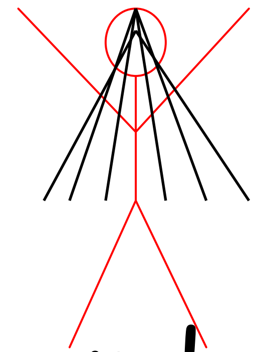


Looking Ahead in Computing and at NSF

Jeannette M. Wing

Assistant Director
Computer and Information Science and Engineering Directorate

and

President's Professor of Computer Science
Carnegie Mellon University

European Computer Science Summit 2007, Berlin, Germany
8 October 2007

Outline

- Looking Far Out
  - Vision: Making Computational Thinking Commonplace
  - Deep Questions for Computing
- Looking Near Term: New Initiative
  - CDI: Cyber-enabled Discovery and Innovation
- Looking Ahead: Themes
  - Math $\leftrightarrow$ Computing
  - Parallel and Distributed Thinking
  - Software for Complex Systems
  - Human in the Loop
  - Understanding the Brain

Looking Far Out:
My Vision for Computing

My Grand Vision for the Field

- Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century.
  - Just like reading, writing, and arithmetic.
  - Imagine every child knowing how to think like a computer scientist!
  - Incestuous: Computing and computers will enable the spread of computational thinking.
  - In research: scientists, engineers, …, historians, artists
  - In education: K-12 students and teachers, undergrads, …

J.M. Wing, "Computational Thinking," *CACM* Viewpoint, March 2006, pp. 33-35.
Paper off CISE AC website; paper and talks off http://www.cs.cmu.edu/~wing/

The Two A's of Computational Thinking

- Abstraction
  - C.T. is choosing the right abstractions
  - C.T. is operating in terms of multiple layers of abstraction simultaneously
  - C.T. is defining the relationships the between layers
- Automation
  - C.T. is thinking in terms of mechanizing our abstractions, abstraction layers, and their relationships
    - Mechanization is possible due to precise and exacting notations and models
  - There is some "machine" below (human or computer, virtual or physical)
- They give us the ability and audacity to scale.

Examples of Computational Thinking

- How difficult is this problem and how best can I solve it?
  - Theoretical computer science gives precise meaning to these and related questions and their answers.
- C.T. is thinking recursively.
- C.T. is reformulating a seemingly difficult problem into one which we know how to solve.
  - Reduction, embedding, transformation, simulation
- C.T. is choosing an appropriate representation or modeling the relevant aspects of a problem to make it tractable.
- C.T. is interpreting code as data and data as code.
- C.T. is using abstraction and decomposition in tackling a large complex task.
- C.T. is judging a system's design for its simplicity and elegance.
- C.T. is type checking, as a generalization of dimensional analysis.
- C.T. is prevention, detection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction.
- C.T. is modularizing something in anticipation of multiple users and prefetching and caching in anticipation of future use.
- C.T. is calling gridlock deadlock and avoiding race conditions when synchronizing meetings.
- C.T. is using the difficulty of solving hard AI problems to foil computing agents.
- C.T. is taking an approach to solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science.

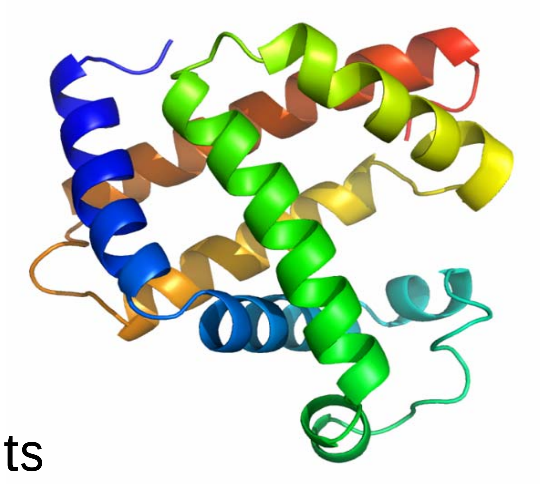
Please tell me your favorite examples of computational thinking!

