

THE DEATH OF COMPUTING

*Neil McBride
School of Computing
De Montfort University
The Gateway
Leicester
LE1 9BH*

THE DEATH OF COMPUTING

Hi, David, Coffee?

Yes please. I need a break after that tutorial group. Sometimes I wonder why they're here particularly now they're paying £3000.

Have you seen this? It appears on the BCS Future of Computing website. A lecturer at De Montfort is saying that computer science is dead or a least dying.

Humph! From the effort my students are putting in I say it's already dead.

The article caused quite a stir. Apparently, the geeks are up in arms. Some email group called SlashDot has been debating it. And the computing professors were really annoyed when a version appeared in THES.

No sugar, thanks. What does it say then?

Have a look.

We all know there's a crisis in university computer science departments. Student numbers in computer science are dwindling. At the same time the computing unit of funding has fallen. And the onset of fees has made students think twice about joining a profession where the plethora of new jobs in the 1990s has reduced to a trickle and it's only just looking as if employment prospects may be on the upturn. Dropping numbers of A Level students, a view that IT is a job for geeks and social misfits and a perception that there's nothing new or interesting in computing doesn't help. And the problem's global. In the US, the number of students choosing computer science dropped by 39% between 2000 and 2005. In Australia, cuts in IT academic staff are the order of the day.

Well that doesn't sound contentious to me. That's the situation for computer science. We all know that. Anyway, we're ok. Information systems in a business school.

You must be joking. We're in a worse state than they are in. Information systems applications were down 9.7% this year according to the very same THES. Only anatomy's worse and who on earth would do a degree in that? And besides, at least school kids know what computer science is. Who knows what IS is? Even we don't. We write books and articles about what the hell is IS? And then expect students to apply for an 'information systems degree.'

Kids are into ICT nowadays.

Maybe that's why this guy at De Montfort has developed an ICT degree. Children know what ICT is and may be willing to do a degree in it.

Is he advertising the degree then?

That seems to be the purpose of the article.

In such dire circumstances, it's tempting to hanker after the glory days when computer science ruled, departments were full, and students flocked to a leading edge discipline where the ideas were fresh. We long for the days when structured programming ruled, when systems analysis was exciting, distributed computers were being created and there were uncharted vistas of applications to be written.

I guess ours was the heyday of SSADM. Sixteen month projects, hordes of analysts, monolithic systems which gave us massive scope for using CASE tools, lots of documentation.

And systems that were well scoped and predictable.

Yes, you knew where you were then. When I wrote a billing system I knew who the users were, what the scope was, what the processes were. But I think ERPs like SAP changed that. Who would invest in writing a billing system nowadays?

It's a mature market now. And we had hordes of business information systems students. That seems to be where he's coming from. Where have all the students gone? What's putting them off?

But that is the past. Today the ship is holed below the waterline.

I doubt whether the computer scientists think so.

They've got their head in the sands. Even the president of the ACM is writing about the dire state of computer science in the Communications of the ACM.

Are we holed below the waterline?

No we're more left high and dry. I think the tide's gone out and the good ship Information Systems is stranded. We're peddling a set of concepts no one wants.

As the ship sinks, computing academics fiddle on the deck hoping to avoid the icy waters. We look to games programming for our salvation, designing games programming courses and reducing a wide-ranging industrial discipline to a set of geeks programming computers to zap spacecraft and dismember aliens. It's a sorry sight to see computing academics fighting for the last few lifeboats. But the heyday of massive liners, full of programmers, plying the commercial sea-lanes is over. There may be room for a few luxury liners, but most of us fly on budget airlines.

At least they've got games programming to turn to. I was at a school's careers fair. Loads of kids wanted to do games programming. There always be a place for the

programming fanatic. But what about the rest of us? There aren't any luxury liners left for us.

Fine where do I collect my P45?

Maybe you don't need a luxury liner. Maybe you adapt and get on the budget airlines.

What are information systems budget airlines then? I mean nowadays you got agile methods, extreme programming. You don't even do any design. You just tell a story, define the tests and get programming.

Yes, but you need much closer connection with the business. There's even more of a role for people with the skills to communicate with the business.

