Solving Problems with Tools and Collaboration

LIKES Workshop
Santa Clara University
November 30, 2007

Lawrence A. Rowe
FX Palo Alto Laboratories

Outline

Organize our thinking about the future

Digital literacy in the 21st century

Requirements for a computer science education

Living and working in the future

Organize Our Thinking

Digital Literacy
- What should a well-educated citizen know about computing and information technology?

Computer Science Education
- What should a well-educated computer science bachelor's graduate know about computing and information technology?

Working in the 21st Century
- What skills are important for members of the workforce?

Digital Literacy in the Past

1950's-60's: "Batch Processing Era"
- Understand computer capabilities
  Financial/accounting, election predictions, rocket flight trajectories, etc.
- Work with experts who program and operate a computer

1970's-80's: "Interactive Computing Era"
- Use computers and applications
  Word processing, spreadsheets, desktop publishing, personal accounting, etc.
- Know how to operate a PC

1990's-2000's: "Internet Era"
- Understand networks of computers
- Use remote computers and applications to communicate
  Web, email, eCommerce, banking, IM, blogs, social networks, telephony, etc.
- New media entertainment
  Audio, video, and massively multiplayer online games

An Anecdote about the Future

Kids view the world differently because it is all they know

Never underestimate people's ability to learn and use tools

Understanding Media and Tools

Before the computer and Internet…
- Printing press/copier – publish material for others to read
- Radio/TV – broadcast audio/video for others to listen and watch
- Typewriter – allows individual to produce readable papers
- Telephone – allows individual to talk to someone at a distance

After the computer and Internet…
- Widely available tools for creating and publishing content
  As the tools got better, more people could do it (e.g., typesetting, photography, drawings, videos, etc.)
- Dramatic reduction of cost to communicate with people at a distance
  More than just audio, now includes video, images, shared experiences, etc.

Everyone can publish and broadcast

Tools Amplify our Abilities

Computer enables development of tools

- Spreadsheets, word processing, order processing systems, on-line airline/travel scheduling systems, etc.

What tools are needed?

- Better tools to author content
  Web pages, videos, interactive multimedia documents, games, etc.
- Better tools to work with remote people
  Face-to-face interaction still better than remote audio/video conference
- Better tools to discover knowledge
  Learn things that you want to know about
- Better tools to solve problems
  Diagnose a situation and change state to achieve a goal
- Better tools to customize my environment
  Allow me to control information/interaction presented to me

Tool Characteristics

WYSIWYG interfaces

- Direct manipulation of visual representation of object being manipulated

Simplify work required by people

- Abstract operations that represent task to be accomplished

Automate as much as possible

- Trade computation for human manipulation and memory
- Semi-automate if necessary

Assumes tool builder can anticipate all actions user wants to perform

Alternative: Programming!

Vast majority of computer users do not program

- Use programs to author content or complete a task, not automate a repetitive task
- Even kids do not program – use computer as a tool for entertainment

Some people do limited programming

- Examples: spreadsheet scripting, report writers, macro recorders, etc.
- Equipment allows user to "program behaviors" (e.g., start dishwasher at 2 AM, water zone 2 once every other day for 10 minutes, etc.)
- Customizing tool to your environment

Need higher level tools – "automatic programming"

- Common abstractions plus algorithms to fill-in details or find re-usable code fragments
- Re-examine research on
  programming-by-demonstration, automatic program creation (theorem proving), tools for finding useful code/libraries and incorporating them into your application, etc.

Need more training on basic programming concepts

- Assignment, iteration, procedure calls, parameter passing, map functions, etc.

Digital Literacy for the 21st Century

Understand and use Internet

Use tools to author content and solve problems

Automate repetitive tasks through programming

Customize on-line environment to improve productivity

Outline

Organize our thinking about the future

Digital literacy in the 21st century

Requirements for a computer science education

Living and working in the future

CS Education in the Past

1950's-60's: "Batch Processing Era"

- Computer architecture, low-level programming, operating systems, data structures, compilers, formal language theory, algorithm analysis, etc.
  Fortran, Basic, Cobol, Lisp(?), IBM mainframes

1970's-80's: "Interactive Computing Era"

