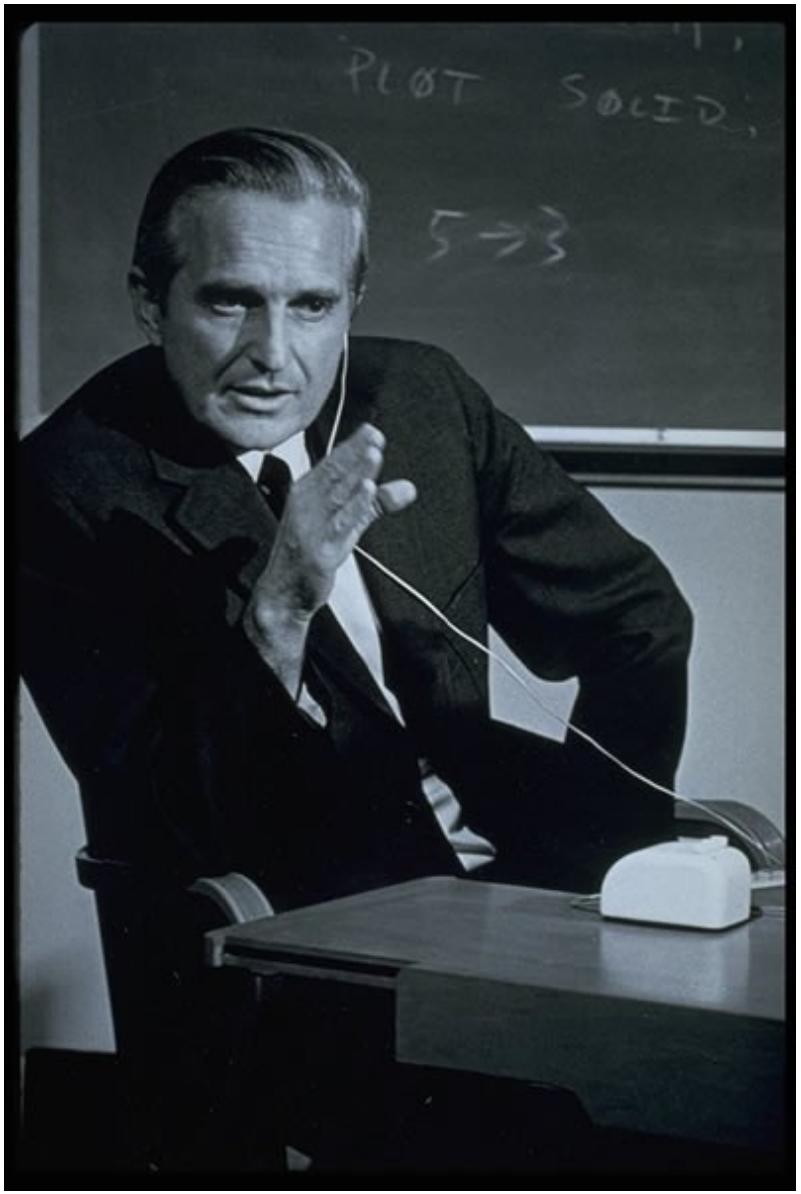
Douglas Engelbart's HyperScope:

Taking Web Collaboration to the Next Level Using Ajax and Dojo

Brad Neuberg
bkn3@columbia.edu
EuroOSCON, September, 2006

Douglas Engelbart



Invented

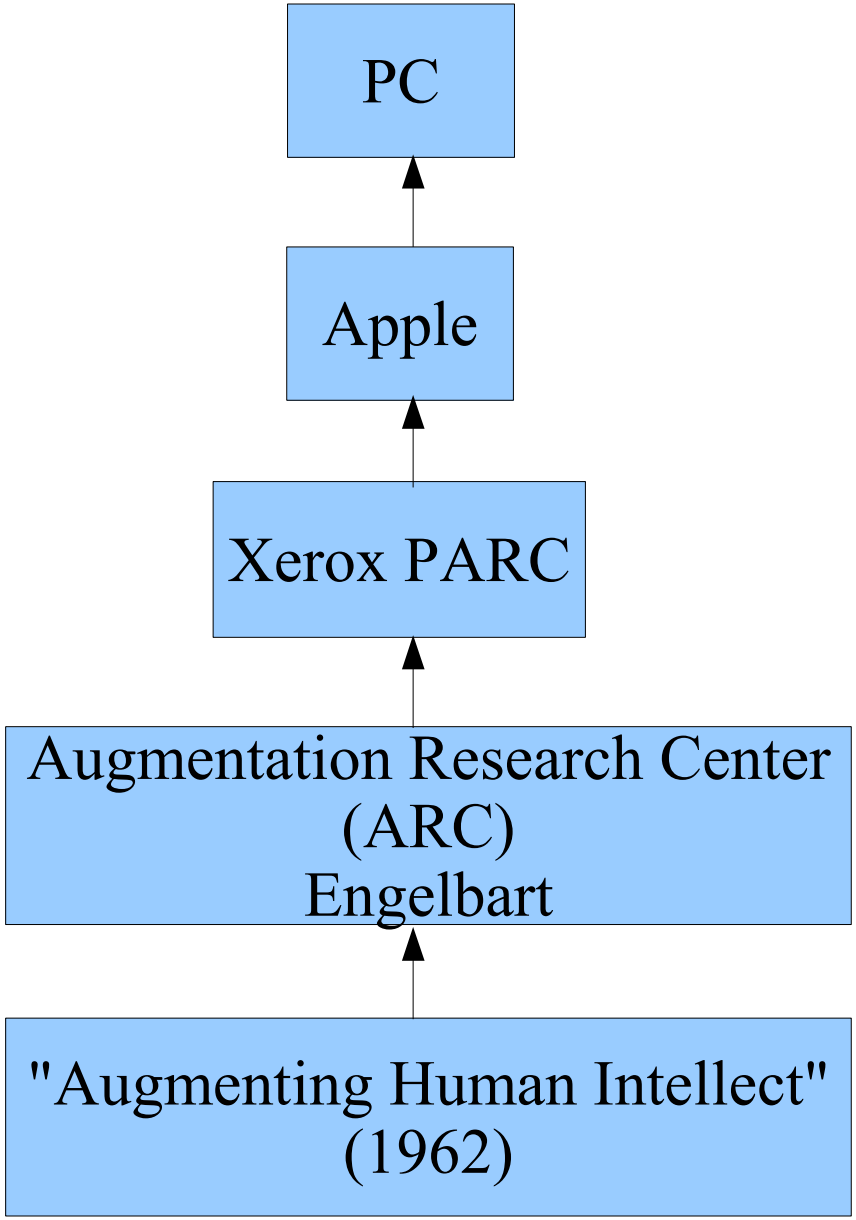
- Computer Mouse
- Hypertext
- Collaborative groupware
- Email
- Windowing
- And more

Augmentation

Augmenting Human Intellect

(1962)

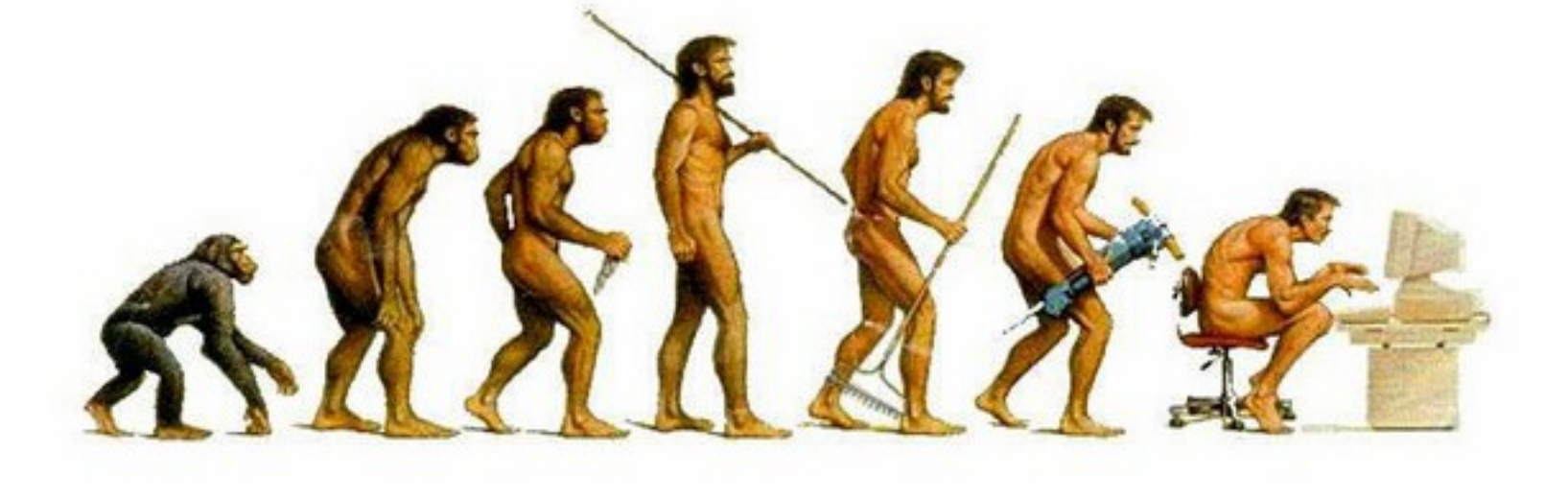
Importance of Engelbart's Paper



What is Augmentation?

"Matrix" Augmentation

Accelerated Evolution



Improving Human Abilities

- Increasing intellectual ability
- Speeding up process of improvement
  - evolution

Why Accelerate Intellectual Evolution?

"Man's population and gross product are increasing at a considerable rate, but the *complexity* of his problems grows still faster, and the *urgency* with which solutions must be found becomes steadily greater..."

"Augmenting Human Intellect,"
Engelbart, 1962

Tools for Thinking

- Language (40,000 years?)
- Writing (3,500 - 6,000 years?)
- Computers (~1950s)

Two Big Ideas in Engelbart's Paper

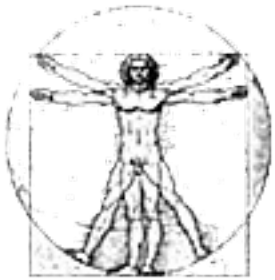
- Look at humans and their tools holistically
- Capabilities

Systems Thinking

Tools

Systems Thinking

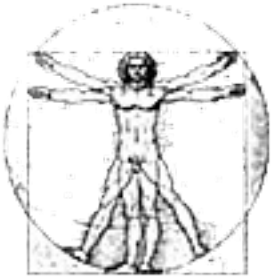
Tools



Human

Systems Thinking

Tools

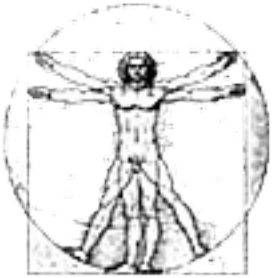


Human

Training

Systems Thinking

Tools



Human

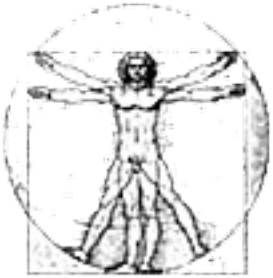
Training

Methodology

Systems Thinking

Tools

Language



Human

Training

Methodology

Capabilities

- The ability or skill to do something
- Examples:
  - Talking on the cell phone
  - Riding a bike
  - Forming thoughts in your mind
  - Giving a presentation

Capabilities

- All 5 parts of system can have capabilities

Major Aspects of Capabilities

1) Can be broken down
2) Work together
3) Some are more core than others

Key Insight

- Target core capabilities
- Make them better
- Payoff will ripple all over
  - Leverage

Leverage

"Give me a place to stand on, and I can move the earth."

Archimedes, 287 BC - 212 BC

Formula to Accelerate Evolution

1) Identify capabilities in all areas of our human system

Tools

Language

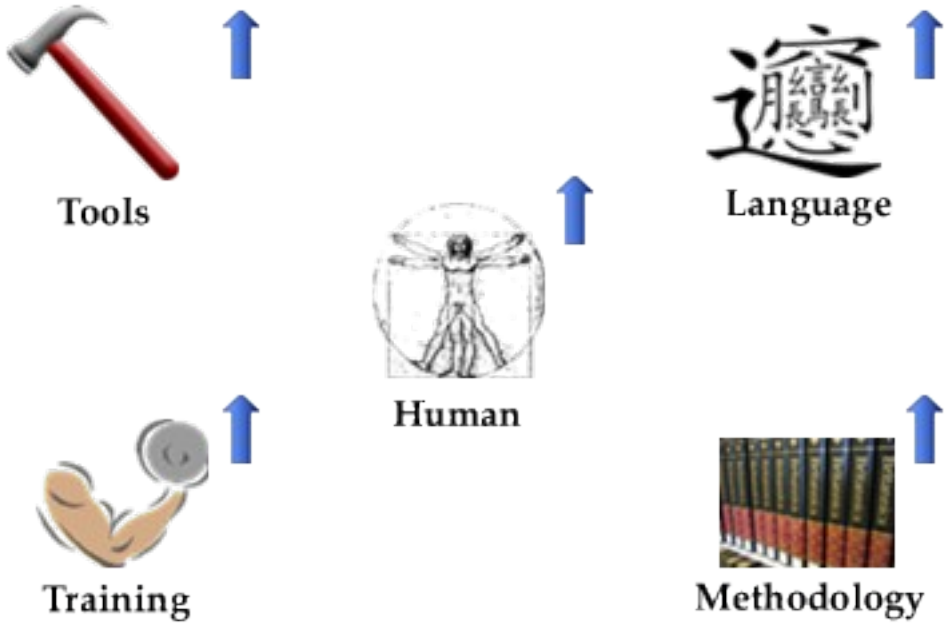
Human

Training

Methodology

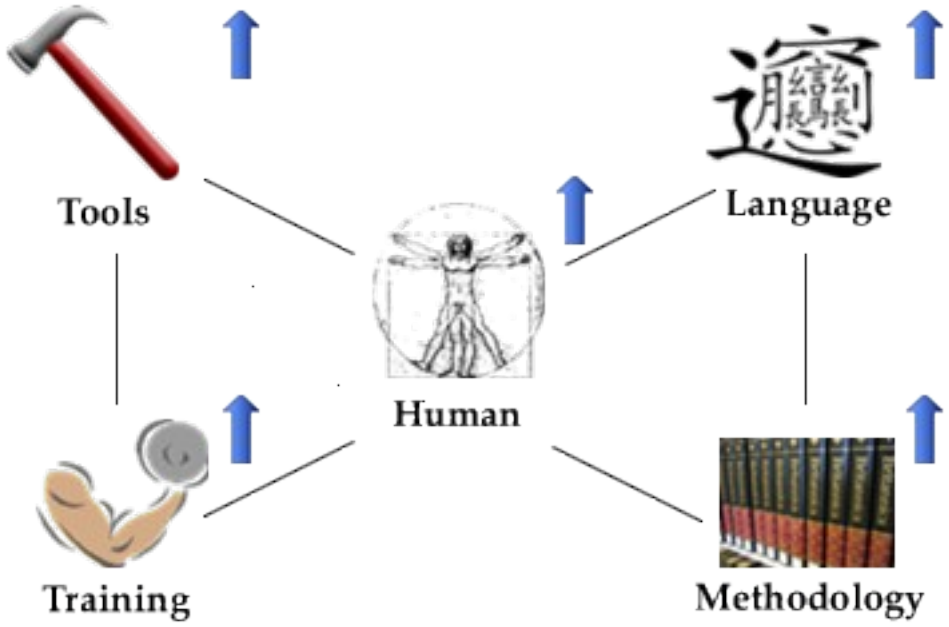
Formula to Accelerate Evolution

2)Target core capabilities and make them more powerful



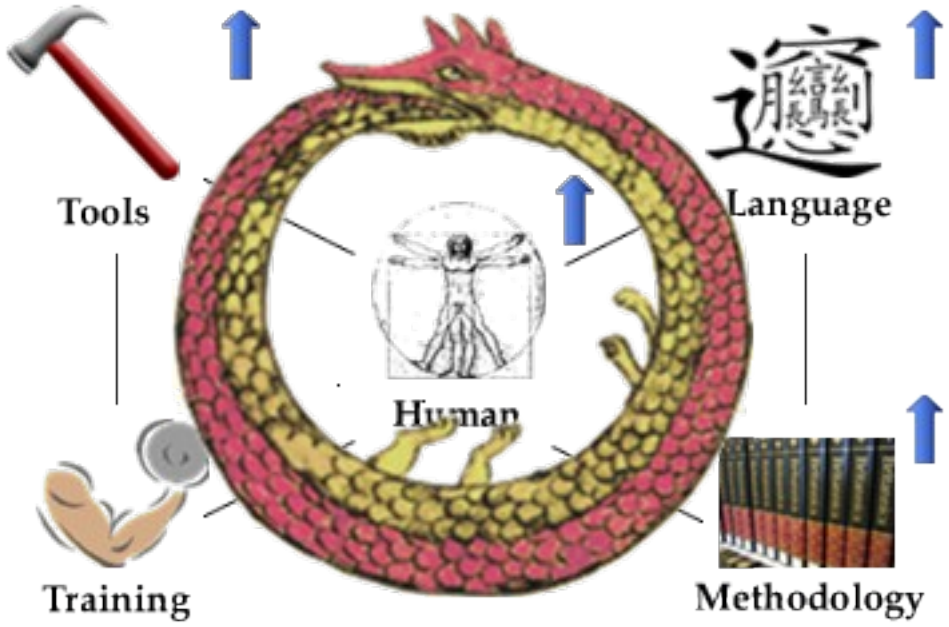
Formula to Accelerate Evolution

3)Have everything start feeding together



Formula to Accelerate Evolution

4)Critical mass happens - repeat over and over - "bootstrapping"