It's a ruse by the computer scientists to eliminate system analysts. Let's just get down to programming. Rigorous programming, that's what matters.

No come on, you're not going to start a fight on the bridge of a sinking ship are you? Fighting over the last few crusts of bread. We should be on the same side as the computer scientists, working together to develop a discipline.

It's easy to think that the problem is that people (read potential students) just don't understand how exciting computing is and that this can be fixed by a bit of sharp marketing, slick videos and some school visits. But the students are not that gullible. The real nature of the problem lies at the roots of the discipline. Something significant has changed. There is the smell of death in the air.

That would certainly make the programmers unhappy. People don't like being told that their skills are dying and their work disappearing.

It does sound a bit over the top. But there is a point here. Students perceive IT as boring, because it's just become part of how they live. It's nothing exciting and different. Students don't go to university to do exactly the same thing they can do in their bedroom at home.

I don't know about that!

We won't go there. The point is, you expect something exciting, different. ICT is all the same. They've done it at primary school, at secondary school. They're online. It's built into their homes, for goodness sake.

There's the issue of employability also.

Yes, but even here there's a perception that the types of degrees we offer don't make people employable. The skills set is not right. It doesn't match what's happening in university. I mean most IS textbooks are based round the systems development life cycle. They teach people to build systems from scratch. They assume that they're working on a greenfield site. It's totally artificial.

I remember developing a system for running a garage. I don't think you'd do that nowadays, there's so many packages out there.

Precisely. Most organisations our students go into won't design and write applications from scratch. They're be involved in integration, implementation, connecting systems together.

I once heard Peter Checkland say that most IT departments are system integration departments.

And do we teach system integration? Enterprise resource packages? Systems migration. No!. We teach based on a fictional IS environment which doesn't exist out there.

Do we know how to do systems integration? Are we researching it? I mean I once examined SSM to see how it might support integration and boundary issues. It didn't seem to cope at all. It's alright with a well-bounded system and dealing with the problems inside that system.

Yes, IS should be about information flows between systems. Connectivity. Organisational interfaces.

Well McBride does get on to that, but first he seems to be saying that the role of computers has changed.

In the early days, computer science was populated by mathematicians and physicists excited at the prospect of vastly accelerated computation. New languages were developed. FORTRAN, Algol, COBOL, and PL/I took root. The foundations of programming were laid. There was excitement at making the computer do anything at all. Manipulating the code of information technology was the realm of experts: the complexities of hardware, the construction of compilers and the logic of programming were the basis of university degrees. The power of hardware has increased, as IBM 370s in air-conditioned warehouses gave way to computers in the home and advanced robots become this year's Christmas toy.

Now there is less need for low-level programming or any programming at all. Who needs C when there's Ruby on Rails? Now vastly complex applications for businesses, for science and for leisure can be developed using sophisticated high-level tools and components. Virtual robots –Zooks – can be created by eight-year olds without needing programming, logic or discrete mathematics skills. Web designers build complex business sites, graphic designers build animations, accountants assemble business systems without needing to go object-oriented. Computer science has lost its mystique. There is no longer a need for a vast army of computer scientists. The applications, games and databases that students once built laboriously in final year projects are bought at bookshops and newsagents.

He's got a point here. Everybody does their own design. Where is the need for the systems analyst?

I think the roles changing. The translation into computer artefacts is much easier, so the information systems professionals skills are much more at the level of systems thinking, of unwinding the business processes, creating coherent processes and showing how they can be translated into business systems.

Yes, we need more of a bottom-up approach. User-Centred Design. Usability Engineering.

Not quite. That smacks too much of HCI for me, although it does have a task focus. The thing is regardless how pretty a web system is because of the efforts of our esteemed colleagues in art and design, the underlying processes need to be designed and connected with the data. This requires an understanding of the business, its information flows, its connections with other businesses that a programmer is neither going to have or be interested in acquiring.

So we embed ourselves in the business more.

Precisely. Haven't you got a meeting to go to?

Another QA discussion. I think they can do it without me. Besides I'm worn out. I'd only fall asleep. Our three year old is keeping us awake. He just won't stay in bed. You know he already knows about mice, spacebars, the Internet. He picks it up from his brother.

