually mean the concrete set of rules defining how to proceed in posing new relevant questions and formulating successful hypotheses. Of course, no such magic recipe exists!

The important advantage of the scientific method is that it is impartial:2 one does not have to believe a given researcher, one can (in principle) repeat the experiment and determine whether certain results are valid or not. The question of impartiality is closely related to openness and universality of science, which are its fundamental qualities. A theory is accepted based in the first place on the results obtained through logical reasoning, observations and/or experiments. The results obtained using the scientific method have to be reproducible. If the original claims are not verified, the causes of such discrepancies are exhaustively studied.

All scientific truths are provisional. But for a hypothesis to get the status of a theory it is necessary to win the confidence of the scientific community. In the fields where there are no commonly accepted theories (as e.g. explanation of the process of creation of the universe- where the "big bang" hypothesis is the most popular one) the number of alternative hypotheses can constitute the body of scientific knowledge.

3. Science, Research, Technology

In his famous reflections on science and technology, Aristotle has identified some key distinctions that are still frequently quoted and even used to analyze modern science and technology.

According to Aristotle there are important distinctions between science (episteme) and technology (techne) in their objects, principles of change, ends, goals and activities. The recent distinctions add methods, innovation form, types of result, and time perspective.

3.1 Aristotle's Science contra Technology

	Science	Technology
Object	unchangeable	changeable
Principle of motion	inside	outside
End	knowing the general	knowing the concrete
Activity	theoria: end in itself	poiesis: end in something else
Method	abstraction	modeling concrete (complex)
Process	conceptualizing	optimizing
Innovation form	discovery	invention
Type of result	law-like statements	rule-like statements
Time perspective	long-term	short-term

Table 2. Standard distinctions: science vs. technology

3.2 Modern Science contra Technology

Traditional sharp binary distinctions between science and technology seem however to fail when applied to contemporary science, because the underlying concepts of science are out-dated. Today's science is much more complex and heterogeneous than science of the Aristotle's time (the contemporary relations are illustrated by Figure 3), the fact that modern theory of science have to take into account.

That is why philosophy of science is in vital need of a deeper, more realistic understanding of contemporary sciences. The time is ripe for paradigm change in philosophy of science!
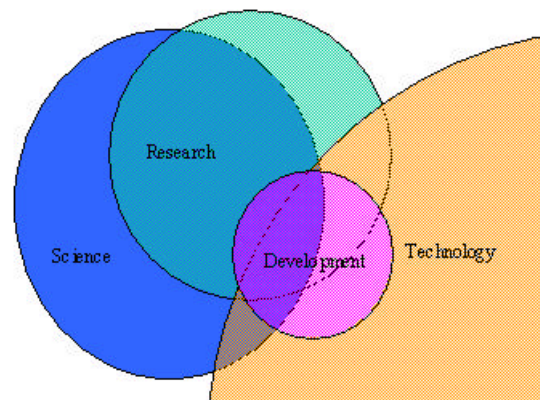
Figure 3. Relations between science, research, development and technology

4. What Is Computer Science?

According to the present view, Computer Science can be situated in a broader context of computing in the following way (see Figure 4) [12].
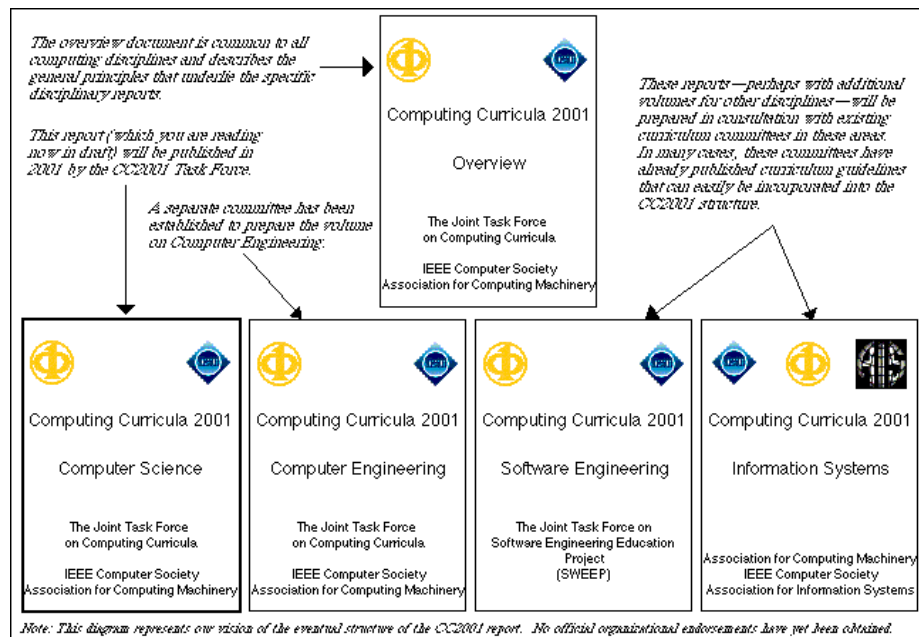
Figure 4. Computer Science in a structure of the field of computing

