Microsoft® Research

Faculty Summit

10 YEAR ANNIVERSARY

Microsoft Research

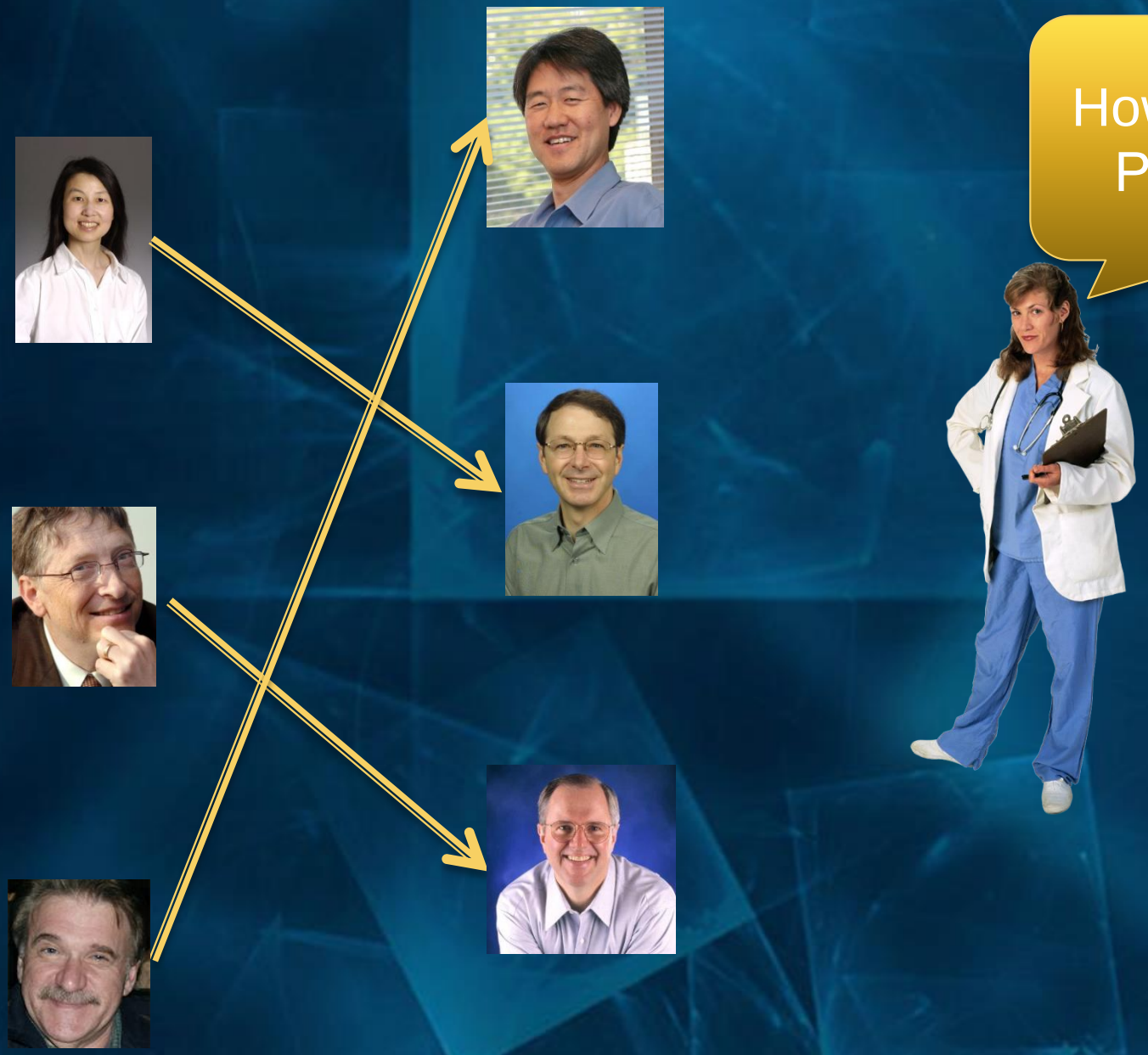
The Spread of Computational Thinking

Peter Lee
Professor and Head of Computer Science
Interim Director, Center for Computational Thinking
Carnegie Mellon University

Microsoft Research

"Computational thinking is a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science."

— *Jeannette Wing*



Fast algorithms for optimizing n-way organ exchange will be adopted by UNOS

The Necessity of CT

Microsoft Research

By thinking computationally (that is, abstracting the problem, understanding the consequences of scale, and applying algorithmic concepts), we extract simplicity from complexity and, in this case, save lives.

Microsoft Research

Center for Computational Thinking
Carnegie Mellon

Sponsored by Microsoft Research

- Research PROBEs
- Seminar Series
- Outreach and Symposia
- Resources
- People

Computer science is having a revolutionary impact on scientific research and discovery. Simply put, it is nearly impossible to do scholarly research in any scientific or engineering discipline without an ability to think computationally. The impact of computing extends far beyond science, however, affecting all aspects of our lives. To flourish in today's world, everyone needs computational thinking.

The mission of the Center for Computational Thinking is to advance computing research and advocate for the widespread use of computational thinking to improve people's lives. The Center accomplishes this by seeding research activities, seminars, and symposia that lead to vivid demonstrations of the value of computational thinking in diverse areas of human life.

What is computational thinking?

wordle.net

- Computational thinking is a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science. To flourish in today's world, computational thinking has to be a fundamental part of the way people think and understand the world.

- Computational thinking means creating and making use of different levels of **abstraction**, to understand and solve problems more effectively.

- Computational thinking means thinking **algorithmically** and with the ability to apply mathematical concepts such as **induction** to develop more efficient, fair, and secure solutions.

- Computational thinking means understanding the consequences of **scale**, not only for reasons of efficiency but also for economic and social reasons.

