reiner@hartenstein.de

RAW 2006
Rhodes Island, Greece, April 25-26, 2006

HPC → VHPC → VVHPC

Reiner Hartenstein

TU Kaiserslautern

(keynote)

6 Profs: nCS in Crisis

TU Kaiserslautern

Q&A: Computer Science Looks for a Remake
How can CS become an appealing career choice again ?

 2

TU Kaiserslautern

Q&A: Computer Science Looks for a Remake
How can CS become an appealing career choice again?

http://www.computerworld.com/printthis/2006/0,4814,110959,00.html

http://www.computerworld.com/careertopics/careers/story/0,10801,110959,00.html

Future Watch by Gary Anthes - MAY 01, 2006 (COMPUTERWORLD) - Two of the world's premier facilities for research and education in computer science are celebrating big birthdays this spring. **Stanford** University's CS department observed its 40th birthday in March, and **Carnegie Mellon** University's school of CS passed the half-century mark last month.

Despite the celebrations on both campuses, there is a deep malaise in computer science these days. Professors bemoan falling enrollments, a decline in prestige and a lack of attention to real-world problems. But, paradoxically, they say the future of CS has never been brighter, both within the discipline and in fields that computer technology will increasingly influence. Computerworld's Gary Anthes recently asked six CS professors what lies ahead for the field.

 3

TU Kaiserslautern

How important is computer science as a discipline today?

Birman: The importance of CS has never been greater. We're **discovering ways to build just about everything** out of small, simple mechanisms glued together with software, so no matter what you do, CS tends to be inside. And the scope of this new CS is amazing: We're **at the center of the action in biology, nanotechnology, particle physics.** If society is ever going to slash medical costs, CS will play the key role. I see CS as a sort of universal science. We're beginning to pervade everything.

Canny: Computers aren't very valuable yet, because the applications they perform are still elementary and routine. It's actually remarkable **how much we spend on IT, considering how little it does**. The most widespread applications are still e-mail and Microsoft Office. That should tell us something.

What we really need to be thinking about is what people are doing with computers and how we could help them to do those things much better. Since most people are doing knowledge tasks, that means machines understanding their owners' work processes much more deeply, finding semantically appropriate resources with or without being asked, critiquing choices and suggesting better ones, and tracking synergies with other groups within a large organization. Computers will leverage the human resources in the company more at a knowledge level. They will directly tie what they do to the creative processes of employees. The economic impact of that would be much bigger than anything we have seen so far.

4

TU Kaiserslautern

How important is computer science as a discipline today?

Carbonell: There are many more innovations in the works -- reliable speech understanding, high-quality machine translation, lightweight, high-capacity e-books, theft-proof electronic wallets and so on. These innovations often require better tools, extended programming languages and even new processor architectures. And we must see if other models of computing could surpass the tried and true. Nanotechnology and quantum computing could well be fundamental ingredients in the next revolution in computing. Massively parallel computation based on swarms of conventional chips underlies another potential revolution.

Chazelle: **CS is the new "new math,"** and people are beginning to realize that. CS, like math, is unique in the sense that many other disciplines will have to adopt that way of thinking. It offers a sort of **conceptual framework for other disciplines**, and that's fairly new.

 5

TU Kaiserslautern

Which areas in CS will show the most important and interesting advancements in the next few years?

Birman: I'm convinced that we're reaching a point where trustworthy computing will finally overcome its historical market-failure problems and become a commonplace requirement. And I see massive databases as a huge area and opportunity, particularly when the data is intrinsically distributed.
Web services will be the next interoperability standard, engulfing CORBA but going further. The world is going to become interoperable, and this leads to large-scale, distributed, service-oriented architectures. There's a lot of research to do, but this is going to be a phenomenally large thing, much larger than the original Web revolution.