The discipline of computing thus encompasses Computer Science, Computer Engineering, Software Engineering and Information Systems. Here are some definitions.

1. The discipline of Computing is the systematic study of algorithmic[3] processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application [13].

2. Computer Science is the study of phenomena related to computers, Newell, Perlis and Simon, 1967.

3. Computer Science is the study of information structures, Wegner, 1968, Curriculum 68.

4. Computer Science is the study and management of complexity, Dijkstra, 1969 [8].

5. Computer Science is the mechanization of abstraction, Aho and Ullman 1992 [9, 13].

6. Computer Science is a field of study that is concerned with theoretical and applied disciplines in the development and use of computers for information storage and processing, mathematics, logic, science, and many other areas [11].

The second definition reflects an *empirical* tradition since it asserts that Computer Science is concerned with the study of a class of phenomena. The first and third definitions reflect a *mathematical* tradition since algorithms and information structures are two abstractions from the phenomena of Computer Science.

The third definition was used by Wegner as the unifying abstraction in his book on Programming Languages, Information Structures and Machine Organization. This view of Computer Science has its historical roots in information theory. It strongly influenced the development of Curriculum 68; a document which has been very prominent in the development of undergraduate Computer Science curricula afterwards. It is implicit in the German and French use of the respective terms "Informatik" and "Informatique" to denote the discipline of Computer Science.

It is interesting to note that the British term "Computer Science" has an empirical orientation, while the corresponding German and French terms "Informatics" have an abstract orientation. This difference in terminology appears to support the view that the nineteenth-century characters of British empiricism and continental abstraction have persisted.

The view that information is the central idea of Computer Science is both scientifically and sociologically indicative. Scientifically, it suggests a view of Computer Science as a generalization of information theory that is concerned not only with the *transmission of information* but also with its *transformation* and *interpretation*. Sociologically, it suggests an analogy between the industrial revolution, which is concerned with the utilizing of energy, and the computer revolution, which is concerned with the utilizing of information.

The fourth definition reflects the great complexity of engineering problems encountered in managing the construction of complex software-hardware systems.

It is argued in [15] that Computer Science was dominated by empirical research paradigms in the 1950s, by mathematical research paradigms in the 1960s and by engineering oriented paradigms beginning with the 1970s.

The diversity of research paradigms within Computer Science may be responsible for the divergences of opinion concerning the nature of Computer Science research. The fundamental question underlying all computing is: What can be (efficiently) automated?

Logic is important for computing not only because it forms the basis of every programming language, or because of its investigating into the limits of automatic calculation, but also because of its insight that strings of symbols (also encoded as numbers) can be interpreted both as data and as programs.

4.1 Sub-areas of Computer Science

Dijkstra said that to call the field "Computer Science" is like calling surgery "Knife Science". He noted that departments of Computer Science are exposed to a permanent pressure to overemphasize the "Computer" and to underemphasize the "Science". This tendency matches the inclination to appreciate the significance of computers solely in their capacity of tools.

According to [12], sub-areas of Computer Science curricula are:

1. Discrete Structures

2. Programming Fundamentals

3. Algorithms and Complexity

4. Programming Languages

5. Architecture and Organization

6. Operating Systems

7. Net-Centric Computing

8. Human-Computer Interaction

9. Graphics and Visual Computing

10. Intelligent Systems

11. Information Management

12. Software Engineering

13. Social and Professional Issues

14. Computational Science and Numerical Methods

We see that the list above includes all the elements mentioned in the previous section. As Computer Science develops, the list is expanding. Fields 7, 8 and 9 e.g. are new compared to predecessor (Denning report) list.