What's New

Computational Thinking Seminar Series:

Suguru Ishizaki
Associate Professor, *English Department, Carnegie Mellon*

Design

http://computationalthinking.org

People who Think Computationally Can…

Microsoft Research

- **Apply algorithmic concepts** to understand, explain, solve, and debug problems, processes, and interactions
- **Understand the consequences of scale** in terms of both engineering and societal impact
- Make **use of abstractions** at multiple levels, to control complexity and collaborate more effectively
- …

People who Think Computationally Can…

Microsoft Research

- **Apply algorithmic concepts** to understand, explain, solve, and debug problems, processes, and interactions
  - *recursion, divide-and-conquer, dynamic programming, …*
- **Understand the consequences of scale** in terms of both engineering and societal impact
  - *exponential growth, greedy vs global, Metcalfe's Law, …*
- Make **use of abstractions** at multiple levels, to control complexity and collaborate more effectively
  - *applied logic, languages, …*
- …

The Spread of CT

Microsoft Research

COMPUTER SCIENCE AND TELECOMMUNICATIONS BOARD

THE NATIONAL ACADEMIES
Advisers to the Nation on Science, Engineering, and Medicine

Workshop series: *Computational Thinking for Everyone*

National Science Foundation
DIRECTORATE FOR
Computer & Information Science & Engineering (CISE)

SEARCH
NSF Web Site

CISE Home | CISE Funding | CISE Awards | CISE Discoveries | CISE News | About CIS

Computer & Information Science & Engineering (CISE)

Cyber-Enabled Discovery and Innovation (CDI)

"…revolutionary science and engineering research outcomes made possible by innovations and advances in computational thinking…"

This Session

Microsoft Research

- A sample of research and education activities in the Center for Computational Thinking
- Called PROBEs, for PROBlem-oriented Explorations, each explores and/or demonstrates the value of computational thinking, in
  - drug discovery
  - parallel computing
  - music and the arts

Microsoft Research

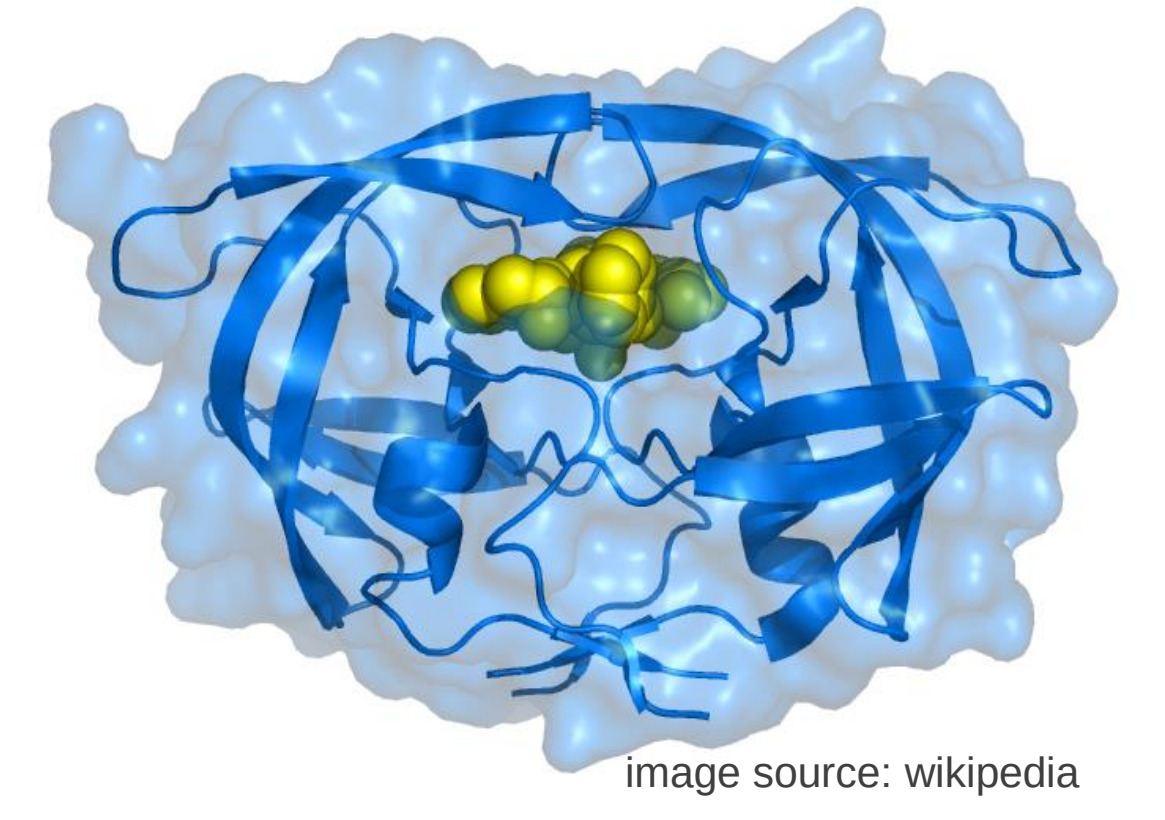
Computational Thinking for Drug Design

Christopher James Langmead
Department of Computer Science
Carnegie Mellon University

Drug Design and Resistance

Example: HIV-1 Protease (HIV PR)

- What is Drug Design?
  - Finding a **compound** that:
    - **Binds** to **target** molecule
    - **Inhibits function** of target
- What is Resistance?
  - A set of **mutations** to target that:
    - **Inhibits binding** of drug
    - **Preserves function** of target

