Fakultät Informatik und Mathematik

Informatik - 2023
Mathematik - 2024

Computer science - challenges in teaching now and in the future

March 2024

Professor Dr Michael Bulenda

Programme advisor for the Bachelor of Computer Science programme

Faculty of Computer Science and Mathematics, OTH Regensburg

Introduction

The physicist Niels Bohr is credited with the following quote: *"It is difficult to make predictions, especially about the future."*. I don't know whether this quote motivated the footballer Paul Gascoigne to say: *"I never make predictions and I never will."*. The quote attributed to Thomas J. Watson Sr, co-founder of IBM, *"I think there is a world market for maybe five computers."* shows just how wrong predictions can be. from 1943.

Computer science is subject to constantly accelerating development. Teaching as part of the Bachelor of Computer Science programme at the OTH is therefore required to enable graduates to participate in these developments on the one hand and to lay a solid foundation of stable knowledge and skills on the other.

In this article, I give a brief personal insight into the special features of computer science compared to other sciences and the influence these special features have on teaching and its development. Despite the warnings about predictions in the first paragraph, the focus is on an outlook for the near future of computer science and the teaching of it.

Computer science in culture and public perception

Before looking into the future, I would like to show a look into the past by R. Kurzweil, Head of Technical Development at Google LLC. The following graphic (from R. Kurzweil: Humanity 2.0, Lola Books. Kindle version. 2nd edition 2014) depicts the key events (from Kurzweil's perspective) of biological evolution and human technology on a logarithmic scale:

