Why can’t smart people figure out what to do about computing education?

Russell Shackelford
ACM Education Board (chair)
Joint Task Force on Computing Curricula 2004 (chair)

http://www.acm.org/education/
The critical role of computing ed

- Computing is the infrastructure by which we:
  - Communicate
  - Work
  - Manage our affairs

- Computing has redefined how we do:
  - Science
  - Engineering
  - Business

- Computing has become fundamental for those who will participate in modern society
Why don’t we have it right?

- Computing is new
  - It doesn’t have a time honored role
- Computing is poorly understood by many
  - What is it *really*?
- People are vulnerable to wishful thinking
  - Cheap and easy? Substantive and hard?
- Computing is evolving
  - The landscape of computing changes
  - We’re trying to hit a moving target
Evolution of computing degrees

Pre-1990s:

EE  CS  IS
Evolution of computing degrees

- **Explosion of computing in the 1990s:**
  - Changed the world
  - Changed the computing education world

- **The “CC 2001” curriculum report:**
  - Provided curriculum guidelines for CS
  - Called for additional reports for other computing disciplines
  - Called for an “Overview Volume” to summarize commonalities and differences among them
How computing education changed

*Computing has become a family of disciplines*

- Before the 1990s:
  - *Computer Science* on the tech side
  - *Information Systems* on the business side

- During the 1990s:
  - *Computer Engineering* became a strong discipline
  - *Software Engineering* became a strong area within CS
  - *Software Engineering* began to form its own identity
  - *Information Technology* programs began to emerge
Computing degree programs

Pre-1990s:

EE  CS  IS

Post-1990s:

EE  CE  SE  CS  IT  IS
Computing degree programs

Pre-1990s:
- EE
- CS
  - Hardware
  - Software
- IS
  - Business

Post-1990s:
- EE
- CE
- SE
- CS
- IT
- IS
Computing degree programs

Pre-1990s:

EE  
CS  
EE  
IS  
Hardware  Software  Business

Post-1990s:

EE  CE  SE  CS  IT  IS  Hardware
Computing degree programs

Pre-1990s:

EE  CS  IS

Hardware  Software  Business

Post-1990s:

EE  CE  SE  CS  IT  IS

Software

Computing degree programs

Pre-1990s:

EE  CS  IS

Hardware  Software  Business

Post-1990s:

EE  CE  SE  CS  IT  IS

Software
Computing degree programs

Pre-1990s:
- EE
- CS
  - Hardware
  - Software
- IS
  - Business

Post-1990s:
- EE
- CE
- SE
- CS
- IT
- IS
  - Organizational Needs
The difference between IT and IS

*Both focus on using Information Technology*

- Information Systems programs:
  - Focus on the *Information* side of *IT*

- Information Technology programs:
  - Focus on the *Technology* side of *IT*
Computing degree programs

Pre-1990s:

- EE (Hardware)
- CS (Software)
- IS (Business)

Post-1990s:

- EE (Hardware)
- CE (Software)
- SE (Software)
- CS (Organizational Needs)
- IT (Organizational Needs)
- IS (Organizational Needs)
College curriculum reports

- **Computer Science**  CC2001 (CS2001)
- **Information Systems**  IS2002
- **Software Engineering**  SE2004
- **Computer Engineering**  CE2004
- **Information Technology**  IT2005
- **The Overview Volume**  CC2004

- Based on the *Body of Knowledge* from each
- Report on commonalities and differences
- A users’ guide to the computing disciplines
- A larger project to create a map of computing
Growing diversity in computing

*The diversity is localized*

- There’s always been a home for hardware
  - Was EE, has become CE
- There’s always been a home for business
  - *Information Systems*
- The increased diversity has occurred in the space between hardware and application
  - The space traditionally filled by CS programs
Growing diversity in computing

- **Root issue:**
  - CS used to be the only game in town
  - Society is demanding more & different things

- **Long-term disconnect between:**
  - What many CS faculty want to teach
  - What many CS students want to learn

- **Students now have meaningful choices**
New choices in computing degrees

- **Computer Science:**
  - Has tradition and pedigree
  - Provides a solid, deep foundation

- **Software Engineering:**
  - For the software professional
  - For most, still an area within CS (Future?)

- **Information Technology:**
  - For the IT practitioner
  - A mix of tech-issues and people-issues
  - The new kid on the block

What portfolio of degree programs will best serve:
- Our students?
- Our community?
- Our future?
Characterizing the disciplines

- *How to convey what each one is about:*
  - For faculty?
  - For administrators?
  - For prospective students?
  - For their parents and guidance counselors?