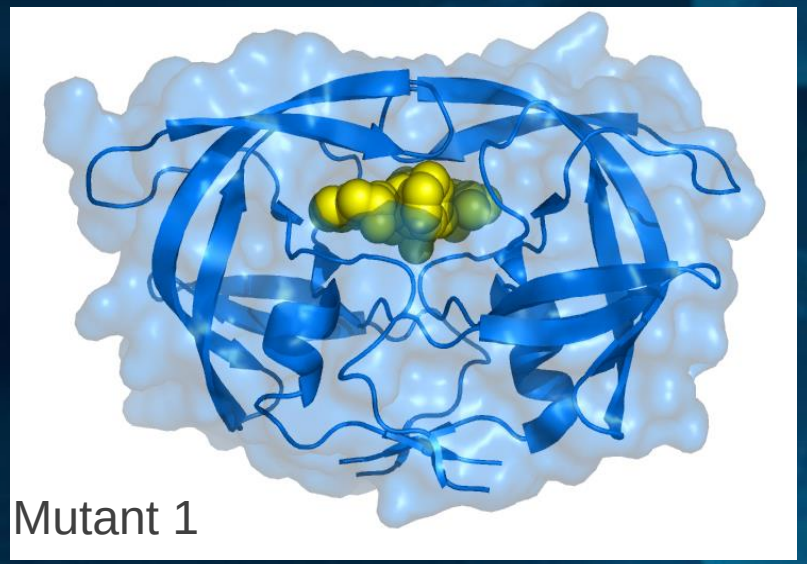


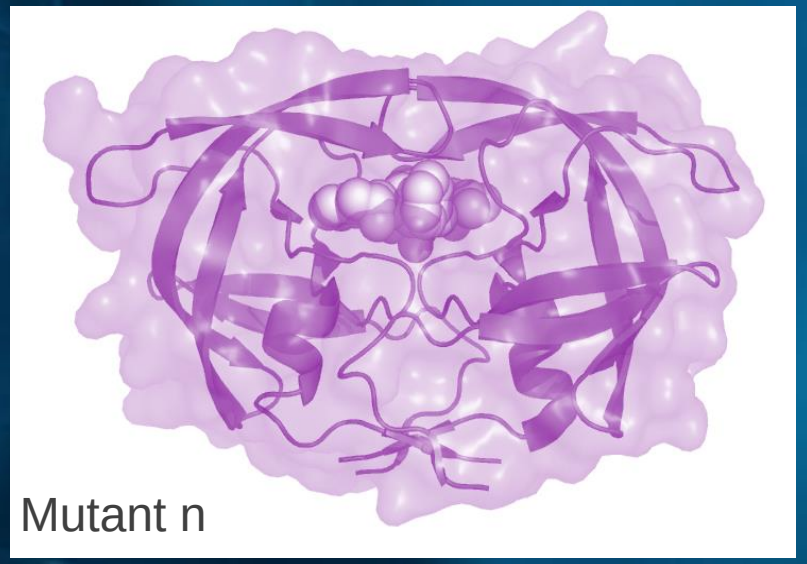
HIV PR and **drug**

Key Question:

Microsoft Research

- Can we design compounds that bind to, and inhibit function of target molecule --- ***and all viable mutants***?





Computational Thinking for Drug Design

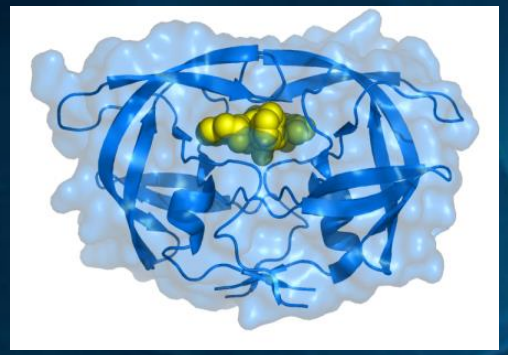
Abstraction #1

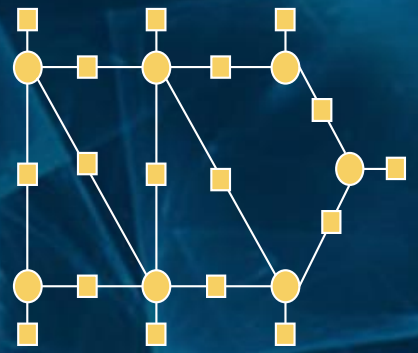
- 2-player Game: Pharma vs. Virus
  - A resistance-evading drug is a “checkmate”
- Moves:
  - Pharma moves by introducing/modifying compound
  - Virus moves by introducing mutations
  - Note: *Pharma can anticipate virus’ moves, but not vice-versa*
- Move Evaluator
  - Nature’s function: (changes in) Free Energy

Computational Thinking for Drug Design

Abstraction #2

- Computing free energies , probabilistic inference
  - Statistical Physics tells us how to derive free energies from the probability distribution over configuration space
  - We can efficiently represent these probability distributions using undirected probabilistic graphical models
  - Free energies are computed via (generalized) belief propagation (BP)





MRF

BP

Free Energy
(move evaluator)

