IBM

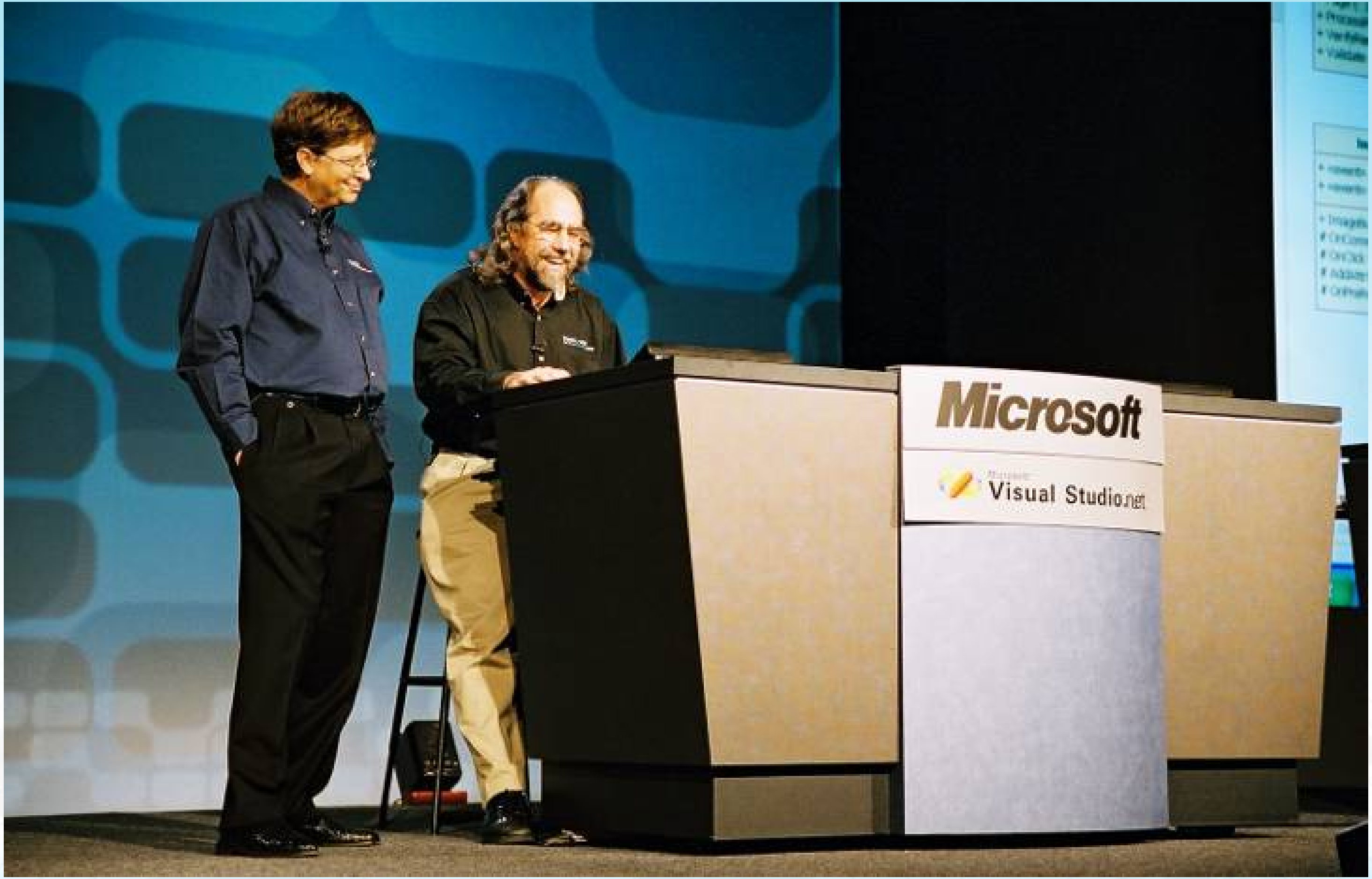
IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