- Computer architecture design, object-oriented programming, OS, DBMS, graphics, artificial intelligence, algorithm design and analysis, compilers, software engineering, etc.
  C/C++, Pascal, Fortran, Lisp/Scheme, minicomputers and PC's

1990's-2000's: "Internet Era"

- OS, computer networks, scripting languages, programming, user-interfaces, graphics, algorithm design and analysis, web page authoring and services, DBMS, AI, software engineering, distributed programming, cluster computing, etc.
  Java/C#, Scheme/Lisp, Perl/Python/Ruby, Javascript/PHP, Ruby-on-Rails/.net2, PC's, clusters, and networked computers

Common Features

Learn fundamental principles
- Operating systems: resource management, interprocess communication, file systems, etc.
- Computer networks: communication technologies, protocol architectures, network operations, etc.
- Algorithm design and analysis

Learn common application development paradigm
- Programming, debugging, and software engineering

Learn several languages and systems
- Changed over time to higher level languages
- Changed from single-user batch processing to multiple-user interactive applications

Analogy with the Past - Electricity

Early manufacturing companies located close to transportation hub and power source
- Typically near flowing water and navigable river or ocean

Development of electricity and motors meant companies could be located elsewhere
- Various types of motors (e.g., steam, sterling, and electrical) lead to different requirements for power

Companies produced own electricity until early 1900's

Electric power grid first proposed by Tesla in 1888
- Common generation plants provided power that could be transmitted to local distribution systems
- Widely deployed world-wide between 1900-1930

***KEY OBSERVATION**: Privately owned equipment replaced by shared resource*

How Computing is Changing

Computing as a utility – analogy with electricity
- Hardware moved to a data center co-located with network centers
- Purchase resources rather than owning hardware
- Applications moving into the cloud

Everyone will program
- Remember: reading, arithmetic, and telephone operation was performed by experts until it was added to common educational curriculum

Open source and application program interfaces
- Open source: companies sell service
- Open API's: create new apps and services by interfacing to other applications and services => mashups!

Programming teams geographically distributed
- Low-cost communication traded for high-cost labor
- Changing nature of work teams

Aside: "Why Pay for Open Source?"

Pay money to save time
- I would gladly pay you for an installable software package rather than having to compile and build it myself
- I would gladly pay you to manage my computing equipment or applications rather than doing it myself or hiring someone to do it

Contribute time to save money
- It will take more time, but I would rather use an open source package and pay you for service than pay the much higher cost to a company selling a closed-source solution
- I will fix bugs in the software to insure that I can solve my specific problem without having to write the entire package

CS Education

Digital literacy
- Use Internet, author content with tools, program, and customize on-line work environment
- Learn architecture and operation of Internet systems and apps

Continue to learn fundamental concepts
- Resource management, distributed systems, programming, software engineering, etc.

Algorithm design
- Design algorithms to solve problems and implement apps & tools

Tool building
- Design and implement tools used by others

Problem solving
- Apply scientific method and other problem solving techniques
- Use end-to-end or systems optimization to improve products and operations

CS Education (cont.)

Working in teams - collaboration
- Learn about tools for working with distributed teams
- Learn best practices for working with different people from different cultures and in different time zones

Business and economics
- Must understand structure and operation of a business
  Strategy, operations, accounting, personnel, sales & marketing, etc.
- Introduction to roles different people play in the marketplace
  Developers, technical support, evangilists, public relations, analysts, management, venture capitalists, investment community, etc.

Outline

Organize our thinking about the future

Digital literacy in the 21st century

Requirements for a computer science education

Living and working in the future

Liberal Arts Education for CS Graduates

Critical thinking

- Question hypotheses and solutions
- Observe, experiment, and build models to predict and possibly control organisms, nature, or behaviors

Learn to work in geographically distributed, multicultural teams

- Many companies geographically distributed
- Learn cultures and languages to support team work
- Learn about tools for distributed collaboration

  Email, IM, newsgroups, blogs, wikis, video conferencing, databases, shared documents, etc.

Learn to publish and broadcast

Summary: Big Ideas

Digital literacy means…

- Use the Internet, software tools and apps
- Program to improve productivity

CS education requires…

- Understand fundamental principles of computing
- Learn architecture and operation of Internet
- Build tools and solve problems

Modern workforce needs…

- Critical thinking skills
- Problem solving skills
- Ability to work in a geographically distributed, multicultural team