Results to Date

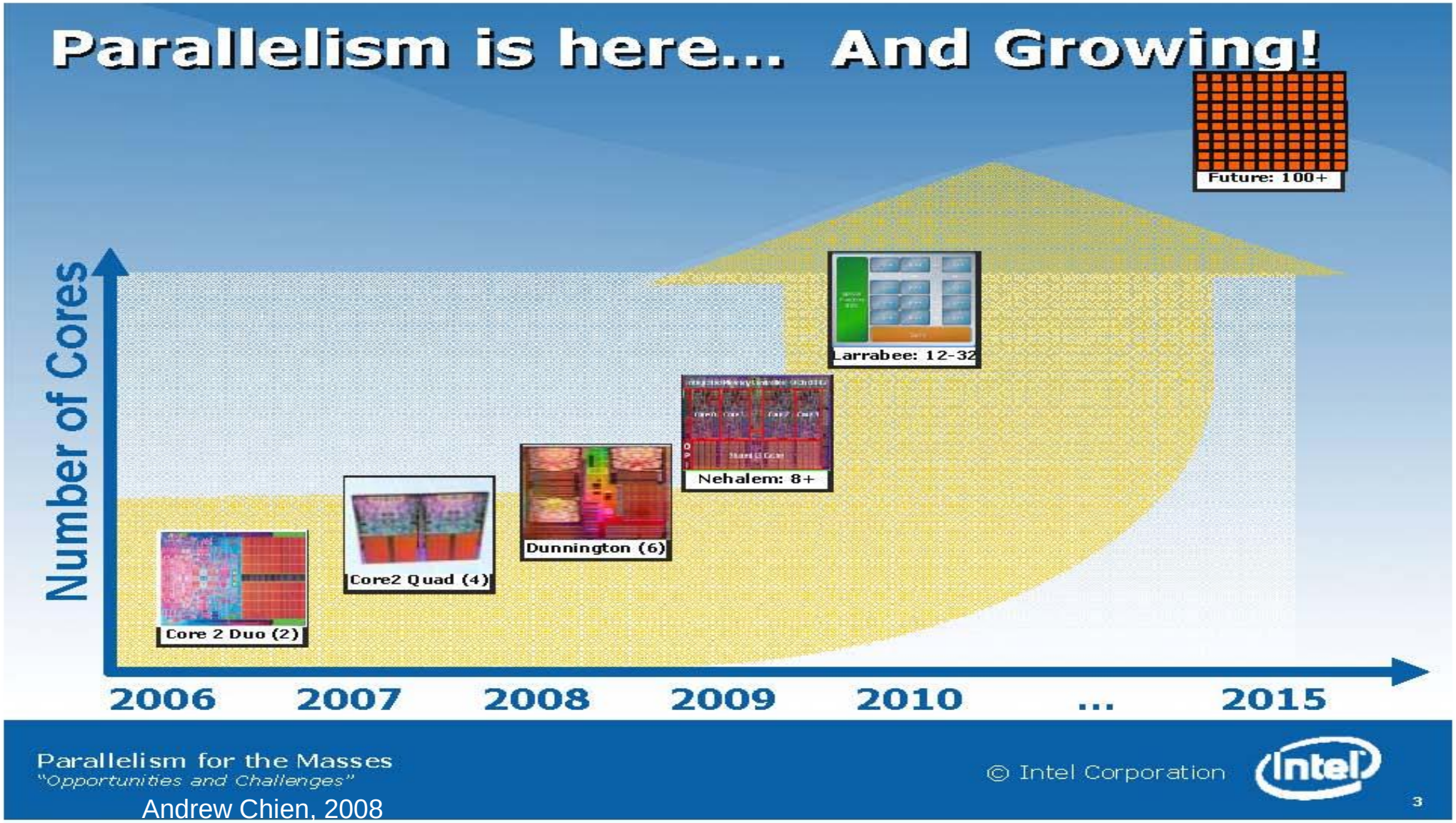
Microsoft Research

- Tasks:
  - Classification: does mutation M confer resistance to drug D?
  - Regression: quantitative accuracy of predicted free energies
- Data:
  - 7 commercially available drugs
  - Wild-type + 14 mutants known to confer resistance
- Accuracy:
  - Classification: 71% accuracy (sens = 68%; spec = 100%)
  - Regression: 0.9 kcal/mol

Microsoft Research

Parallel Thinking

Guy E. Blelloch
Professor
Carnegie Mellon University



Andrew Chien, 2008

Problems in Moving to Multicores

Microsoft Research

- Building the hardware?
- Developing good programming languages or extensions to existing languages?
- Developing efficient parallel algorithms?
- Developing good compilers, run-time systems, and debuggers?
- Developing good OS support?
- Rewriting the existing code base?
- **Educating programmers to "think parallel"?**

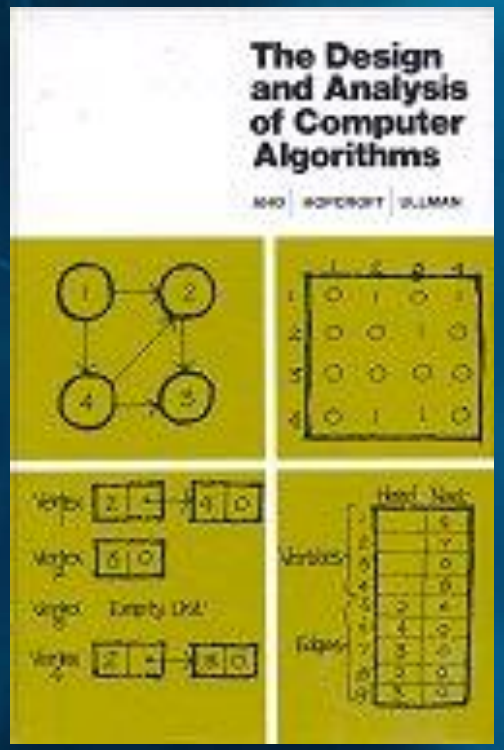
Parallel Thinking

Microsoft Research

- As with “Computational Thinking” in general, “Parallel Thinking” is not a set of ad-hoc libraries, programming languages, and interfaces to learn, it is a set of **core ideas** and a **way of approaching problems**.
- If explained at the right level of **abstraction** many algorithms are naturally parallel? We currently **brainwash** programmers to think sequentially.
- Conjecture: If done right with appropriate knowledge parallel programming might be as easy as sequential programming for many uses?

Example: Quicksort

procedure QUICKSORT(**S**):
  if S contains at most one element **then return S**
  else
    begin
      choose an element **a** randomly from **S**;
      let $\mathbf{S_1}$, $\mathbf{S_2}$ and $\mathbf{S_3}$ be the sequences of
          elements in **S** less than, equal to,
          and greater than **a**, respectively;
      return (QUICKSORT($\mathbf{S_1}$) followed by $\mathbf{S_2}$
        followed by QUICKSORT($\mathbf{S_3}$))
end



Two forms of natural parallelism

Example: Parallel Selection

Microsoft Research

```
{e in S | e < a};
```

S = [2, 1, 4, 0, 3, 1, 5, 7]

F = S < 4 = [1, 1, 0, 1, 1, 1, 0, 0]