Perhaps too much computing is keeping him awake.

I hadn't thought of that.

Another coffee?

Please.

The thing is. Information systems is a discipline of the artefact. Its design, its use, its role in society. And the nature of the artefact has changed beyond imagination. The Internet, Ubiquitous computing, RFID, Grid computing, Films and music delivered on the mobile. Mobile communications in the middle of Africa. The artefact is much more diverse. It's not just commercial information systems in organisations with class diagrams, use cases and sequence diagrams. It's blogs, shopping malls, Web 2.0. It's disappearing computers embedded in walls, shoes, brains, blood. It's 100,000 different systems interacting in every area of life. It's independent agents seeking and gathering information. It's a wide variety of applications – bioinformatics, environmental computing. And we don't have the models to deal with such a diversity of information systems.

You're right there. We need new design models, new ways of evaluation, new ways to focus on the social aspects, on the technology adoption.

We always got TAM.

Very funny. Doesn't our obsession with TAM just show the paucity of ideas in Information Systems? But I think we need to pick up on design science, to return to the massively evolving artefacts, to find new design and evaluation approaches. We can't just exist on UML. It's up to us to develop new models and techniques. How do you design and evaluation any ubiquitous computing artefacts? How do you analyse e-commerce applications where the focus has moved from the corporate to the individual, where attracting the customer is as important if not more important than completing the business process? Anyway, what else does he say?

If the gap between public knowledge and academic curriculum isn't large enough, the gap between academia and industry practice is a gaping hole. While academic departments concentrate on developing new computer systems in an ideal organisational environment, a lot of industry has moved away from in-house development to a focus on delivering a service. As commercial software products have matured, it no longer makes sense for organisations to develop software from scratch. Accounting packages, enterprise resource packages, customer relationship management systems are the order of the day: stable, well-proven and easily available. IT departments now focus on contracts, tenders, service level agreements, training, system usage and incident management. IT is about deploying resources to meet the information needs of its customers. Implementation, facility management, systems integration, service management, organisational change even environmental audit are the language of IT. These hardly feature on computing courses.

This is the same point really isn't it? What we teach on IS courses is nothing to do with what happens in an IT department.

I worked in a city council.

So you often tell us.

Well the student's like to hear about real life.

Real life ten years ago!

Give us a break. I go back regularly. I was only there two weeks ago. The whole emphasis is on service management. They're implementing ITIL. They're thinking about ISO 20000. Incident management, problem management. Managing availability communicating with suppliers. Getting users onboard. Training. Upgrading. And most of all business alignment. Alignment of IT strategy with business strategy. Alignment of the operations, the governance, the culture.

These are the core competencies identified by Feeny and Wilcocks.

Yes, the vendor development, contract management etc. No mention of endless exercises in analysis and design. It's service sciences. Even IBM are promoting service sciences and talking to leading academics in service management research: Stephen Brown, Roland Rust, Bo Edvardsson.

Name dropper!

They're looking at how you interact with the customer, Service encounters. The helpdesk. There was even an issue of Communications of the ACM on it.

If it's in CACM it must be good.

Well it would get them RAE points!

The point is, all the issues and subjects that concern IT departments and vendors – even Microsoft – do not get mentioned in IS courses. Who teaches IT service management? Systems Integration? Contract management?

So information systems is just as out-of-tune with industry as computer science then?

If not more.

The environment within which computing operates in the 21 century is dramatically different to that of the 60s, 70s, 80s and even early 90s. Computers are an accepted part of the furniture of life, ubiquitous and commoditised. Like cars, a limited number of people are interested in their construction, more live by supporting and maintaining them; most of us accept them as a black box, whose workings are of no interest but which confer status, freedom and convenience. Indeed, whereas building a new car needs mechanical know-how, building a new computer application can be done by the user who has no grounding in computer science.

Black box sounds like a reference to actor network theory.

It probably is.

Computing is also affected by globalisation. The loss of jobs in IT and the declining computer science enrolments is a global problem for developed countries. Since the software product can be transmitted almost instantaneously, why develop it in expensive facilities in the west? Armies of highly trained computer scientists are available in India, Sri Lanka and China. The expertise is easily off-shored. In India, over 100,000 new IT graduates a year are ready to support an off-shored IT industry. Companies like Microsoft, Hewlett-Packard and Siemens have well established software development operations in India. Why are we not co-operating more with the Indian IT industry?