Simple Daily Examples

- Looking up a name in an alphabetically sorted list
  - Linear: start at the top
  - Binary search: start in the middle
- Standing in line at a bank, supermarket, customs & immigration
  - Performance analysis of task scheduling
- Putting things in your child's knapsack for the day
  - Pre-fetching and caching
- Taking your kids to soccer, gymnastics, and swim practice
  - Traveling salesman (with more constraints)
- Cooking a gourmet meal
  - Parallel processing: You don't want the meat to get cold while you're cooking the vegetables.
- Cleaning out your garage
  - Keeping only what you need vs. throwing out stuff when you run out of space.
- Storing away your child's Lego pieces scattered on the LR floor
  - Using hashing (e.g., by shape, by color)
- Doing laundry, getting food at a buffet
  - Pipelining the wash, dry, and iron stages; plates, salad, entrée, dessert stations
- Even in grade school, we learn algorithms (long division, factoring, GCD, …) and abstract data types (sets, tables, …).

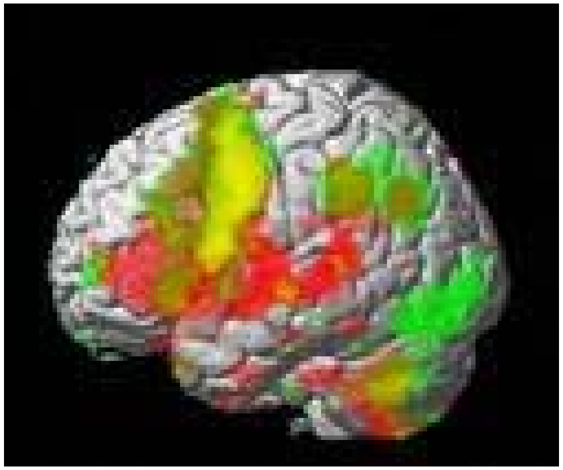
Research Implications

CT in Other Sciences, Math, and Engineering



Biology

- Shotgun algorithm expedites sequencing of human genome
- DNA sequences are strings in a language
- Protein structures can be modeled as knots
- Protein kinetics can be modeled as computational processes
- Cells as a self-regulatory system are like electronic circuits



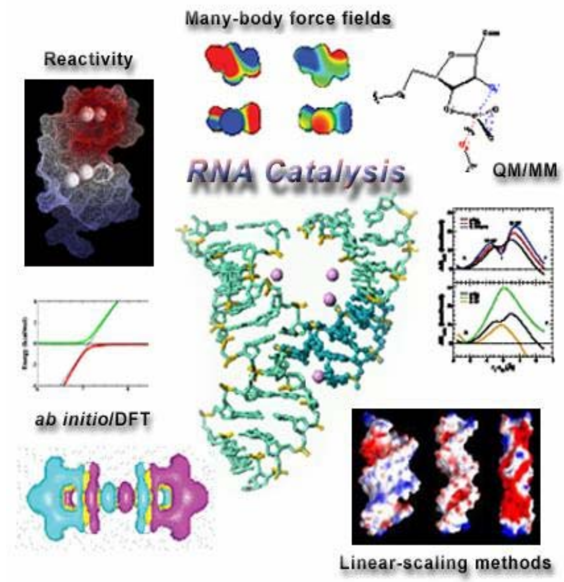
Brain Science

- Modeling the brain as a computer
- Vision as a feedback loop
- Analyzing fMRI data with machine learning

CT in Other Sciences, Math, and Engineering

Chemistry [Madden, Fellow of Royal Society of Edinburgh]

- Atomistic calculations are used to explore chemical phenomena
- Optimization and searching algorithms identify best chemicals for improving reaction conditions to improve yields

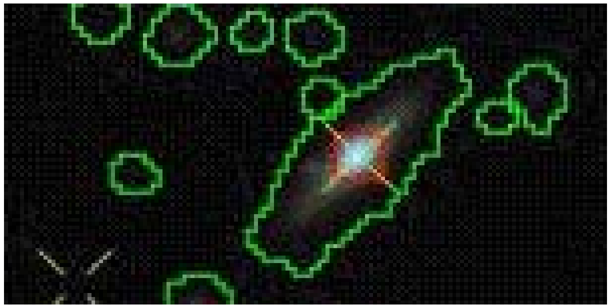


[York, Minnesota]

Geology

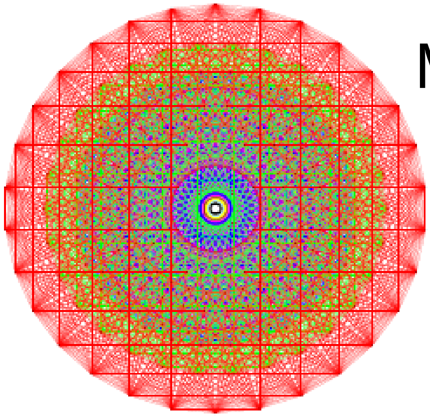
- Modeling the earth's surface to the sun, from the inner core to the surface
- Abstraction boundaries and hierarchies of complexity model the earth and our atmosphere

