

Rediscovering the Passion,
Beauty, Joy and Awe:
Making Computing Fun Again

Andrew McGettrick, Eric Roberts,
Dan Garcia, Chris Stephenson

Regarding the title ...

- Dangerously ambitious ...
- In discussing these attributes of beauty, wonder, awe, etc. we have a duty (probably) to address issues such as the elegance and the mystery of Computing from our perspectives

Comment from SIGCSE 07

- Grady Booch in his key-note address at SIGCSE 07 spoke about the need to articulate the ‘wonder and the awe’ of computing
- This session is an opportunity for us to address that challenge
- Everyone will have their own perspective; the four of us will offer our perspectives
- *Importantly* we will leave time for other views to be aired

Grady's perspective?

- We followed up on last year's event and had a conversation with him
- What emerged as his 'wonder and awe'
 - The breadth of the applications of computing
 - It is a privilege to work in an area where we can change lives

Going further ...

- Given the rate of change of technology and the increased miniaturization, the range of applications keeps changing
- Each new generation is faced with new challenges and new opportunities

Links with Innovation

- These observations point to the fact that there is an inevitable link between computing and innovation ... the key to prosperity, competitiveness, etc.
- Innovation for all, not just a small section of the community
- Innovation across a huge range of application areas .. science, engineering, education, entertainment, transport, etc.

Elegance - some suggestions

- Simple formal grammar definition of language and semantics - finite describing the infinite - with recursive descent based compiler
- Public key encryption
- Design patterns
- Visualization - aspects thereof
- The notion of undecidability
- SQL, Prolog, RISC, the Internet

Mystery of computing?

- Personal, of course
- What are the basic principles and concepts underlying the discipline?
 - Peter Denning has made progress here
- How can we simplify perceptions of the discipline (and so curricula) and make it more attractive to different audiences (young and not so young)?
- What are the best and most effective ways of teaching it?

And more ... suggestions

- Reading programs
 - How important is this and how can we teach it effectively
 - to stimulate (e.g. teach about fractals)
 - to teach good practice (e.g. avoid security holes)
- Role of *simple* formalisms
 - Role of proof typified by link between induction and recursion
 - Through reuse (of programs and accompanying proofs) building environments to simplify programming

And even more ...

- What are the important elements of the idea of *computational thinking* and how do we promote these effectively?
- Can we find unifying ideas that stretch the imagination, e.g from Jeannette Wing, seeing programs as data and data as programs?

Your thoughts?