- *Must provide:*
  - Sketches and snapshots
  - Not tomes
Characterizing the disciplines

- **Sketches and snapshots:**
  - Brief descriptive prose (< 1 p.)
  - Chart of relative emphasis of study
  - Chart of relative capabilities of graduates
  - Graphics showing “occupied space”

- **Criteria:**
  - The characterization of a discipline *must be acceptable* to those from the other disciplines
Characterizing the disciplines

- Sketches and snapshots:
  - Brief descriptive prose (< 1 p.)
  - Chart of relative emphasis of study
  - Chart of relative capabilities of graduates
  - Graphics showing “occupied space”

- Criteria:
  - The characterization of a discipline must be acceptable to those from the other disciplines
Prose: Computer Science

Computer science spans a wide range, from theoretical foundations to cutting-edge developments in robotics, computer vision, intelligent systems, and bioinformatics. The work of computer scientists falls into three areas:

• **Designing and implementing software.** They are assigned key programming jobs and help keep other programmers aware of new approaches.

• **Developing effective ways to solve computing problems.** Computer scientists develop the best possible ways to store data in databases, send data over networks, and display complex images. Study of theory allows them to determine what performance is possible, and study of algorithms lets them develop new problem-solving approaches for better performance.

• **Devising new ways to use computers.** Progress in CS areas of networks database, and human-computer interaction enabled the World-Wide-Web which changed the world. They are now working to make robots be practical aides, databases create new knowledge, and computers do new things.

Computer science spans the range from theory to programming and is the least-specialized of the computing disciplines. Other disciplines can produce graduates better prepared for specific jobs, while computer science offers a foundation that permits graduates to adapt to new technologies and ideas.
Prose: Information Technology

*Information technology* specialists possess the necessary combination of knowledge and practical, hands-on expertise to take care of an organization’s information technology and the people who use it. Today, organizations of every kind are dependent on information technology. They need to have the appropriate systems in place. Those systems must work properly and be upgraded, maintained, and replaced as appropriate. The people of the organization require support from IT staff committed to solving whatever computer-related problems they might have. IT specialists meet these needs.

Their perspective on “Information Technology” emphasizes the technology itself. They assume responsibility for selecting appropriate hardware and software products, integrating those products with organizational needs and infrastructure, and installing, customizing and maintaining those resources. Examples of their responsibilities include installing and administering computer networks, managing e-mail systems, designing web pages, and developing multimedia resources and other digital media. They also devise and manage the plans for maintaining, upgrading, and replacing the organization’s IT resources to ensure they are adequate and up-to-date.
Characterizing the disciplines

- **Sketches and snapshots:**
  - Brief descriptive prose (< 1 p.)
  - Chart of relative emphasis of study
  - Chart of relative capabilities of graduates
  - Graphics showing “occupied space”

- **Criteria:**
  - The characterization of a discipline must be acceptable to those from the other disciplines
  - Adapting CC2001’s “Consensus Rule”

- **Will satisfy no one who’s on the inside.**
## Relative Emphases in Programs of Study

<table>
<thead>
<tr>
<th>Knowledge/Skill Area</th>
<th>CE</th>
<th>CS</th>
<th>IS</th>
<th>IT</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Fundamentals</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Algorithms and Complexity</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Computer Architecture &amp; Organization</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Operating Systems: Principles, Design</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Operating Systems: Use, Configuration</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Net-centric: Principles, Design</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Net-centric: Use and Configuration</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Theory of Programming Languages</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Human-Computer Interaction</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Graphics and Visualization</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Intelligent Systems (AI)</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Information Management (DB): Theory</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Information Management (DB): Practice</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Scientific computing (Numerical mthds)</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Organizational Theory</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge/Skill Area</td>
<td>Knowledge/Skill Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management of IS organization</td>
<td>e-Business</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision Theory</td>
<td>Security: Theory and Principles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizational Behavior</td>
<td>Security: Implementation and Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizational Change Management</td>
<td>Computer Systems Engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Legal/Professional/Ethics/Society</td>
<td>Embedded Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General Systems Theory</td>
<td>Circuits and Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Systems Development</td>
<td>Electronics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk Management (Project &amp; Safety)</td>
<td>Digital Logic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Management</td>
<td>Distributed Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis of Business Requirements</td>
<td>Digital Signal Processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering Foundations for Software</td>
<td>VLSI Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering Economics for Software</td>
<td>Hardware Testing and Fault Tolerance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Modeling and Analysis</td>
<td>Systems Administration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Design</td>
<td>Systems Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Verification and Validation</td>
<td>Digital Media Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Evolution (Maintenance)</td>
<td>Technical Support</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Process</td>
<td>Interpersonal Communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Quality</td>
<td>Mathematics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Characterizing the disciplines