Bryant: A recent big growth area in computing has been in using statistical methods to process vast quantities of data. Google is a prime example of that: They can return a query to you in a few seconds based on the contents of the entire [Web]. How do they do it? By maintaining massive data repositories that allow thousands of processors to operate on terabytes of data. **This data-centric style** of computing will drive many future efforts in **natural-language translation and understanding, astronomy and even epidemiology**. For example, we have a project that provides early detection of public health concerns by monitoring the sales of cough and cold remedies at regional pharmacies -- giving a heads-up before doctors start seeing disease trends among their patients.

 6

R. Hartenstein (panel chair): **6 Profs: nCS in Crisis**; RAW 2006 (co-located with IPDPS 2006), R hodos Island, Greece, April 25 - 26, 2006

Which areas in CS will show the most important and interesting advancements in the next few years?

TU Kaiserslautern

Carbonell: Artificial intelligence. Although those words may be somewhat out of fashion these days, much of the deep excitement and universally useful apps descend therefrom. For example: **speech understanding and synthesis in handheld devices, in cars, in laptops; machine translation of text and spoken language**; new search engines that find what you want, not just Web pages that contain query words; self-healing software, including adaptive networks that reconfigure for reliability; **robotics for mine safety, planetary exploration; prosthetics for medical/nursing care and manufacturing**; game theory for electronic commerce, auctions and their design to ensure fairness and market liquidity and maximize aggregate social wealth.

Chazelle: Definitely **algorithms**. What are the most amazing technological breakthroughs in recent years? TCP/IP, **whole-genome shotgun sequencing**, Google, quantum factoring -- all of them algorithms. We're in for huge surprises.

Is the looming end of Moore's Law a key driver for CS today?

TU Kaiserslautern

Birman: Not in my view. My machine is fast enough. Trustworthy computing is a much more urgent issue. After all, if we move all these critical systems to computer networks and don't solve the trust issue, we'll be cooked, and faster machines are really not going impact that issue one iota.

Canny: We're still stuck in the paradigm of "more cycles, more value." Now is the time to really start looking around. The market for computing has changed radically and irreversibly. What really matters is how useful our artifacts are to the people who are buying them. We don't know these people anymore, and we don't care much what they're doing with IT. That has to change. We're at an economic cusp right now. Sales have slowed, jobs have slowed, enrollments have slowed. It's not a technical problem; it's a problem of creating value. Every mature industry has to face that eventually. We've had a fun childhood without many cares, but now we're grown, and we have to figure out what we can do for the world that really matters.

Is the looming end of Moore's Law a key driver for CS today?

TU Kaiserslautern

Chazelle: We're **at the tail end of Moore's Law**. Because of power-dissipation issues, the party's essentially over. Back to parallel architectures. Huge work ahead.

But this will be the best thing that can happen to CS. Moore's Law has been tremendously beneficial to society. At the same time, it's been so damn powerful that it has set back the development of algorithms. But that's about to change. Any student interested in science and technology needs to learn to think algorithmically. That's the next big thing.

Dally: The current road maps show [Moore's Law] continuing for at least another 15 years. Even after that point, it will take many years for information systems to fully exploit the densest semiconductor devices. One can also expect that other technologies may emerge to continue scaling -- perhaps in a different way.

How can CS be made a more attractive choice for students?

TU Kaiserslautern

Birman: Not in my view. My machine is fast enough. Trustworthy computing is a much more urgent issue. After all, if we move all these critical systems to computer networks and don't solve the trust issue, we'll be cooked, and faster machines are really not going impact that issue one iota.

Bryant: We should stop scaring them away. Predicting that all IT jobs will move offshore could become self-fulfilling. New jobs are growing faster than old jobs are moving offshore, and that trend will continue. We need to stop putting them to sleep. Students who take computer science classes in high school are taught how to write programs in Java, and their assignments have them writing code that does tedious things like sort lists of numbers. They do not learn about any of the big ideas of computer science.