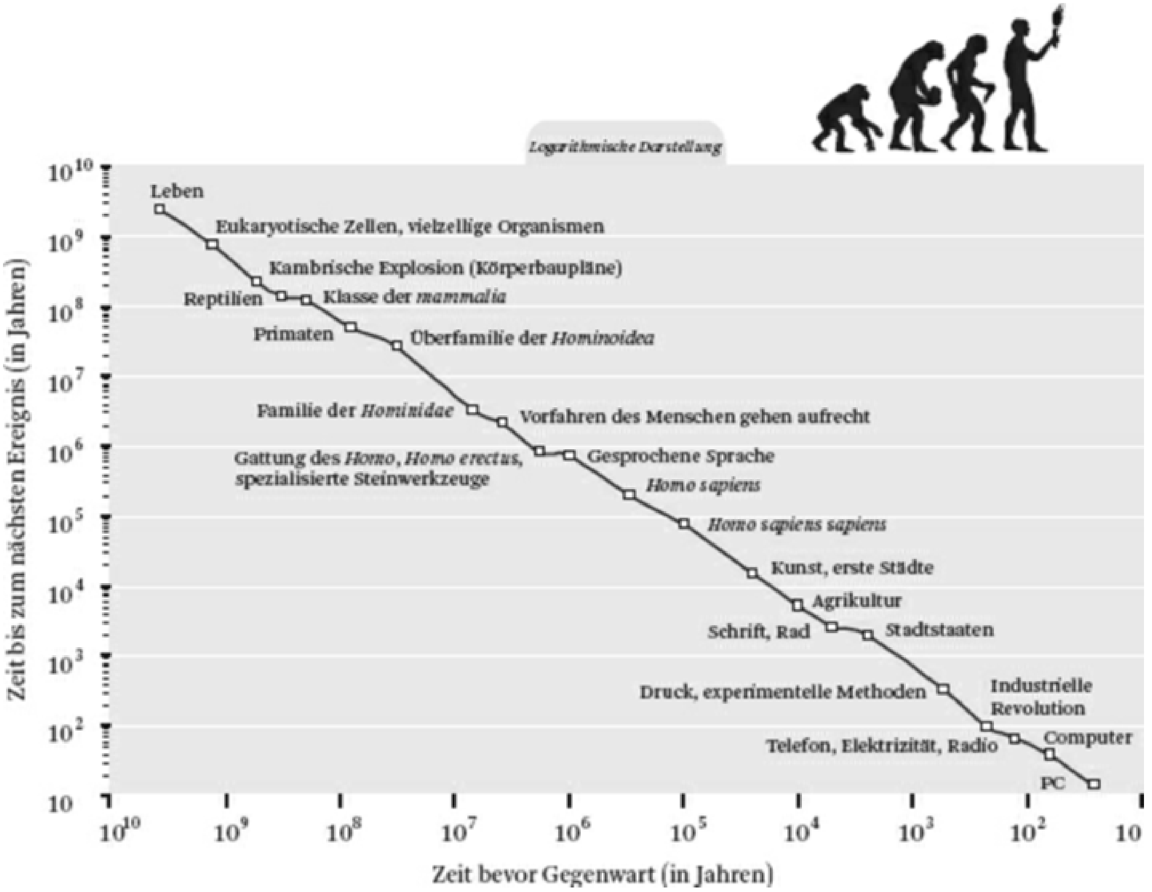

Illustration 1 Increasingly accelerated development (from R. Kurzweil: Humanity 2.0, Lola Books. Kindle version. 2nd edition 2014)

Based on the data presented, Kurzweil comes to the conclusion that a singularity is imminent. There are certainly other opinions on this. Without going into detail here about the term singularity and Kurzweil's prediction: The illustration above gives a strong indication that the development of human technologies is counter-intuitively exponential. The development in the recent past has been strongly characterised by computer science. Here, developments in software and hardware complement each other. As an example in the area of hardware, I would just like to give the following striking comparison (based on the price of an external hard drive): For 2 cents you can get a single sheet of a Zewa kitchen roll or 10 to the power of 9 transistors, for example.

This rapid and constantly accelerating development is also reflected in the fact that computer science topics are portrayed as science fiction in literature and films, which shortly afterwards either emerge as current research topics or are even realised. For topics outside of computer science, Jules Verne is probably the forerunner here, but there are also many examples of computer science-related topics. As early as 1968, Philip. K. Dick's novel "Do Androids Dream of Electric Sheep?" dealt with the differentiability of androids and humans, which Roger Penrose, a mathematician and physicist, addressed scientifically in his 1995 book "Shadows of the Mind". Karl Olsberg's novels "The System" from 2010 and "Mirror" from 2016 deal with topics that are current in computer science as a vision of the future: Social platforms, smart devices, artificial intelligence. Ernest Cline's well-known novel "Ready Player One" from 2017 also deals with the topics of artificial intelligence and virtual reality. The question of consciousness in connection with computer software is dealt with by Raphaela

Edelbauer in her 2021 novel "Dave" - as well as theoretically in the above-mentioned book by Roger Penrose. These are just a few examples that can be expanded to include the topics of swarm intelligence, metaverse, big brother scenarios, brain reading and the ever-present topics of autonomous vehicles, powerful corporations and monopolistic systems. All of these topics are often treated as dystopias in visions of the future - both in books and films. Examples of films from different decades include "Blade Runner" (1982), "Total Recall" (1990) and "The Circle" (2017).

In addition to its prominence in literature and film, computer science is of course also present in the general public perception. The subject of "digitality" in schools is just one topic that is discussed without the term "digitality" really being defined. There are also statements from politicians such as "Programming is as important as writing and arithmetic", which I cannot agree with, but which underpin the relevance of computer science in the political debate. Experts are also needed here to counteract an oversimplified view. Training these experts is one of the tasks of universities.

Bachelor of Computer Science - Challenges in teaching

The great importance of computer science, its rapid development and the often distorted public perception influence the question of how teaching should be organised in order to best prepare students for professional life. The focus should not only be on ensuring that graduates find jobs that are fun and well paid, but also that graduates are able to recognise that and how they can help shape their own future and that of society in a highly volatile environment. In other words, students should not ask themselves during their studies "Which train can I jump on?", but rather "What should the world look like in the future and how can we achieve this?".