Globalisation is a big issue for information systems as well.

It doesn't help our students if any available jobs are being done in India at a tenth of the cost.

But what kinds of jobs? The Java programming, the call centre operations. Keith Mander wrote a reply to McBride's article on the BCS web site.

Who's Keith Mander?

Only the head of the Committee of heads of computing. He recognises the problems in computer science and says what we need is more of an emphasis on design. What worries me is not the coding jobs going to India, but the design and innovation being offshored. We're got to teach more design get better at innovation. If we don't get better at innovation in information systems we're in deep trouble.

I reckon that will need a change in the academic process. Conference refereeing and journal refereeing selects against off-the-wall or different ideas. The result is everything is the same. We all study, and write about the same thing because it will be acceptable to our peers. Anything unusual gets thrown out.

So where does that leave computing departments in universities? The old generation needs to look to a new generation, to new approaches. The focus is moving away from system construction. The jobs are in the application of technology. There is a need to be closer to the application, closer to the user, to replace a reductionist, convergent discipline with a complex, divergent discipline. The complexity of embedded systems, of modern computing applications requires a different way of thinking. A reductionist, programming mindset does not adapt well to uncertainty, emergent behaviour, the unexpected and the study of the whole.

Isn't he open to the accusation of not being scientific?

I guess so, but in the computing community we're really engineers we look for certainty, for measurable answers, for business solutions that can be encoded in a precise, even provable computer system.

I got a suspicion he's into complexity sciences, complex adaptive systems and all that. I don't go with that. Management is about control, removing chaos, simplifying, producing manageable organisations. That's what information systems or at least computer systems are for. If we're not careful we'll be talking about a New Kind of Science next.

You've lost me

You know Stephen Wolfram?

Who?

Forget it.

Relationship is important. The new computing discipline will really be an inter-discipline, connecting with other spheres, working with diverse scientific and artistic departments to create new ideas. Its strength and value will be in its relationships. There is a need for innovation, for creativity, for divergent thinking which pulls in ideas from many sources and connects them in different ways. The new computing department will be the department of interdisciplinary studies, drawing ideas from biology, design, history, medicine and contributing a rich computing foundation to those disciplines. It will be looking outwards rather than inwards, concerned to address the vast landscapes of computing application.

Here I'm with him. Being interdisciplinary is very important for information systems. We take ideas from other disciplines we inform other disciplines. We deal with applications.

Rather like applied computing?

So how many computer science departments will exist in 30 years time? Perhaps a few will support the elite luxury liners. Most will have given way to interdisciplinary study departments, and computing service departments, producing innovative graduates who can corral and manage the IT resources organisations need.

Our question is: where does this leave information systems?

Marginalised I think. A foreign body in computing departments and total absorbed and forgotten in business schools. After all isn't computing and information systems just part of the furniture, part of the decoration.

That's a worry in industry. I read somewhere that people were questioning whether companies really needed a CIO.

Hmph. You're swallowing the "Does IT Matter" argument. If IT is a utility then why treat it any differently to electricity, gas, water. I don't buy that. You focus too much on the technology – networks, database capacity – it's the information, the knowledge that matters and how that's used.

Computer science are old, stale and increasing irrelevant. Curricula needs to be vocational, and divergent, widening the computing student's view of the world, not creating a sterile bubble, closed off from the wider issues in the world, and from the networking, the integration, the global reach of computers. There is a need for a drastic rethinking of what the discipline is about. There is a need for new curricula which represents a real paradigm shift, not just a move from keyboards to pen computing.

In information systems we spend too much time thinking: what's the discipline about.

Perhaps we need some practical action to establish new curricula and new industry involvement.

Hmm. Or perhaps McBride's wrong and it's just a lot of hot air. Anyway, my practical action is another UML tutorial.

And I need some lunch.

Are you going to Manchester this year?

What the UKAIS conference?

I don't know. John says there's no money in the budget for conferences what with the drop in student numbers, but I might be able to fiddle it.