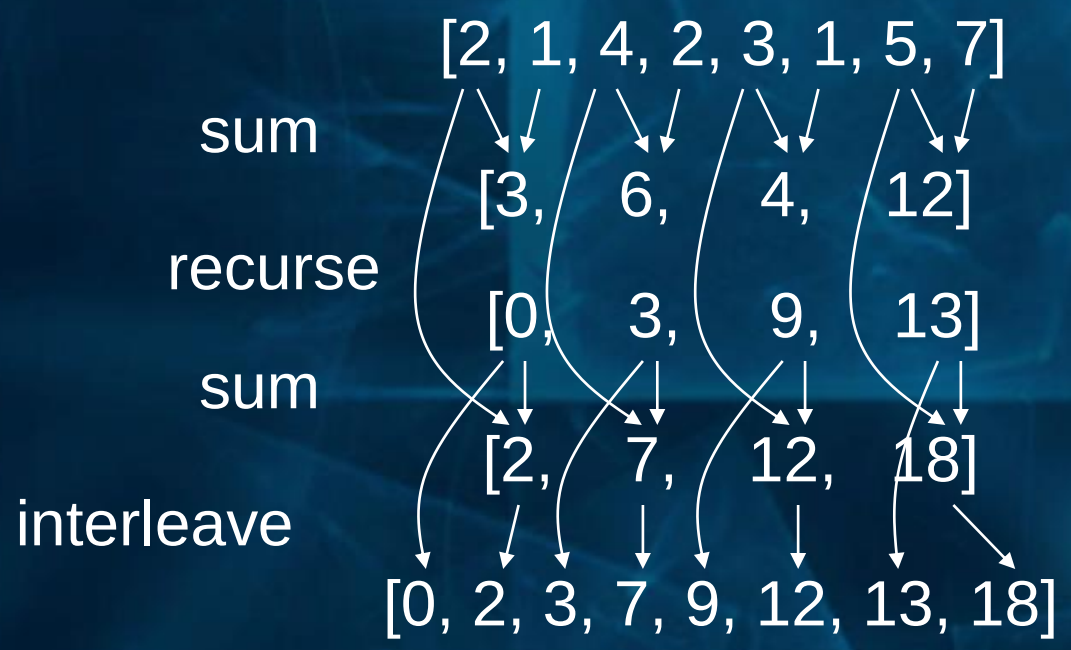
I = addscan(F) = [0, 1, 2, 2, 3, 4, 5, 5]

where F

R[I] = S = [2, 1, 0, 3, 1]

Each element gets sum of previous elements.
Seems sequential?

Example: Scan



Conclusion

Microsoft Research

Educating programmers and researchers on parallelism is key.

We need to identify the “core” ideas in parallel thinking and concentrate on these.

Perhaps we can teach parallelism from day 1 and without much more effort than teaching sequential computing.

Microsoft Research

Computational Thinking and Music Performance

Roger B. Dannenberg
Associate Research Professor of Computer Science and Art
Carnegie Mellon University

The Big Picture

Microsoft Research

- *Everyone likes music*
- Most just listen, but *many* play
- Music Merchants: $8B, 5M instruments (US, 2006)
- Sound reinforcement: $1.5B (US, 2006)
- Audacity Audio Editor (Dannenberg & Mazzoni): 1M/month

- *Computation can enhance the musical experience by providing automated, live, musical partners*

The Performer

Microsoft Research

- Real-time performance synchronized to human musicians
- Assumes quasi-steady tempo
  - research: characterize tempo variation in human performance
- Uses foot-tapping to give the beat to the computer
  - research: interfaces and methods for tempo acquisition and cues
- Uses pre-recorded audio (20 instruments in real time)
  - research: high-quality, low-latency, time-variable, ensemble time stretching

Microsoft Research

The Performer in Concert

Video

Carnegie Mellon Jazz Ensemble + strings, directed by Dave Pellow, “Alone Together” arranged and conducted by Dr. John Wilson, strings recorded at Carnegie Mellon School of Music

Future Work

Microsoft Research

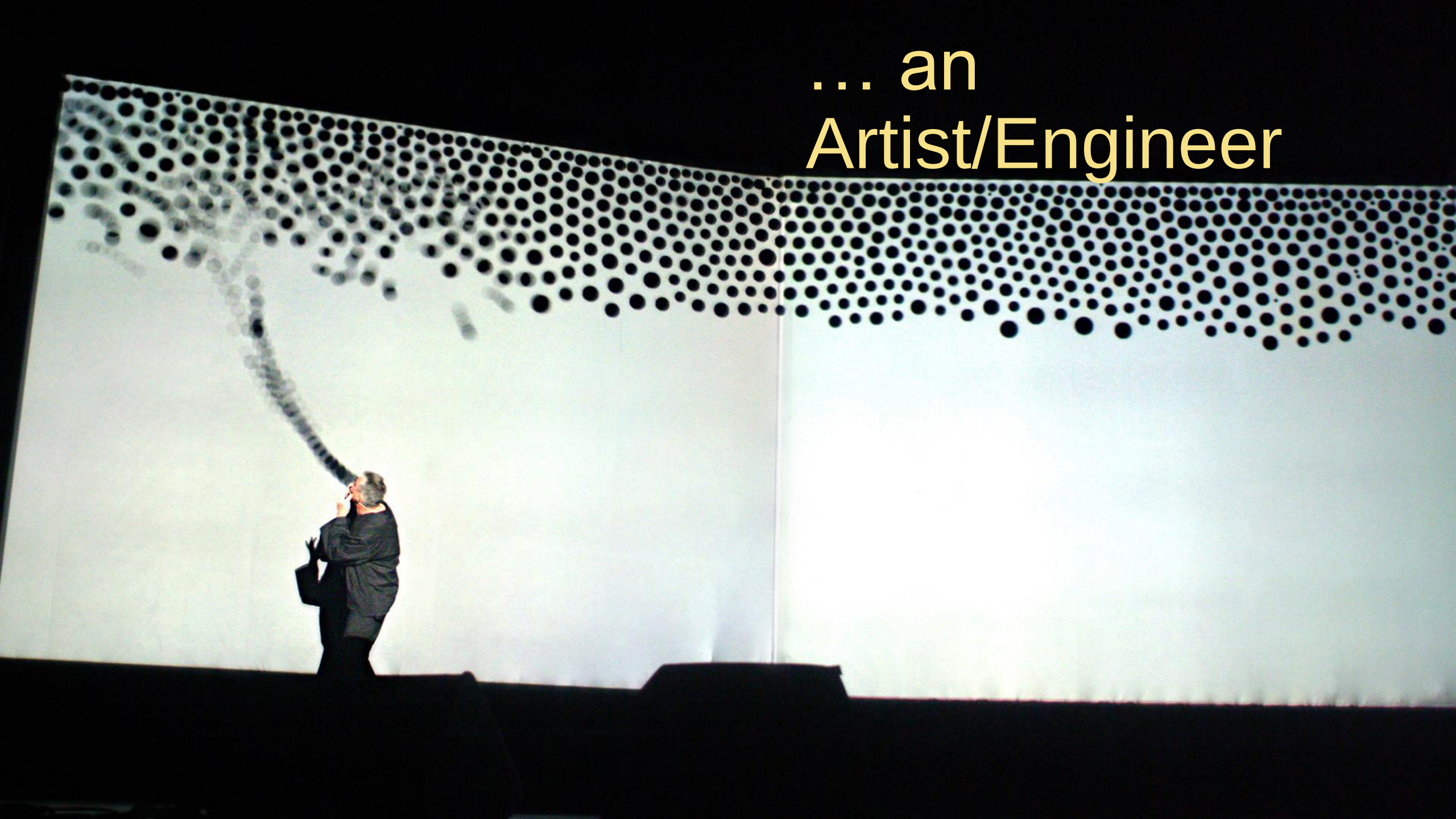
- Interface, interface, interface
- Sensing
- Display
- Computational Thinking and the Digital Music Stand
  - Tablet PC and smaller platforms (Kindle? Cell phones?)
  - Capture music notation as digital photos
  - Record all rehearsals
  - “Learn the music” for page turns, etc.
  - Feedback: location, intonation, cues