- *Sketches and snapshots:*
  - Brief descriptive prose (< 1 p.)
  - Chart of relative emphasis of study
  - Chart of relative capabilities of graduates
  - Graphics showing “occupied space”

- *Criteria:*
  - The characterization of a discipline *must be acceptable* to those from the other disciplines
  - Adapting CC2001’s “Consensus Rule”

- *Will satisfy no one who’s on the inside.*
## Relative Performance Capability of Graduates

<table>
<thead>
<tr>
<th>Area</th>
<th>Performance Capability</th>
<th>CE</th>
<th>CS</th>
<th>IS</th>
<th>IT</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Programs</strong></td>
<td>Design an application program</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Implement an application program</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Use application program features well</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Train and support application users</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td><strong>Information Management</strong></td>
<td>Design a database program</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>(Database)</td>
<td>Use a database program well</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Implement information retrieval software</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Select database products</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Configure database products</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Manage databases</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Train and support database users</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td><strong>Programming</strong></td>
<td>Do small-scale programming</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Do large-scale programming</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Do systems programming</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Develop new software systems</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Area</td>
<td>Performance Capability</td>
<td>Area</td>
<td>Performance Capability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------</td>
<td>------------------------------</td>
<td>------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Info Systems</td>
<td>Determine info system requirements</td>
<td>Algorithms</td>
<td>Prove theoretical results</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design an information system</td>
<td></td>
<td>Develop ways to attack problems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implement an information system</td>
<td></td>
<td>Develop proof-of-concept software</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manage an information system</td>
<td></td>
<td>Determine if better solutions possible</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Infra-structure</td>
<td>Manage websites</td>
<td>Intelligent Systems (AI)</td>
<td>Design automated reasoning systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create e-commerce software</td>
<td></td>
<td>Implement automated reasoning syst’s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create multimedia systems</td>
<td></td>
<td>Implement intelligent systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Develop health-related info system</td>
<td>Network &amp; Communications</td>
<td>Design network configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create e-learning software</td>
<td></td>
<td>Select network components</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Develop business applications</td>
<td></td>
<td>Install a computer network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluate new forms of search engine</td>
<td></td>
<td>Manage computer networks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer-based Systems</td>
<td>Design embedded systems</td>
<td>IT Resource Planning</td>
<td>Implement communications software</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implement embedded systems</td>
<td></td>
<td>Manage communications resources</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design computer peripherals</td>
<td></td>
<td>Implement mobile computing app’s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implement computer peripherals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design complex sensor system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implement complex sensor system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design a chip</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design a computer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Characterizing the disciplines

- Sketches and snapshots:
  - Brief descriptive prose (< 1 p.)
  - Chart of relative emphasis of study
  - Chart of relative capabilities of graduates
  - Graphics showing “occupied space”

- Criteria:
  - The characterization of a discipline must be acceptable to those from the other disciplines
  - Adapting CC2001’s “Consensus Rule”
  - Will satisfy no one who’s on the inside.
Simple Snapshots

<table>
<thead>
<tr>
<th>Organizational Issues &amp; Information Systems</th>
<th>Application Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Methods and Technologies</td>
<td></td>
</tr>
<tr>
<td>Systems Infrastructure</td>
<td></td>
</tr>
<tr>
<td>Computer Hardware and Architecture</td>
<td></td>
</tr>
</tbody>
</table>

DEVELOPMENT

More Theoretical  More Applied

Application Deployment Configuration
Snapshot: Computer Engineering
Snapshot: Computer Science

Computer Science (CS) encompasses a wide range of topics including:

- **Organizational Issues & Information Systems**
- **Application Technologies**
- **Software Methods and Technologies**
- **Systems Infrastructure**
- **Computer Hardware and Architecture**

A spectrum illustrates the development process, moving from more theoretical to more applied.

The categories are placed on a grid, with theoretical concepts on the left and application deployment on the right, showing the progression from innovation to configuration.

Key areas include:
- Theory
- Principles
- Innovation
- Development
- Deployment

The diagram highlights the balance between theoretical and applied aspects of computer science, emphasizing the importance of both theoretical foundations and practical applications in the field.
Snapshot: Information Systems

Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture

More Theoretical
More Applied

DEVELOPMENT

Theory Principles Innovation
Application Deployment Configuration

IS
Snapshot: Information Technology

- Organizational Issues & Information Systems
- Application Technologies
- Software Methods and Technologies
- Systems Infrastructure
- Computer Hardware and Architecture

IT

Theory Principles Innovation
DeVELOPMENT
More Theoretical
More Applied
Application Deployment Configuration
Snapshot: Software Engineering

- Organizational Issues & Information Systems
- Application Technologies
- Software Methods and Technologies
- Systems Infrastructure
- Computer Hardware and Architecture

SE

- Theory Principles Innovation
- DEVELOPMENT
- More Theoretical
- More Applied
- Application Deployment Configuration

- More Theoretical
- More Applied
The CC2004 Overview Volume

The Overview Report

- Chapter 1: Introduction
- Chapter 2: The computing disciplines
- Chapter 3: Degree programs and expectations of graduates
- Chapter 4: Institutional considerations
- Chapter 5: Guide to the curriculum reports

Audience: College faculty and administrators

Target length: 50 pp.
The CC2004 Overview Volume

The Users’ Guide

- Chapter 1: Introduction
- Chapter 2: The computing disciplines and degrees
- Chapter 3: The computing professions
- Chapter 4: Finding what’s best for you

Audience: Students, parents, teachers, and guidance counselors

Target length: 15 pp document
   6 pp brochure
The Two Overview Projects

- **Computing Curricula 2004 is:**
  - The smaller project
  - Focused on the *intersections*
  - Characterizing the *differences*

- **The Computing Ontology Project is:**
  - The larger project
  - Focused on the *union*
  - Characterizing the *problem space*
Simple snapshots are *too simple*
The Two Overview Projects

- Computing Curricula 2004 is a guide for:
  - College faculty and administrators
  - Students, parents, guidance counselors

- The Computing Ontology is a map for:
  - Curriculum revision
  - Discipline definition
  - Topic classification
  - Degree program accreditation
The Computing Ontology Project

**Motivation:**

- Keeping the computing-related disciplines as a single voice to maintain strength
- Addressing the challenges of updated curriculum recommendations
- Supporting curriculum development for creative new types of programs
- Easing the path toward accreditation for non-standard programs
The Computing Ontology Project

Goals:

- Produce an interactive structure for representation and exploration of the unified body of knowledge of all of the computing-related disciplines
- Support development of academic programs
- Assist with program validation
- Illuminate content relationships among disciplines
- Support development of interdisciplinary programs
The Computing Ontology Project

**Vision:**

- An *interactive representation* of all of the parts of computing-related fields
- A *visual display* of where a given program or curriculum report fits within the overall field
- A *framework* designed to handle updates as needed
The Computing Ontology Project

**Challenge:**

- Have long lists of “knowledge units” for each of five computing disciplines (CS, CE, IS, IT, SE).
- How can we show the superset of knowledge units and their interdependencies and overlaps?
- How can we to provide clarity when a given term is used to mean different things from one discipline to another?
ACM’s Role in K-12

- Historically, a society of CS researchers and academics
- Is now redefining its focus in two ways:
  - Increased focus on computing professionals
  - Expanding its educational focus beyond college
- Support for CSTA
- Creating an *Education Council*
  - Broader constituency than its *Education Board*
  - Creating a forum for communication, sharing, and forming partnerships among educators
  - A place for K-12 to be “at the table” with others
ACM’s view of K-12 education

*K-12 Computing Educators:*

- Are in position to play a *crucial role*
- Have been *ignored* and *discounted*
- Are facing an *uphill battle*
- Are trying to accomplish things that are *more important for society than society seems to realize*
A key challenge

Develop a united voice, one message

- As important for K-12 as for others
- Find common ground between CS and IT
  - *Neither one* is sufficient by itself
- Don’t permit anyone to drive a wedge
  - *Not* one discipline *versus* the other
  - *Both* are key elements of *computing education*
  - *Don’t* repeat the mistakes college faculty made!
  - Work this out *as allies behind closed doors*
The status pyramid

Height = faculty status & available resources

Shaded area = faculty workload & contact w/ students

Elite schools

Large universities

Public 4-year colleges

Public 2-year colleges

K-12 education
Why can’t smart people figure out what to do about computing education?

Russell Shackelford
ACM Education Board (chair)
Joint Task Force on Computing Curricula 2004 (chair)

http://www.acm.org/education/