Faced with this challenge, training to become a computer scientist must take two aspects into account: On the one hand, solid foundations in maths, computer science and an understanding of software projects must be taught. On the other hand, the foundations must be laid to be able to judge trends and hypes and, if necessary, follow them. New trends such as certain development frameworks or generative artificial intelligence must be viewed from two sides. Students or graduates of the degree programme should be able to deal with these new concepts, i.e. use them sensibly, as well as implement or further develop new concepts themselves. The fact that certain concepts are assessed differently over time or find new uses must also be taken into account. One example is the Java programming language: It was initially developed in the 1990s mainly to run small programmes on the computers of Internet users. Java quickly developed into a fully-fledged language for web applications whose compiled bytecode is platform-independent, i.e. it can be delivered for virtual machines on a wide variety of operating systems. This feature has greatly facilitated the development and delivery of programmes. This advantage of Java is now increasingly being relativised by the containerisation of applications, which has become more and more popular in recent years.

Bachelor of Computer Science - Looking to the future

As described above, students should be put in a position to actively follow, assess and contribute to current topics in computer science. Nobody knows what the future holds. Nevertheless, I will try to predict some trends in computer science and their influence on education.

- Artificial intelligence
  At this point, we will not be looking at
  the development of AI systems, but rather the use of generative systems such as ChatGPT and their influence on the work of computer scientists. There are speculations that traditional software developers will no longer be needed in the near future because AI systems will take over all the development work.
  The more likely development variant seems to me to be that such systems can speed up development work enormously and will be used specifically for simple programming tasks, but all higher-value tasks such as software architecture, software design, quality assurance and tasks that require communication with the customer, in particular the whole complex of requirements management, will remain in human hands.
- Ethical
  software now permeates all areas of life and is becoming increasingly complex. For example, a mid-range car today contains software consisting of around 100 million lines of code. As a result, ethical issues will increasingly come to the fore and play a role in education. This applies not only, but especially, to generative AI systems, where legal issues are also becoming more prominent. This concerns more everyday questions such as copyright law, as well as more complex questions such as how to prevent a newly developed general AI not adhering to ethical values after the singularity predicted by Ray Kurzweil and, in the worst case scenario, a Terminator scenario.
- Legacy code
  Back from science fiction topics to the banality of everyday life: ever since large software systems have been installed, these systems have become obsolete. They do this entirely without any parts wearing out or tyres bursting. Software becomes obsolete for various reasons. Increased user expectations that are no longer met, unsupported hardware platforms, unprofitable billing models, programming languages that only a few experts still master and which therefore represent a risk for companies, changed business models that are not supported or only inadequately supported by the old system are just some of the reasons why software systems need to be renewed. This not only applies to old technologies and programming languages, Java systems can also be categorised as legacy systems. Due to rapid development, there are also an increasing number of systems in need of renovation. It is also extremely rare for a software project to be carried out on a greenfield site, i.e. without legacy systems. There are almost always peripheral systems and/or systems that need to be replaced and/or modernised.
  These aspects will play a greater role in the future of training than in the past.
- Interdisciplinarity As
  described above, the simpler activities of a computer scientist are increasingly being supported by tools and may even be taken over completely. However, what will remain in human hands - at least for the foreseeable future - is the ability to familiarise oneself with the technical aspects of a software project, i.e. to familiarise oneself sufficiently with the specialist area for which the software can be used. This alone is necessary and has so far been impossible to automate because the requirements for a software system are practically never available in such a quality that the software can be developed without highly communicative processes. Furthermore, in most cases the requirements are not stable over time.
  This requirement for computer scientists is supported more by teaching a work attitude that is ready to learn than by specific modules in the curriculum.
- Cybersecurity
  Increasing complexity, completely new programming models and agile process models in software projects are opening up new points of attack for hacking software. This increases

the demands on the corresponding security measures. In order to stay ahead of the attackers, it is essential to continuously develop security topics at the university.

To summarise, computer science graduates need to be able to keep pace with constantly changing advances, based on a stable technical foundation. The essential aspects of the work will shift to increasingly high-value activities and require excellent communication skills.

The IT profession and training therefore remain exciting, varied, challenging and continue to develop in interesting ways.