Let me summarize the characterization of CS. Computer Science does not deal merely with computer use, technology or software (program and data). It is a science that encompasses abstract mathematical thinking and includes an element of engineering. The mathematical element is expressed in finding solutions to problems, or in proving that the solutions do not exist, while the engineering element demands designing skills.

5. Scientific methods of computer science

Basically, we find characteristic features of classical scientific methods also in CS. The scheme of Figure 2 is applicable here as well. CS contains elements from different scientific areas of Figure 1, with main focus on 1 and 2.

What is specific for CS is that its objects of investigation are artifacts (computer-related phenomena) that change concurrently with the development of theories describing them and simultaneously with the growing practical experience in their usage.

A computer from the 1940s is not the same as a computer from the 1970s, which in its turn is different from a computer in 2002. Even the task of defining what a computer is in the year 2002 is far from trivial.

Computer science can be divided into Theoretical, Experimental and Simulation CS, which are three methodologically distinct areas. One method is however common for all three of them, and that is modeling.

5.1 Modeling

Modeling is a process that always occurs in science, in a sense that the phenomenon of interest must be simplified, in order to be studied. That is the first step of abstraction. A model has to take into account the relevant features of a phenomenon. It obviously means that we are supposed to know which features are relevant. That is possible because there is always some theoretical ground that we start from when doing science.

A simplified model of a phenomenon means that we have a sort of description in some symbolic language, which enables us to predict observable/measurable consequences of given changes in a system. Theory, experiment and simulation are all about (more or less detailed) *models of phenomena*.
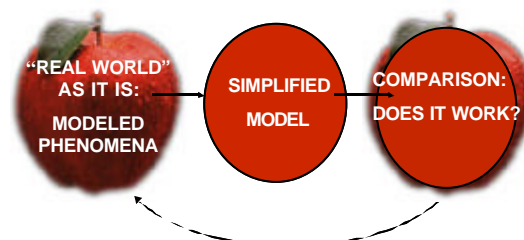

Figure 5. Modeling

"Real World"	Model
Program	Compiler theory
Artificial Neural Networks	Experiments testing ANN
Computer hardware	Simulation

Questions that must be answered during the modeling process:

- How to model? What to take into account? What to neglect?[4] Do we need to describe neglected features indirectly? What formalism to use in modeling?

- Is the model appropriate? - Does it serve its purpose? A model is always done with a certain "resolution". Do we have right level of abstraction?

- In what aspects does the features/behavior of the model differ from what is expected?

- In what way model differs from "reality" (other existing models or experimental results)?

- Validation: are the results valid (for the presumptive purpose)?

- Sometimes there are some special constraints put on models such as e.g. required conservatism (a consequence of general Precautionary principle). Conservative models are made in safety related systems. It means that it must be assumed that uncontrolled parameters (those not explicitly modeled, or those outside the modeling system) have their worst (most unfavorable) credible value.

It is always necessary to "benchmark" new models against old models in known specific cases and analyze their relative strengths/weaknesses. It is the part of *Comparison: Does it work?* of Figure 5.

We can compare Figure 5 with the Figure 2, which is illustrating scientific method. *"Real world" as it is: Modeled Phenomena* of Figure 5 corresponds *to Wonder: starting from world as it is (1)* of Figure 2; *Simplified model* corresponds to both *Hypothesis and Predictions (2, 3)*; *Comparison: Does it work?* corresponds to *Tests and new observations, Theory accepted, Selection among competing theories (4, 5, 6).*

Figure 5 has a model as a result, while Figure 2 is concerned with theories. Figure 2 is more detailed. But we can see that the modeling process scheme follows the general scheme of scientific method given in Figure 2.

5.2 Theoretical Computer Science

Concerning Theoretical Computer Science, which adheres to the traditions of logic and mathematics, we can conclude that it follows the very classical methodology of building theories as logical systems with stringent definitions of objects (axioms) and operations (rules) for deriving/proving theorems.