Canny: **We're losing in quality -- principally to bioengineering which is now the best students' top choice** -- and diversity. I a problem of social relevance. Minorities and women moved fastest in areas such as law and medicine that have obvious and compelling soci impact. We've never cared much about social impact in CS.

How can CS be made a more attractive choice for students?

TU Kaiserslautern

Chazelle: I roll my eyes when I hear students say, **"CS is boring, so I'll go into finance."** Do they know how dull it is to spend all-nighters running the numbers for a merger-and-acquisition deal?

No.
People have run away from CS **because they are worried about outsourcing.** This is a valid concern that can't be waved away by simply repeating the mantra that CS is cool.

Dally: We need to clear up many misconceptions about the field. Prospective students should understand that there are plenty of CS jobs in the U.S. and they pay well, that most CS jobs involve working with teams of people and place a premium on communication skills and teamwork -- it's not just a bunch of nerds working individually at terminals -- and that CS is so central to so many aspects of our economy that a CS education is good preparation for many careers.

How should CS programs be modernized?

TU Kaiserslautern

Birman: We need to realize that we're losing a lot of students around Grade 10. So we need to revamp the way CS is taught in high school to focus much less on programming and much more on problem-solving and puzzles.

Kids also need to work with things that are fun -- robot dogs that follow their owner around and growl at people who are wearing pink socks -- and do much less coding. Kids need to be grappling with information management issues, like the challenges of securely managing medical records and the legal and ethical issues that arise if we put monitoring systems in homes to keep an eye on the elderly, or in cars to provide emergency services.

TU Kaiserslautern

How should CS programs be modernized?

Chazelle: Much of the curriculum is antiquated. Why are we still demanding fluency in assembly language today for our CS majors? Some curricula seem built almost entirely around the mastery of Java. This is criminal. The curriculum is changing to fulfill the true promise of CS, which is to provide a conceptual framework for other fields. Students need to understand there's more, vastly more, to CS than writing the next version of Windows. For example, at Princeton, we have people who major in CS because they want to do life sciences or policy work related to security, or even high-tech music. In all three cases, we offer tracks that allow them to acquire the technical background to make them intellectually equipped to pursue these cross-disciplinary activities at the highest level.

TU Kaiserslautern

Bernard Chazelle says CS lacks a "great popularizer" such as Stephen Hawking in physics. Does CS need such a person?

Birman: I tend to view Bill Gates as our most public advocate, but I agree with Chazelle about this. We need a guy like Oliver Sacks or Carl Sagan, and the field doesn't have anyone in this role.

Bryant: For years, we didn't have to worry about our public image. Students clamored to get involved with computers, and government agencies were generous with their funding. We need a Stephen Hawking, a Carl Sagan or a Richard Feynman, but I have no idea who that would be.

Canny: To be brutally realistic, I think that CS as it's constructed today isn't as exciting as other sciences. It's necessarily a handmaiden, and other scientists see it as such. But Nicholas Negroponte [co-founder of the MIT Media Lab] is almost surely as charism.

TU Kaiserslautern

Bernard Chazelle says CS lacks a "great popularizer" such as Stephen Hawking in physics. Does CS need such a person?

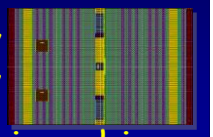
Carbonell: CS needs a great communicator who lives the excitement, is deeply respected by his or her peers, and can reach out and communicate clearly with any educated person via his books. We have no such person in CS. Perhaps Raj Reddy [a Carnegie Mellon computer science professor] has the right kind of talents.

Dally: I would say that CS needs a serious public relations effort. A prime-time TV show, Silicon Valley Software, could also do a lot of good, if done right, to show people what computing careers are really about and to clear up many misconceptions about the field.

TU Kaiserslautern

modern FPGA bestsellers:

The new model is reality:
FPGA fabrics, together with
several µprocessors,
many memory banks,
and other IP cores,
on the same COTS microchip



TU Kaiserslautern

DSP platform FPGA
[courtesy Xilinx Corp.]