CT in Other Sciences, Math, and Engineering



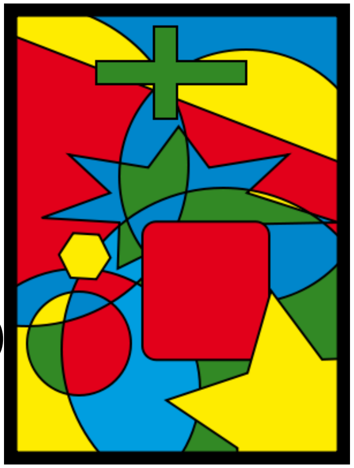
Astronomy

- Sloan Digital Sky Server brings a telescope to every child
- KD-trees help astronomers analyze very large multi-dimensional datasets



Mathematics

- Discovering E8 Lie Group:
  18 mathematicians, 4 years and 77 hours of supercomputer time (200 billion numbers).
  Profound implications for physics (string theory)
- Four-color theorem proof



Engineering (electrical, civil,mechanical, aero&astro, ...)

- Calculating higher order terms implies more precision, which implies reducing weight, waste, costs in fabrication
- Boeing 777 tested via computer simulation alone, not in a wind tunnel

CT for Society

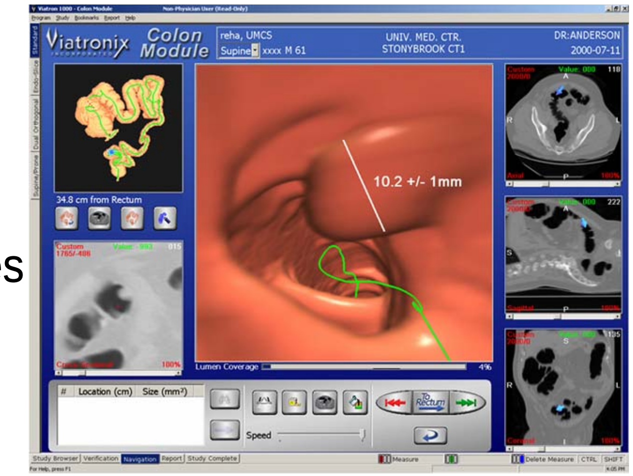
Economics

- Automated mechanism design underlies electronic commerce, e.g., ad placement, on-line auctions, kidney exchange
- MIT PhDs in CS are quants on Wall Street

Social Sciences

- Social networks explain phenomena such as MySpace, YouTube
- Statistical machine learning is used for recommendation and reputation services, e.g., Netflix, affinity card

CT for Society



Medicine

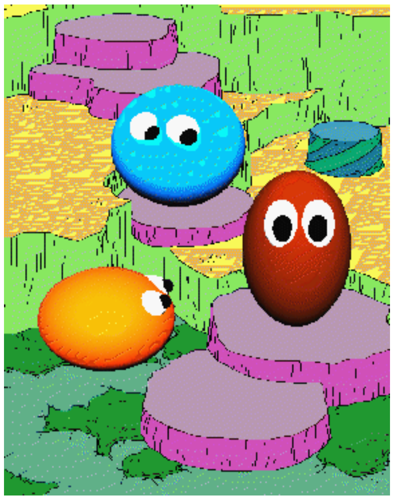
- Robotic surgery
- Electronic health records require privacy technologies
- Scientific visualization enables virtual colonoscopy

Law

- Stanford CL approaches include AI, temporal logic, state machines, process algebras, petri nets
- POIROT Project on fraud investigation is creating a detailed ontology of European law
- Sherlock Project on crime scene investigation

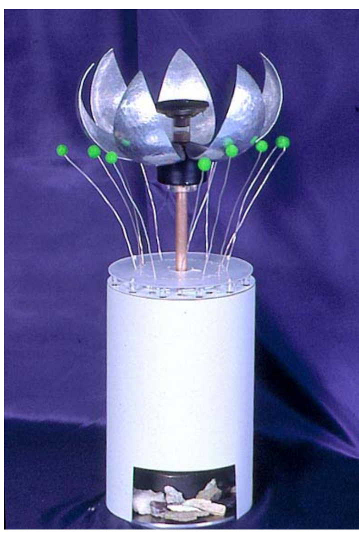
CT for Society

Entertainment



- Games
- Movies
  - Dreamworks uses HP data center to render *Shrek* and *Madagascar*
  - Lucas Films uses 2000-node data center produce *Pirates of the Caribbean.*