Summary of Paper

- "Augmenting Human Intellect" key paper
- Two big ideas:
  - Look at Humans and Tools holistically
  - Capabilities
- Identify key capabilities and make more powerful
- Self-perpetuating bootstrapping will occur

Augmentation Research Center

- 1960s and 1970s
- Engelbart and team applied framework to themselves
- Bootstrapped over and over, using all 5 human/tool areas

Demo of NLS/Augment

- One copy still running in world
- Hosted at Logitech
- Running Debian Linux on Solaris
- Emulating PDP-10 and TOPS-20
- Still thinks it's second node on Internet

Demo of NLS/Augment

- Probably less than 100 people in world have directly used
- Mother of All Demos

Smalltalk Augment

- Built in early 90s by Engelbart and team
- Tim Berners Lee saw it in early 90s
- Implemented some extra GUI elements
  - Quick buttons
  - Jump Window
  - Viewspec Window

Demo of Smalltalk Augment

HyperScope

- Goal:
  - Bring Augment to contemporary web
  - Translate, don't innovate

HyperScope

- Uses modern technology:
  - Ajax, DHTML, OPML, Dojo, Web
- Implements following Augment features:
  - Jumping
  - Addressing
  - Viewing
  - Command Bar

HyperScope

- New feature:
  - *transformers* to allow all document formats to do advanced addressing

HyperScope

- Open source (GPL)
- NSF funded Phase I
- Phase 1 finished

Demo of HyperScope

- Demo Link - Design Document
- Demo Link - Example Addressing
- Demo Link - OPML 1.0 Document

Transformers

- Transforms other document types into HyperScope OPML
- Created by community

Transformers

- Currently have:
  - RSS -> OPML
  - Microsoft Word -> OPML
  - Microsoft Powerpoint -> OPML
  - XOXO -> OPML
  - Augment -> OPML

Demo of Transformers

- XOXO:
  - Before
  - After
- RSS
  - Before
  - After

Architecture

- *Everything* is client-side
  - Except for small, optional PHP gateway for cross-host transcludes
- OPML is file format

Architecture

- Client applies XPath to resolve addressing
- Uses XSLT to render viewspecs and final document
- HTML is produced and pushed to screen

Architecture

- Has JavaScript classes that represent domain:
  - hs.address.Address
    - A HyperScope address that can be resolved and manipulated
  - hs.model.Document
    - An outline document that can be rendered and jumped through
  - hs.model.Node
    - A node in an hs.model.Document

Architecture

- Divided into two major pieces, both on client-side:
  - "Front-End"
    - UI that knows how to interact with user, mouse, screen, etc
  - "Back-End"
    - Resolves documents, does addressing, etc. Independent of browser environment

Architecture

- Front-End turns all user operations into an hs.address.Address
- We then resolve this address
- Back-End does hard work of figuring out how to do this
- We then render results in Front-End

Architecture

- We use Dojo:
  - Dojo Events
  - Dojo IO
  - Dojo Widgets
- Sarissa for cross-browser XPath and XSLT
- Massive unit testing done with JSUnit
  - Clone of Java JUnit

Final Thoughts

- HyperScope brings Augment to contemporary audience
- Lets do what ARC did in 60s and 70s:
  - Keep applying the Augmentation framework
  - Bootstrap ourselves in all 5 areas (human, language, tools, methodologies, training)
  - See what innovations come out

Final Thoughts

- Next step:
  - Start innovating
  - Bring in editing
  - More transformers
  - Engage community

Final Thoughts

- Play with HyperScope and learn more at http://hyperscope.org/

Douglas Engelbart's HyperScope:

Taking Web Collaboration to the Next Level Using Ajax and Dojo

Brad Neuberg
bkn3@columbia.edu
EuroOSCON, September, 2006