Microsoft Research

ART and CODE

Golan Levin
Associate Professor of Art
Director, Studio for Creative Inquiry
Carnegie Mellon University

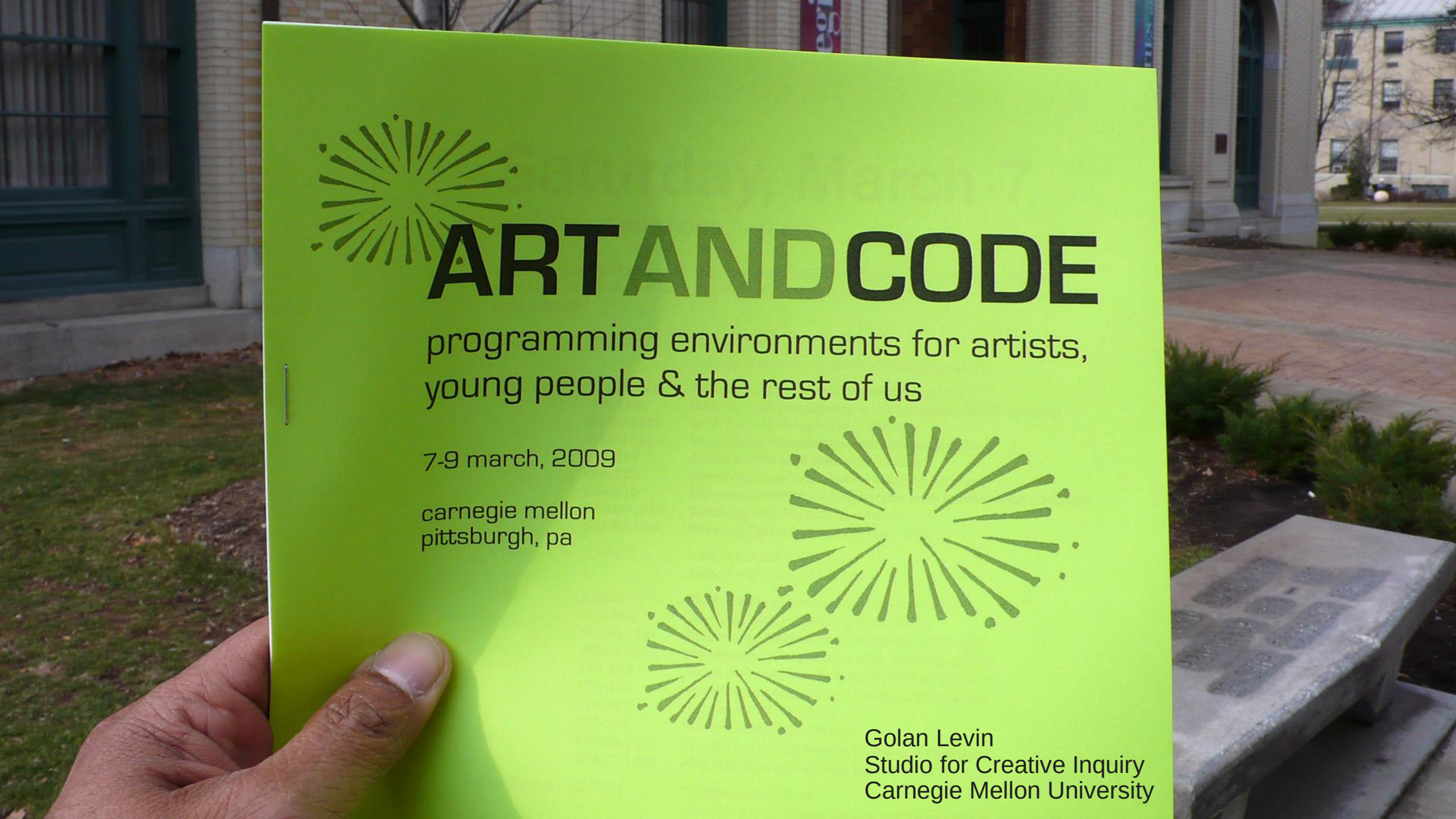
… a Director


… an
Artist/Engineer

… and an Educator

Microsoft Research

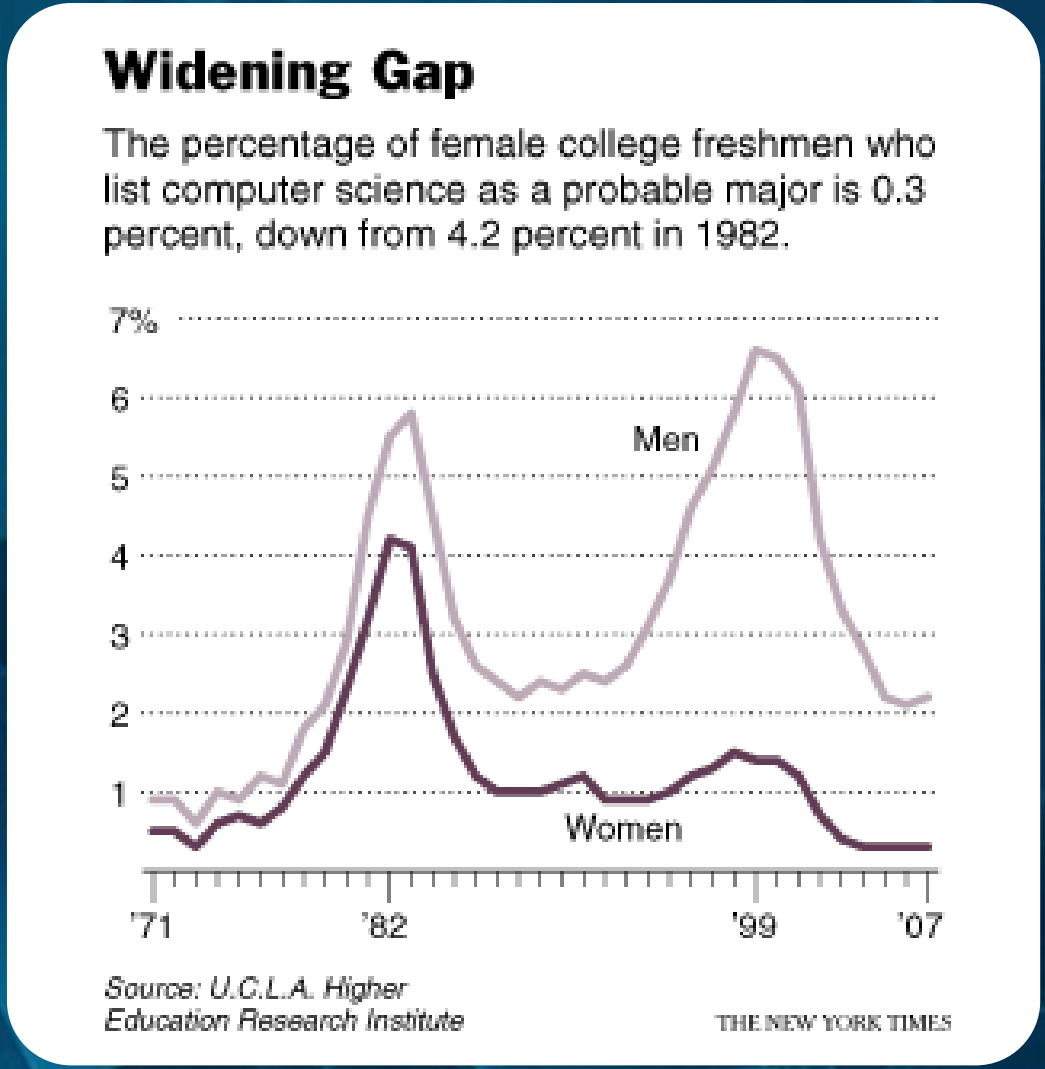
- “Studio Arts courses in Computer Science”
- Where the *medium* is code,
  but the *objective* is self-expression
  - Introduction to Computational Form
  - Information Visualization as Personal Inquiry
  - Interactive Technologies for Live Performance
  - Audiovisual Systems and Machines
  - Generative & Digital Fabrication



Golan Levin
Studio for Creative Inquiry
Carnegie Mellon University

"Where's my Place?"

Microsoft Research



"Where's my Place?"

- There is no category for
  “art software”
  in this landscape.

Reason 1.
("That's Not Art/CS!")

Microsoft Research

- We're too slow to acknowledge that Computation and Culture exist *in feedback*
  - Computing influences the evolution of culture
  - Culture motivates and prompts advances in computing
  - "Culture" = people communicating together

Reason 2.
("Who, me?")

- A pervasive belief that software is made *by someone else*

Microsoft Research

Reason 2.
("Who, me?")

- A pervasive belief that software is made *by someone else*
- Generally, we can’t even pimp it the way we can pimp our cars

Microsoft Research

What is Software *Literacy*?

Microsoft Research

- Just as true literacy in English means being able to *write* as well as *read* …

What is Software *Literacy*?

Microsoft Research

- Just as true literacy in English means being able to *write* as well as *read* …
- True literacy in software demands not only knowing how to *use* commercial software tools, but also how to *create* new software for oneself and for others

Programming as a Liberal Art

Microsoft Research

- CS departments should not have the monopoly on teaching programming
- *Programming* (an everyday skill) and *Computer Science* (a research subject) are different
- Computer scientists are not necessarily good at teaching programming to people from other disciplines *(who may have very different motivations for learning)*

FUBU

THE COLLECTION

11 Different Arts-programming Languages

Microsoft Research

- Actionscript/Flash (Adobe)
- Extendscript (Adobe)
- Alice (CMU)
- Hackety Hack
- Max / MSP / Jitter
- openFrameworks (C++)
- Processing (Java)
- PureData
- Scratch
- Silverlight
- VVVV

… not "Research Microlanguages"

Microsoft Research

- The significance of these tools is **not theoretical**
- Each of these toolkits has **10K – 1M+** users
- They have totally transformed the landscape of which kinds of people are now writing code
- People like: “artists, young people, and the rest of us”


SCRATCH
scratch.mit.edu


Alice was originally designed to address
Dropout rate in first year for CS majors
Informal surveys: 35 – 70%
Typically 40 – 50 %
Percentage of women in CS is abysmal (less than 20% of BS degrees are awarded to women)
The future demand for IT professionals is greater than the projected ability to meet that demand


Processing . . .
. . . is an open source project for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool.
Ben Fry, Casey Reas et al.


openFrameworks

Atelier-Style Education

Microsoft Research

Importing methods of arts pedagogy:

- Curiosity-driven problem generation and solving
- Individualized feedback for individual needs
  - Similar to a music instrument lesson


Drawing Cats
with Hackety Hack

Attendee Diversity

Microsoft Research

- Many, many people want to learn to code
  - **234** total registered attendees
  - Hailing from **7** countries (inc. France, Norway, …)
  - Hailing from **23** different American states
    - 22% from Carnegie Mellon University
    - 39% from Pittsburgh and Allegheny County
    - 67% from East Coast + Rust Belt
  - Ages 11 through 75
  - 35% female

Attendee Diversity

Microsoft Research

- So diverse, in fact…
  - “Is this event for developers, or kids?” *yes…*
  - “Is this event for developers, or artists?” *yes…*
  - “…” (sound of brain breaking)

Some introductions…


Bennie Whitcomb
Future Engineer
magus5@qis.net
(410) 532-1883
635 Saint Johns Road
Baltimore, MD 21210


Mâa
Software Explorer
15, rue Poissonnière
75002 Paris - France
33-1-44 82 05 49
33-6-78 14 23 58
maa@lagraine.com


Jon
Schull
Rochester Institute of
Technolgy
ARTANDCODE
Jon
Schull
Rochester Institute of
Technolgy
ARTANDCODE


Dr. Holly Pellerin
Fond du Lac Elder &
Program Director,
Gidakiimanaaniwigamig
Native American Youth
Science Enrichment Program,
Fond du Lac, MN
Beverly Nye
Arts Teacher
McPherson Middle School
McPherson, KS

Workshops at Art && Code

Microsoft Research

- 21 workshops in 11 different programming languages
- Generally 10-20 students per workshop
- About 1100 person-hours of learning in one weekend


DELL

Art && Code: a Summit / Symposium

Microsoft Research

Creators of these toolkits had never been gathered together

- Topic: what motivated the creation of these toolkits
- Topic: Feedback between user communities + features *(particularly for open-source toolkits)*


Canon

A New Digital Community

Microsoft Research

- Built with Ning, quickly grew to 500+ global members
- Continues to grow, even months later
- Symposium videos receiving thousands of views
- All online… *have a look!*
  artandcode.com

ARTANDCODE: programming environments for artists, young people & the rest of us

march 7-9, 2009 at carnegie mellon, pittsburgh, pa

ART AND CODE is a symposium on programming tools for artists, young people, and the rest of us -- hosted March 7-9, 2009 at Carnegie Mellon.

Download the PDF schedule Edit

Click here to register!

ARTANDCODE

SAT	SUN	MON

Click to download the conference schedule (PDF)

Community Sponsors Edit

Microsoft Research

Center for Computational Thinking Carnegie Mellon

STUDIO FOR CREATIVE INQUIRY

Welcome to ART AND CODE 2009 (#artandcode) Edit

ART AND CODE is a conference and online community focused on programming environments for artists, young people, and the rest of us. The conference took place the weekend of March 7-9, 2009 on the campus of Carnegie Mellon University in Pittsburgh, PA. It featured hands-on workshops and a symposium showcase for 11 different creative toolkits -- programming languages made by artists, for artists. If you have questions or concerns, please email Golan Levin at golan@andrew.cmu.edu.

ART AND CODE is made possible by a generous grant from Microsoft Research, with oversight by the Center for Computational Thinking at CMU. **ART AND CODE** is a project of the CMU STUDIO for Creative Inquiry, directed by Golan Levin.

facebook Join our Facebook group!

Latest Activity Edit

Woroud Ahdali is now a member of Art and Code — 14 hours ago

Welcome Them!

Annette Vee replied to Kyle Geske's discussion 'Introduction to Programming' in the group Educators — yesterday

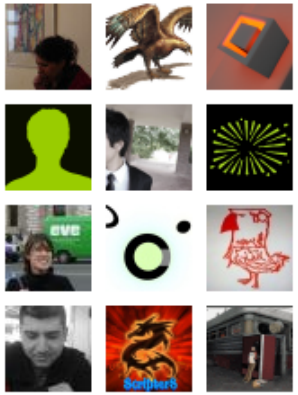
"Kyle, I think that 90% of teaching isn't content or syllabus but a je ne sais quoi of personality and enthusiasm, and you're so thoughtful and smart--I'm sure whatever you teach will be great! But that's not very helpful, is it? :) If this is an..."

Angel Giovanni Cortes Solorzano is now a member of Art and Code — on Wednesday

Golan Levin

Sign Out
Inbox (3 new)
Friends – Invite
Settings
Quick Add...

Members Edit



+ Invite More View All

Groups Edit

Educators
31 members

Plug In / Plug Out
4 members

Student

Show me newest videos in thumbnail format Search these videos

ART && CODE SYMPOSIUM: Keynote, Tom McMail
9 hours ago

ART && CODE SYMPOSIUM: vvvv, Sebastian Oschatz
3 days ago

ART && CODE SYMPOSIUM: Processing, Ben Fry and Casey Reas
3 days ago

ART && CODE SYMPOSIUM: ExtendScript, Dr. Woohoo
3 days ago

ART && CODE SYMPOSIUM: Max/MSP/Jitter, Luke DuBois
3 days ago

ART && CODE SYMPOSIUM: Pure Data, Hans-Christoph Steiner
3 days ago

ART && CODE SYMPOSIUM: Introduction, Golan Levin
3 days ago

ART && CODE Symposium: Hackety Hack, why the lucky stiff

STUDIO for Creative Inquiry's videos

Here are all STUDIO for Creative Inquiry's videos on Vimeo. You can see both the videos this user has uploaded, as well as any other users' videos they appear in.

Choose "Uploaded" to see the videos this user has uploaded to Vimeo. Choose "Other Credits" to see the videos that STUDIO for Creative Inquiry is credited in by other users.

We recommend using the sort bar which allows you to view these videos in different orders or formats. If you are looking for a particular video, use the "search these videos" link.

Partners

Microsoft Research

Center for Computational Thinking
Carnegie Mellon

STUDIO
FOR CREATIVE INQUIRY

Microsoft Research

Microsoft®

Your potential. Our passion.™