Arts



- Art (e.g., Robotticelli)
- Drama
- Music
- Photography

Sports

- Lance Armstrong's cycling computer tracks man and machine statistics
- Synergy Sports analyzes digital videos NBA games

Educational Implications

Pre-K to Grey

- K-6, 7-9, 10-12
- Undergraduate courses
  - Freshmen year
    - "Ways to Think Like a Computer Scientist" aka Principles of Computing
  - Upper-level courses
- Graduate-level courses
  - Computational arts and sciences
    - E.g., entertainment technology, computational linguistics, …, computational finance, …, computational biology, computational astrophysics
- Post-graduate
  - Executive and continuing education, senior citizens
  - Teachers, not just students

Looking Far Out:
Fundamental Questions

Deep Questions for Computer Science

- P = NP ?

- What is computable?
  - What is a computer?
    - Not just a PC anymore: The Internet, server farms, supercomputers, multi-cores, …, nano, bio, quantum, etc.
  - What is the power of computing, by machine and human together?

- What is intelligence?
  - Understanding the brain
  - What is the mind?

- What is information?
  - From nature to knowledge

- How can we build complex systems simply?
  - Can we build systems with simple designs, that are easy to understand, modify, and maintain, yet provide the rich complexity in functionality of systems that we enjoy today?
  - Is there a meaning of system complexity that spans the theory and practice of computing?

Two Messages for the General Public

- Intellectually challenging and engaging scientific problems in computer science remain to be understood and solved.
  - Limited only by our curiosity and creativity

- One can major in computer science and do anything.
  - Just like English, political science, or mathematics

Looking Ahead at NSF

Back to Basics : Transformative Research

- NSF is about basic science and engineering.
  - Preserve CISE core.
- It's all about good ideas and good people.
- It's about "high risk" long term impact.
  - Impact may be far in the future.
  - Impact is long-lasting (that is real science).
  - Impact can create new economies and change societal behavior.
  - Say "No" to incrementalism!
  - Promote new, emerging areas of computing.

CDI: Cyber-Enabled Discovery and Innovation

- Computational Thinking for scientists and engineers
- Paradigm shift
  - Yesterday: metal tools (transistors and wires)
  - Today: mental tools (abstractions and methods)
    - "Algorithms" is becoming a household word, e.g., NY Times, Forbes magazine, Harvard Business Review, …
- It's a partnership.
  - To advance BOTH computer science and the other science/engineering discipline.
  - Computer scientist says "I have an incomplete solution that might help you solve your problem." In working on X's problem, new computer science is invented and new science is discovered.
- FY08: $52M agency-wide, $20M CISE

CDI Dimensions

1. Extracting knowledge from data

2. Understanding complexity in natural, physical, and built systems

3. Virtual organizations

N.B. Feedback loop between 1 and 2.
N.B. 3 cuts across 1 and 2.

Looking Ahead: Themes

Theme: Math $\leftrightarrow$ Computing

What is computable?
What is information?

- Game Theory
  - Computational microeconomics
    - e.g., ad placement, on-line auctions, organ exchange
  - Networking
    - e.g., congestion control, adaptive wireless networks
  - Security
    - e.g., two-party games (adversary and administrator)
- Topology
  - Distributed systems, sensor nets
  - Robotics, e.g., self-configurable, robot arms, motion planning
  - Protein structure, e.g., knot theory
- Spectral Graph Analysis (algebraic graph theory meets linear algebra)
  - Image segmentation, e.g., medical, face recognition
  - Data clustering, e.g., data mining, market research, social networks
  - Scientific computing, e.g., energy and angular momentum of electrons; absorption spectrum of chemicals

Theme: Parallel and Distributed Thinking

What is computable?
What is a computer?
What is information?