The key recurring ideas fundamental for computing are [13]:
- Conceptual and formal models (including data models, algorithms and complexity)

- Different levels of abstraction

- Efficiency

Data models [9] are used to formulate different mathematical concepts. In CS a data model has two aspects:
- The values that data objects can assume, and

- The operations on the data.

Here are some typical data models:
- The tree data model (the abstraction that models hierarchical data structure)

- The list data models (can be viewed as special case of tree, but with some additional operations like push and pop. Character strings are an important kinds of lists)

- The set data model (the most fundamental data model of mathematics. Every concept in mathematics, from trees to real numbers can be expressed as a special kind of set)

- The relational data model (the organization of data into collections of two-dimensional tables)

- The graph data model (a generalization of the tree data model: directed, undirected, and labeled)

- Patterns, automata and regular expressions. A pattern is a set of objects with some recognizable property. The automaton is a graph-based way of specifying patterns. Regular expression is algebra for describing the same kinds of patterns that can be described by automata.

Theory creates methodologies, logics and various semantic models to help design programs, to reason about programs, to prove their correctness, and to guide the design of new programming languages.

However, CS theories do not compete with each other as to which better explains the fundamental nature of information. Nor are new theories developed to reconcile theory with experimental results that reveal unexplained anomalies or new, unexpected phenomena, as in physics. In computer science there is no history of critical experiments that decide between the validity of various theories, as there are in physical sciences. The basic, underlying mathematical model of digital computing is not seriously challenged by theory or experiments.

There is a strong effort to define and prove the feasible limits of computation, but even here the basic model of computation is not questioned. The key effort is to prove that certain computations cannot be done within given resource bounds, as illustrated by the $P = NP$? question. The solution of this problem could have broad implications. It could also lead to a deeper understanding of the limits of human-computer reasoning power. In general, the "separation" problems, that is the questions if $P + NP$? PSPACE ?EXPTIME ? NEXPTIME ?EXPSPACE? are among the most important open problems in theoretical computer science [17].

In computer science, results of theory are judged by the insights they reveal about the mathematical nature of various models of computing and/or by their utility to the practice of computing and their ease of application. Do the models conceptualize and capture the aspects computer scientists are interested in, do they yield insights in design problems, and do they aid reasoning and communication about relevant problems?

The design and analysis of algorithms is a central topic in theoretical computer science. Methods are developed for algorithm design, measures are defined for various computational resources, tradeoffs between different resources are explored, and upper- and lower-resource bounds are proved for the solutions of various problems. In the design and analysis of algorithms measures of performance are well defined, and results can be compared quite easily in some of these measures (which may or may not fully reflect their performance on typical problems). Experiments with algorithms are used to test implementations and compare their "practical" performance on the subsets of problems considered important.
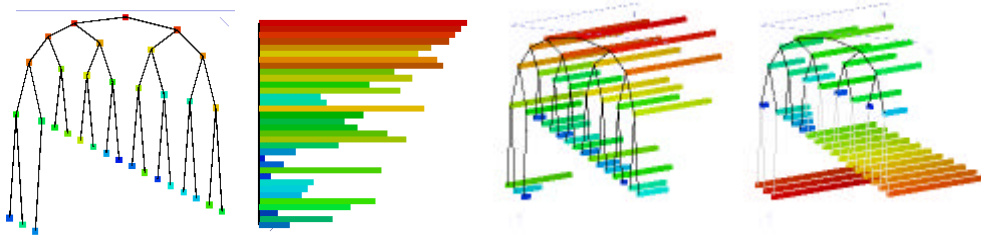
Figure 6. A 3D View of Heapsort Algorithm [21]

Some of the main methodological themes in theoretical Computer Science (inherited from mathematics) are *iteration, induction and recursion.*

- *Iteration.* The simplest way to perform a sequence of operations repeatedly is to use an iterative construct such as for- or while- statement.

- *Recursion.* Recursive procedures call themselves either directly or indirectly. This is self-definition, in which a concept is defined in terms of it self. (E.g. a list can be defined as being empty list or as being an element followed by a list). There is no circularity involved in properly used self-definition, because the self-defined subparts are always "smaller" than the object being defined. Further, after a finite number of steps, we arrive at the basis case at which the self-definition ends.

- *Induction.* Inductive definitions and proofs use basis and inductive step to encompass all possible cases.