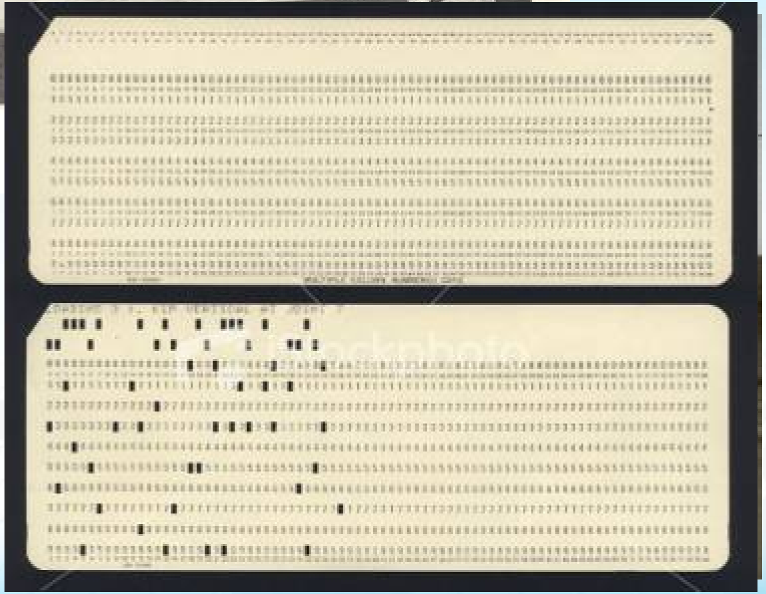
GRADY BOOCH

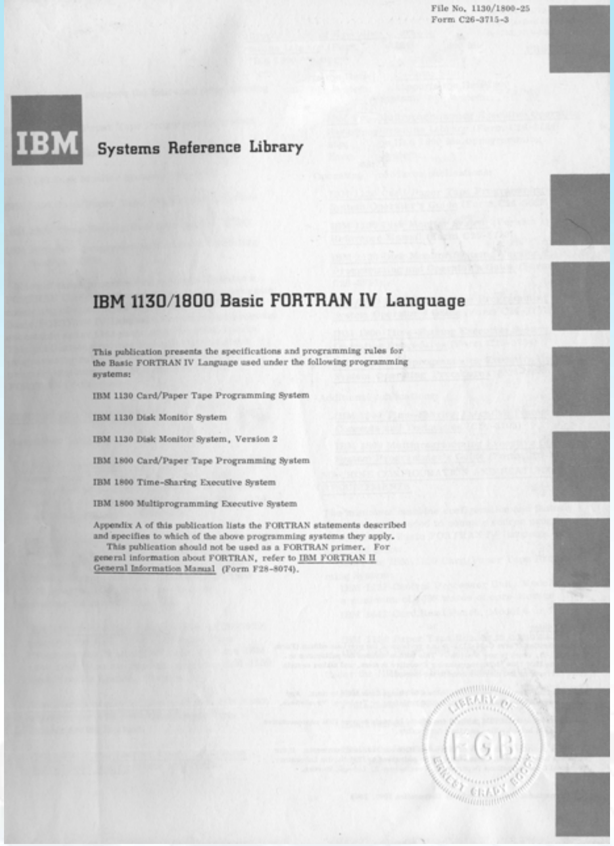
IBM FELLOW
CHIEF SCIENTIST FOR SOFTWARE ENGINEERING,
IBM RESEARCH
FREE RADICAL

Rational. software


Microsoft
Visual Studio.net

The Future Of Software





File No. 1130/1800-25
Form C26-3715-3

IBM Systems Reference Library

IBM 1130/1800 Basic FORTRAN IV Language

This publication presents the specifications and programming rules for the Basic FORTRAN IV Language used under the following programming systems:

IBM 1130 Card/Paper Tape Programming System

IBM 1130 Disk Monitor System

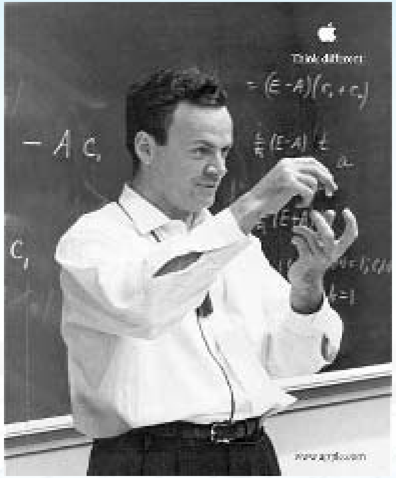
IBM 1130 Disk Monitor System, Version 2

IBM 1800 Card/Paper Tape Programming System

IBM 1800 Time-Sharing Executive System

IBM 1800 Multiprogramming Executive System

Appendix A of this publication lists the FORTRAN statements described and specifies to which of the above programming systems they apply.
This publication should not be used as a FORTRAN primer. For general information about FORTRAN, refer to IBM FORTRAN II General Information Manual (Form F28-8074).


Think different.