- Trends
  - New kinds of "computers"
    - Now: The Internet, server farms, supercomputers, multi-cores, …
    - Tomorrow: nano, programmable matter, bio, quantum (?)
  - Industry is seeking help from academia
    - Economic argument (e.g., Intel's bottom line)
    - Competitiveness argument (skilled workforce needed)

- We need new models of computation to harness the different and differing capabilities of these "computers."
  - Programming models, parallel and distributed algorithms, programming languages, systems architecture, systems software, communication patterns and protocols, …, new applications

Theme: Software for Complex Systems

How can we build complex systems simply?

Trends

- Nature of tomorrow's systems
  - Dynamic, ever-changing, 24/7 reliability
  - Self-* (aware, diagnosing, healing, repairing, managing)
- Two important classes converging
  - Embedded
    - Networked architecture, e.g., sensor nets (see below)
    - Safety-critical apps, e.g., medical, automotive, aero&astro
    - Challenge: Reasoning about uncertainty, e.g., Human, Mother Nature, the Adversary
  - Pervasive and mobile
    - Focus on sensors and actuators, not just the devices and communication links
    - Prevalence of cell phones, iPods, RFIDs, …
    - Implications for HCI, embedded systems, sensor nets (see above)

➤ It's the software that effects this functional complexity. What new scientific and engineering principles, especially unique to software, are needed so that we can build and understand these systems?

Theme: Software for Complex Systems

How can we build complex systems simply?

Two Complementary Themes

1a. We need new advances in software foundations.

- What does "correctness" mean?
  - Factor in context of use, unpredictable environment, emergent properties, dynamism
- What are the desired properties of and metrics for both software (e.g., weak compositionality) and systems (e.g., power)?

1b. We need new engineering processes for creating such software-intensive systems.

- Traditional ones won't work.

2. Our approaches can inspire other science and engineering disciplines. E.g,. Geo: "The Earth is an Analogue Computer"; Bio: cellular- to human-level. We have the advantage of being able to build a virtual world and parameterize the Laws of Nature.

Theme: Human in the Loop

What is computable?
What is a computer?
What is intelligence?
Understanding complexity.

- Trends
  - Human (as a First-Class Computer) + Machine
    - What is computable, by human and computer together?
      - Human Computation, e.g., CAPTCHAs, The ESP Game
      - Formal Methods, e.g., interactive theorem provers
  - Human-Robot Interaction (HRI): Humanoid robotics, medical robotics, social robots, quality-of-life robots
    - Pervasive sensing and actuating in time and space (see previous slide)
    - AI "applications": Convergence of vision and graphics; advances in animation, speech, NLP, machine translation
  - Virtual Communities
    - Social networks, networks of brains
  - Computational and Creative Thinking
    - Computing and the Arts and Humanities
- We need to revisit fundamental questions of computing, when we consider the human in the loop.
  - Across directorate: IIS certainly, but we need new foundations (CCF) and systems (CNS) too.
  - Across agency: CISE, Eng, Bio, SBE

Theme: Understanding the Brain

What is Intelligence?
What is Information?

- Trends
  - Advances in computational neuroscience, cognitive science, social and behavioral sciences
  - Massive collection and processing of fMRI data
  - Interest by computer scientists in a Theory of Consciousness
- The Brain is a Computer, The Computer is a Brain ?
  - What computer science abstractions help us understand the brain?
  - What neuroscience can inform and push computer science? How does the brain process information that is different from the way a computer does?
- Across NSF (CISE, Bio, Eng, SBE)
- Inter-agency (NSF+NIH)

Education

- Push
  - Computational Thinking for everyone
    - Rethinking CS curricula (CPATH), outreach, diversity (BPC)
    - K-12
- Pull
  - Maths we drawn on
    - Probability and statistics
    - Game theory
    - Topology
  - Engineering foundations we drawn on
    - Control theory
- Push and Pull
  - Technology and Society efforts, e.g., $100 laptop, to push IT into Africa and other developing regions
  - Asia, Europe, Middle East seeking to elevate their higher educational system and research capacity

- Implications for training next generation researchers and practitioners
  - Interdisciplinary
  - Teams of collaborators, across time zones and cultures
  - International

Beyond NSF/CISE

- Cross-directorate programs, e.g., CDI
  - CISE + {MPS, Eng, Bio, Geo, SBE, OCI, Polar, …}
- Inter-agency programs
  - NSF/CISE + {NIH, DOE, …}
- Goal: Build communities.
  - NSF/CISE can spawn.
  - CCC can foster and sustain (we hope)
- International
  - NSF and EU?

Discussion Question

- Looking ahead, what would make sense for NSF and the EU/ECSS to do together?

Thank you!