Theoretical Computer Science seeks to understand both the limits of computation and the power of computational paradigms. Theoreticians also develop general approaches to problem solving.

One of theoretical Computer Science's most important functions is the distillation of knowledge acquired through conceptualization, modeling and analysis. *Knowledge is accumulating so rapidly that it must be collected, condensed and structured in order to get useful.*

5.3 *Experimental Computer Science*

The subject of inquiry in the field of computer science is information rather than energy or matter. However, it makes no difference in the applicability of the traditional scientific method. To understand the nature of information processes, computer scientists must observe phenomena, formulate explanations and theories, and test them.

Experiments are used both for theory testing and for exploration [10], [18], [19]. Experiments test theoretical predictions against reality. A scientific community gradually accepts a theory if all known facts within its domain can be deduced from the theory, if it has withstood experimental tests, and if it correctly predicts new phenomena. Conditio sine qua non of any experiment is repeatability/reproducibility. Repeatability ensures that results can be checked independently and thus raises confidence in the results.

Nevertheless, there is always an element of uncertainty in experiments and tests as well: To paraphrase Edsger Dijkstra, an *experiment can only show the presence of bugs (flaws) in a theory, not their absence*. Scientists are keenly aware of this uncertainty and are therefore ready to disqualify a theory if contradicting evidence shows up.

A good example of theory falsification in computer science is the famous Knight and Leveson experiment, [20] which analyzed the failure probabilities of multiversion programs. Conventional theory predicted that the failure probability of a multiversion program was the product of the fail-

ure probabilities of the individual versions. However, John Knight and Nancy Leveson observed that real multiversion programs had significantly higher failure probabilities. In fact, the experiment falsified the basic assumption of the conventional theory, *namely that faults in different program versions are statistically independent*.

Experiments are also used in areas to which theory and deductive analysis do not reach. Experiments probe the influence of assumptions, eliminate alternative explanations of phenomena, and unearth new phenomena in need of explanation. In this mode, experiments help with induction: deriving theories from observation.

Artificial neural networks (ANN) are a good example of the explorative mode of experimentation. After ANN having been discarded on theoretical grounds, experiments have demonstrated properties better than those theoretically predicted. Researchers are now developing better theories of ANN in order to account for these observed properties [19].

Experiments are made in many different fields of CS such as search, automatic theorem proving, planning, NP-complete problems, natural language, vision, games, neural nets/connectionism, and machine learning. Furthermore, analyzing performance behavior on networked environments in the presence of resource contention from many users is a new and complex field of experimental computer science. In this context it is important to mention Internet.

Yet, there are plenty of computer science theories that haven't been tested. For instance, functional programming, object-oriented programming, and formal methods are all thought to improve programmer productivity, program quality, or both. Yet, none of these obviously important claims have ever been tested systematically, even though they are all 30 years old and a lot of effort has gone into developing programming languages and formal techniques [19].

Some fields of Computing such as Human-Computer Interaction and parts of Software Engineering have to take into consideration even humans (users, programmers) in their models of the investigated phenomena. It is therefore resulting in a "soft" empirical approach more characteristic for Humanities and Social Sciences, with methodological tools such as interviews and case studies.

5.4 Computer Simulation

In recent years computation which comprises computer-based modeling and simulation, see Figure 7, has become the third research methodology within CS, complementing theory and experiment.

Computational Science has emerged, at the intersection of Computer Science, applied mathematics, and science disciplines in both theoretical investigation and experimentation.
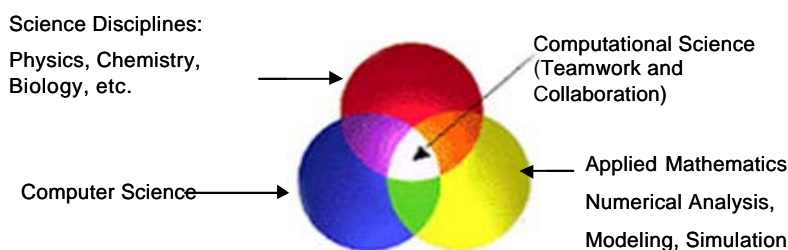

Figure 7. Computational Science

Mastery of Computational Science tools, such as modeling with 3D visualization and computer simulation, efficient handling of large data sets, ability to access a variety of distributed resources

and collaborate with other experts over the Internet, etc. are now expected of university graduates, not necessarily Computer Science majors. Those skills are becoming a part of scientific culture. Today, computing environments and methods for using them have become powerful enough to tackle problems of great complexity. With the dramatic changes in computing, the need for dynamic and flexible Computational Science becomes ever more obvious.

Computer simulation makes it possible to investigate regimes that are beyond current experimental capabilities and to study phenomena that cannot be replicated in laboratories, such as the evolution of the universe. In the realm of science, computer simulations are guided by theory as well as experimental results, while the computational results often suggest new experiments and theoretical models. In engineering, many more design options can be explored through computer models than by building physical ones, usually at a small fraction of the cost and elapsed time.
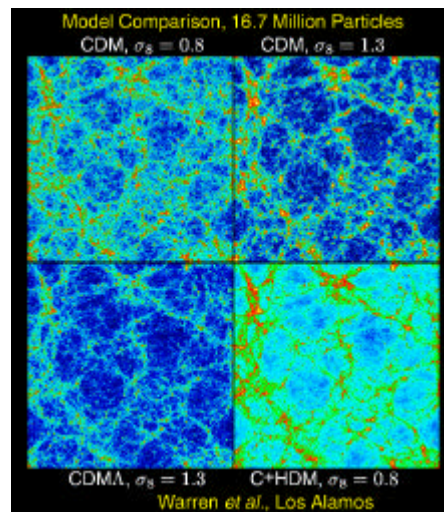

Figure 8. Simulation: Comparison of astrophysical *N*-body Cold Dark Matter Model variants.

Simulations such as the galaxy formation studies in Figure 8 can only be conducted on very powerful computers.

Science often proceeds with bursts of intense research activity. Even though the term "simulation" is old, it reflects the way in which a good deal of science will be done in the next century. Scientists will perform computer experiments in addition to testing scientific hypotheses by performing experiments on actual physical objects of investigation. One can also say that simulation represents a fundamental discipline in its own right regardless of the specific application.

Computational science involves the use of computers ("supercomputers") for visualization and simulation of complex and large-scale phenomena. Studies involving N body simulations, molecular dynamics, weather prediction and finite element analysis are within the thrust of computational science. If Computer Science has its basis in computability theory, then computational science has its basis in computer simulation.

Let's take some of the key focus areas of the past to shed light on the potential or existing role that simulation plays in each of them.

- *Chaos and Complex Systems:* The idea that one can observe complexity within a structurally simple deterministic model is of fundamental interest. Qualitative topological phase space features of linear systems may be determined statically, but for non-linear systems simulation must be used.

- *Virtual Reality:* Virtual reality means to *immerse* the analyst within the simulated world. Although, it is often seen as being synonymous with man-machine hardware interfaces, the tech-

nology must incorporate methods for building dynamic digital (virtual) worlds, which is a typical problem of computer simulation.

- *Artificial Life:* Artificial life is an outcome of computational science that challenges our definition of the term *experiment*. An experiment in artificial life is one where a computer program is written to simulate artificial life forms, often carrying along metaphors such as genetic reproduction and mutation.

- *Physically Based Modeling and Computer Animation:* Within computer graphics, there has been a noticeable move forward in the direction of physically based modeling (constraint-based models derived from physical laws).

The computing power of present day machines enables us to simulate an increasing number of phenomena and processes; especially the non-linear ones. Modern graphic capabilities makes this method a very attractive and user friendly.

6. Conclusions about the Scientific Methods of Computer Science

In spite of all characteristics that differ the young field of Computer Science from several thousand years old sciences such as mathematics, logic, and natural sciences we can draw a conclusion that Computer Science contains a critical mass of scientific features to qualify as a science.

From the principal point of view it is important to point out that all modern sciences are very strongly connected to technology. This is very much the case for biology, chemistry and physics, and even more the case for Computer Science. That is a natural consequence of the fact that the research leading to the development of modern computers very often is conducted within industry.

The engineering parts in the Computer Science have more or less close connection to the hardware (physical) aspects of computer. Software engineering is concerned with the problems of design, construction and maintenance of the, often huge, software systems that are typical of industry.

Theoretical Computer Science, on the other hand, is scientific in the same sense as theoretical parts of any other science. It is based on a solid ground of logic and mathematics.

The important difference is that the computer (the physical object that is directly related to the theory) is not a focus of investigation (not even in the sense of being the cause of certain algorithm proceeding in certain way) but it is rather *theory materialized, a tool always capable of changing in order to accommodate even more powerful theoretical concepts.*

References

[1] Universal Decimal Classification (UDC), http://www.chem.ualberta.ca/~plambeck/udc/

[2] Dewey Decimal Classification, http://www.tnrdlib.bc.ca/dewey.html

[3] *The Logic of Scientific Discovery*, Popper, K.R. NY: Routledge, 1999

[4] *An Introduction to The Philosophy of Science,* Carnap, R. NY: Basic Books, 1994

[5] *The Structure of Scientific Revolutions,* Kuhn, T. Chicago: Univ. Chicago Press, 1962

[6] *What is This Thing Called Science?,* Chalmers, A. Hackett Publishing Co., 1990

[7] *Against Method,* Feyerabend, P. London, U.K.: Verso, 2000

[8] *Research Paradigms In Computer Science,* Peter Wegner, Proc. 2nd Int. Conference on Software Engineering, 1976, San Francisco, California

[9] *Foundations of Computer Science,* A. V. Aho, J. D. Ullman, W.H. Freeman, New York, 1992.

[10] *Academic Careers for Experimental Computer Scientists and Engineers,* National Research Council Washington, D.C. http://books.nap.edu/html/acesc/

[11] *Computer Science; The Search for the Mathematical Theory*, Michael S. Mahoney, in Science in the Twentieth Century

[12] *Computing Curricula 2001,* http://www.computer.org/education/cc2001/index.htm

[13] *Computing as a Discipline*, Denning, P.J. et al. Commun. ACM 32, 1 (January 1989), 9

[14] *Tools For Thought: The History and Future of Mind-Expanding Technology,* Howard Rheingold, Simon & Schuster, 1985 http://www.well.com/user/hlr/texts/tftindex.html

[15] *Observations About the Development of Theoretical Computer Science,* J Hartmanis, in Foundations of Computer Science. 20th Annual Symposium Papers, 1979

[16] *Exploring the Power of Computing,* J.E. Savage, Addison-Wesley,1998.

[17] *Turing Award Lecture: On Computational Complexity and the Nature of Computer Science*, J Hartmanis, ACM Computing Surveys, Vol. 27, No 1, March 1995

[18] *ACM president's letter: performance analysis: experimental computer science as its best,* Peter J. Denning, ACM Communications, Vol 24, Issue 11, November 1981.

[19] *Should Computer Scientists Experiment More?*, Tichy, W.F., Computer , Vol 31 Issue 5 , May 1998.

[20] *An Experimental Evaluation of the Assumption of Independence in Multiversion Programming,"* J.C. Knight and N.G. Leveson, IEEE Trans. Software Eng., Jan. 1986, pp. 96-109.

[21] Algorithm Animation, A 3D View of Heapsort http://research.compaq.com/SRC/zeus/heapsort3D.html

---

[1] This is obviously a gross simplification. For e.g. computational biology and bioinformatics is mathematics the very essence of the field!

[2] *Impartial* is used here as synonymous for *objective, unbiased, unprejudiced, and dispassionate*. Note, however that this is the statement about *science, not about individual scientists* whose attitude to their pursuit is on the contrary as a rule indeed passionate. The fact that science is shared by the whole scientific community results in theories that are in a great extent *free from individual bias.* On the other hand the whole of scientific community use to share *common paradigms*, which are the very broad concepts deeply rooted in the *culture. Paradigm shift is a process that occurs in a very dramatic way, partly because of cultural (not strictly rational) nature of paradigm, (Kuhn).*

[3] An algorithm is *a set of rules for solving a problem in a finite number of steps.*

[4] These two questions might seem equivalent. Nevertheless it is important to address both of them in order to get a complete picture of a phenomenon, and to decide if the features not modeled explicitly in some way influence the explicitly modeled features. For example studying radiation safety of a system composed of several different materials, we can neglect materials that are "transparent" for original, let say high-energy radiation. However it is important to think what the consequence of omitting those materials would be if the original high-energy radiation induces some other radiation processes in materials that we have neglected or the neglected materials might not be transparent for secondary radiation in the system.