The Current State

The typical software-intensive system is

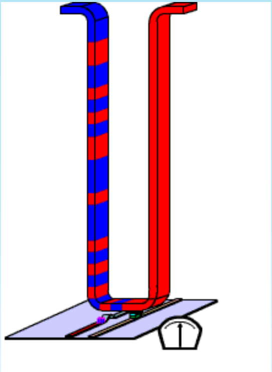
- **Continuously evolving**
- **Connected, distributed, & concurrent**
- **Multilingual & multiplatform**
- **Secure & autonomic**
- **Developed by geographically- temporally-distributed teams**

Most systems are actually systems of systems

- **Services & other messaging mechanisms dominate**
- **Such systems encompass both hardware & software**

Every advance leading to the future state of the world requires the presence of software yet-unwritten as of today

Growth Of Storage



The production of data is growing

- **Google processes 20 petabytes/day**[1]
- **The Internet handles over 627 petabytes/day**[2]

- ***Storage densities are increasing***
  - **200 gigabytes/inch^2 are common today**
  - **Racetrack memory could increase storage density by a factor of two (20,000 gigabytes/inch^2)**[3]

[1] http://www.niallkennedy.com/blog/2008/01/google-mapreduce-stats.html
[2] http://en.wikipedia.org/wiki/Petabyte
[3] http://www.almaden.ibm.com/spinaps/research/sd/?racetrack

Opportunity

Growth Of Computational Power

Computational power is abundant

- **A single BladeCenter can reach 7 teraflops**
- **IBM Road Runner may reach one petaflop**
- **Hardware costs are around 20 cents/gigaflop; operating costs are approximately 3 watts/gigaflop[1]**

- ***The frequency scaling wars are ending***
  - **At 10 atoms/transistor, quantum effects & power dissipation become critical issues issues**
  - **Multicore processors are becoming the norm**

Opportunity

[1]http://en.wikipedia.org/wiki/FLOPS

Growth Of Connectivity

Bandwidth is increasing

- **Copper may reach 10 gigabytes/second**
- **Wireless networks are becoming pervasive**

- ***Out of 3.7 billion IPv4 addresses[1]***

– ***China***	***19.246 million***
– ***US***	***13.610 million***
– ***Germany***	***5.414 million***
– ***Italy***	***3.881 million***
– ***Indonesia***	***3.465 million***
– ***Taiwan***	***3.455 million***

[1]http://www.bgpexpert.com/addressespercountry.php

Given relatively unlimited storage, abundant computational power, & pervasive connectivity…

What will future software-intensive systems look like?

How will we develop, deploy, & evolve such systems?

What is the value proposition?

Growth Of Storage

- **Searching & indexing problems grow exponentially**
- **What are the privacy implications of having your entire life recorded & nothing forgotten?[1]**
- **Will we enter a digital dark age?[2]**

[1] http://www.guardian.co.uk/science/2005/dec/28/research.highereducation
[2] http://www.rense.com/general38/escap.htm

Challenge

Growth Of Computational Power

- Writing correct software for intimate concurrency is a wicked problem
- Data center energy costs are becoming a limiting factor
- There's no lack of sloppy software and/or hard problems that eat cycles
  - XML
  - Ray tracing
  - NP complete problems

Growth Of Connectivity

- **Connectivity is unevenly distributed**
- **Opportunities for security breaches abound**
- **Opportunities for offensive cyberwarfare are emerging**[1]

[1]http://www.afcyber.af.mil/

Challenge

Furthermore…

- How can you trust/have confidence in a system of systems over which you have no control of its parts?
- What are the implications for the globalization of systems development, deployment, & evolution?
- What are the economic implications of the commoditization of hardware & software?
- How does one address the inertia of legacy systems?

Design “Flaws” In The Web

- Poor separation of concerns between presentation & semantics
  - Evolution of the semantic web
- Address exhaustion
  - Moving from IPv4 to IPv6
- Changing assumptions regarding sessions
  - From stateless connections to always on video streaming

Challenge

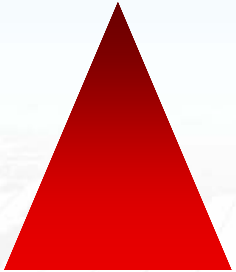
Future Software-Intensive Systems

- Future systems will be just like contemporary ones except they will be
  - More massive
  - More pervasive
  - More transparent
  - More critical

Developing, Deploying, & Evolving

- **Limiting factors are rarely due to the laws of physics or the laws of software**
- **The wicked problem centers around the intrinsic human ability to manage complexity**

Fundamental



Human

Laws of physics
Laws of software
Challenge of algorithms
Difficulty of distribution & concurrency
Problems of design
Importance of organization
Impact of economics
Influence of politics
Limits of human imagination

Value Proposition

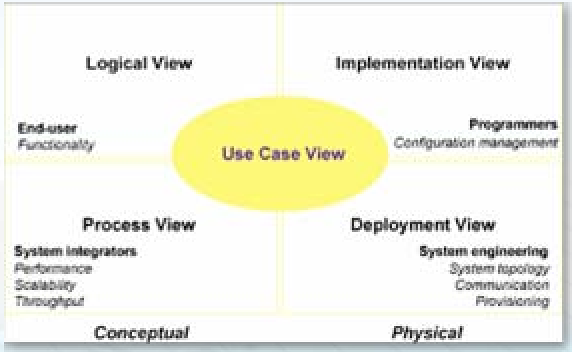
- **There are no limits to human imagination**
- **You can’t outsource innovation**

What We Know

- The fundamentals never go out of style
  - Craft crisp & resilient abstractions
  - Maintain a good separation of concerns
  - Create a balanced distribution of responsibilities
  - Refactor to simplicity
- Process
  - Grow a system through the incremental & iterative release of executable architectures

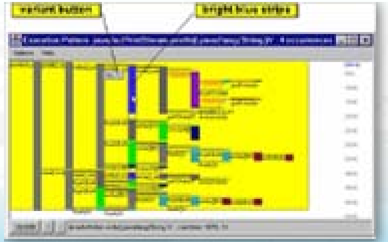
Software Architecture

- ***Every system has an architecture; most are accidental, some are intentional***
- ***Different stakeholders have different concerns & therefore different viewpoints***
- ***All well-structured software-intensive systems are full of patterns***



Multicore

- The average developer does not know how to build secure intimately concurrent software
- However, we can
  - Push some aspects to the operating system
  - Hide some complexity in compilers
  - Offer new programming languages & pattern languages
  - Provide better tools for debugging and visualization



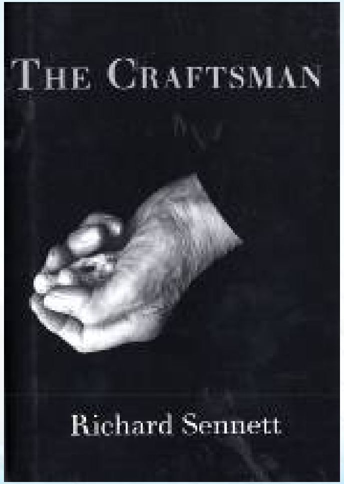
Collaboration

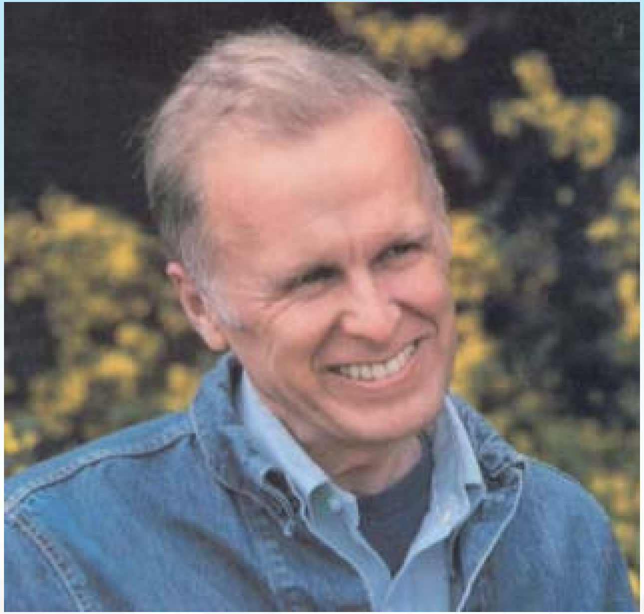
- **Geographic distribution**
  - **Development across time zones and political boundaries**
  - **Issues of trust**
  - **The water cooler problem**
- **Temporal distribution**
  - **The preservation of tribal memory**

Every advance leading
to the future state of the
world requires the presence
of software yet-unwritten
as of today

Ethical/Moral/Legal Considerations

"All craftsmanship is founded on skill developed to a high degree….At its higher reaches, technique is no longer a mechanical activity; people can feel fully and think deeply what they are doing once they do it well. It is at the level of mastery… that ethical problems of craft appear."





For Grady Booch

IBM Systems Reference Library

IBM 1130/1800 Basic FORTRAN IV Language

Go be a hero to someone

IBM

RU READY TO SAVE THE DAY

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES