

APIs of the Future: Are You Ready?

Mike Amundsen
@mamund

<http://linkedin.com/mamund>
<youtube.com/mamund>



Mike Amundsen
@mamund

The
Pragmatic
Programmers

Design and Build Great Web APIs

Robust, Reliable, and Resilient



Mike Amundsen
edited by Katharine Dvorak

O'REILLY®

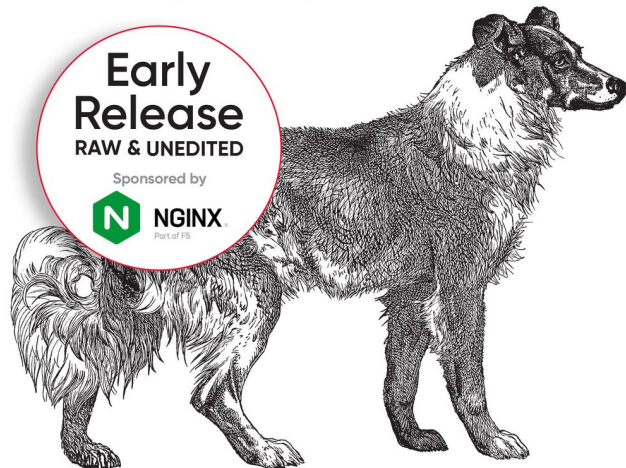
2nd Edition

Continuous API Management

Making the Right Decisions in
an Evolving Landscape

Early
Release
RAW & UNEDITED

Sponsored by



Mehdi Medjaoui, Erik Wilde,
Ronnie Mitra & Mike Amundsen

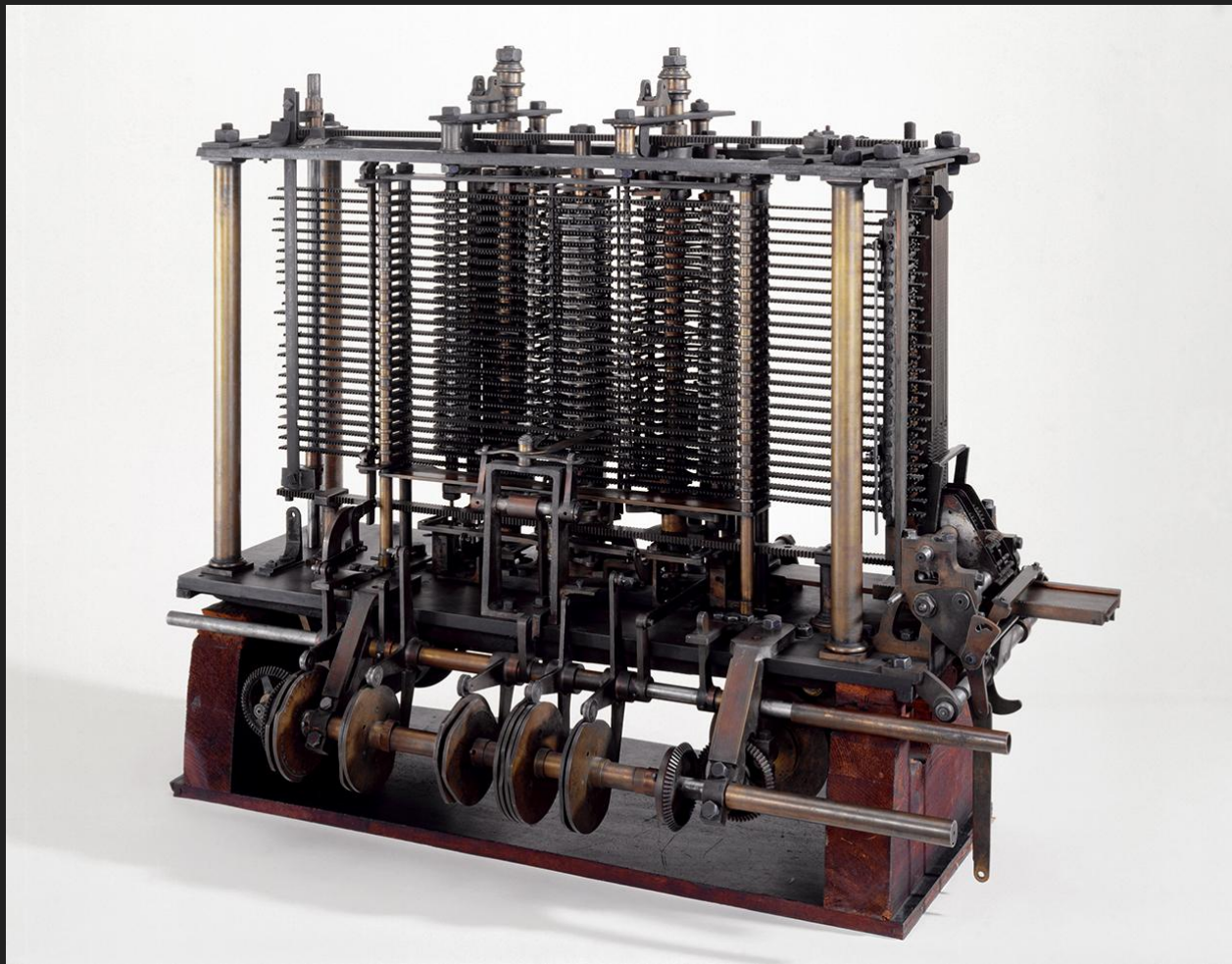
APIs of the Future: Overview

- The First Age of Computing : Where we were
- The Second Age of Computing : Where we are
- The Third Age of Computing: Where we're going
- And So...





Where We Were 1850-1950



By Science Museum London / Science and Society Picture Library - Babbage's Analytical Engine, 1834-1871. Uploaded by Mrjohncummings, CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=28024313>

| Number of Operation. | Nature of Operation. | Variables acted upon. | Variables receiving results. | Indication of change in the value on any Variable. | Statement of Results. | Data. | | | | | | | | | | | | Working Variables. | | | | Result Variables. | | | |
|---|----------------------|-------------------------------------|---|---|--|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|------------------|------------------|------------------|--------------------|------------------|---------------------------------------|---------------------------------------|---------------------------------------|---|--|--|
| | | | | | | 1V ₁ | 1V ₂ | 1V ₃ | 0V ₄ | 0V ₅ | 0V ₆ | 0V ₇ | 0V ₈ | 0V ₉ | 0V ₁₀ | 0V ₁₁ | 0V ₁₂ | 0V ₁₃ | 1V ₂₁ | 1V ₂₂ | 1V ₂₃ | 0V ₂₁ | | | |
| | | | | | | 1 | 2 | n | | | | | | | | | | | | B ₁ in a decimal fraction. | B ₂ in a decimal fraction. | B ₃ in a decimal fraction. | 0 | | |
| 1 | × | 1V ₂ × 1V ₃ | 1V ₄ , 1V ₅ , 1V ₆ | 1V ₂ = 1V ₄ 1V ₃ = 1V ₅ 1V ₄ = 1V ₆ | = 2n | 1 | 2 | n | 2n | 2n | 2n | | | | | | | | | | | | | | |
| 2 | - | 1V ₄ - 1V ₁ | 2V ₄ | 1V ₄ = 1V ₁ | = 2n - 1 | 1 | | | 2n - 1 | | | | | | | | | | | | | | | | |
| 3 | + | 1V ₅ + 1V ₁ | 2V ₅ | 1V ₅ = 1V ₁ | = 2n + 1 | 1 | | | 2n + 1 | | | | | | | | | | | | | | | | |
| 4 | + | 2V ₅ + 2V ₄ | 1V ₁₁ | 1V ₅ = 2V ₅ 1V ₄ = 2V ₄ | 2n - 1 2n + 1 | | | | 0 | 0 | | | | | | | | | | | | | | | |
| 5 | + | 1V ₁₁ + 1V ₂ | 2V ₁₁ | 1V ₁₁ = 2V ₁₁ 1V ₂ = 1V ₂ | 1/2 * 2n - 1 2n + 1 | | 2 | | | | | | | | | | | | | | | | | | |
| 6 | - | 0V ₁₃ - 2V ₁₁ | 1V ₁₃ | 2V ₁₁ = 0V ₁₃ 0V ₁₃ = 1V ₁₃ | - 1/2 * 2n - 1 = A ₀ | | | | | | | | | | | | | | | | | | | | |
| 7 | - | 1V ₈ - 1V ₁ | 1V ₁₀ | 1V ₈ = 1V ₃ 1V ₁ = 1V ₁ | = n - 1 (= 3) | 1 | | n | | | | | | | | | | | | | | | | | |
| 8 | + | 1V ₂ + 0V ₇ | 1V ₇ | 1V ₂ = 1V ₇ 0V ₇ = 1V ₇ | = 2 + 0 = 2 | | 2 | | | | | | | | | | | | | | | | | | |
| 9 | + | 1V ₆ + 1V ₇ | 2V ₁₁ | 1V ₆ = 1V ₆ 1V ₇ = 1V ₁₁ | 2n = A ₁ | | | | | 2n | 2 | | | | | | | | | | | | | | |
| 10 | × | 1V ₂₁ × 2V ₁₁ | 1V ₁₂ | 1V ₂₁ = 1V ₁₁ 2V ₁₁ = 2V ₁₁ | = B ₁ · 2n = B ₁ A ₁ | | | | | | | | | | | | | | | | | | | | |
| 11 | + | 1V ₁₂ + 1V ₁₀ | 2V ₁₃ | 1V ₁₂ = 1V ₁₂ 1V ₁₀ = 2V ₁₃ | = 1/2 * 2n - 1 + B ₁ · 2n | | | | | | | | | | | | | | | | | | | | |
| 12 | - | 1V ₁₀ - 1V ₁ | 1V ₁₀ | 1V ₁₀ = 2V ₁₀ 1V ₁ = 1V ₁ | = n - 2 (= 2) | 1 | | | | | | | | | | | | | | | | | | | |
| 13 | + | 1V ₆ - 1V ₁ | 2V ₆ | 1V ₆ = 1V ₆ 1V ₁ = 1V ₁ | = 2n - 1 | 1 | | | | | 2n - 1 | | | | | | | | | | | | | | |
| 14 | | 1V ₁ + 1V ₂ | 2V ₇ | 1V ₁ = 1V ₁ 1V ₂ = 2V ₇ | = 2 + 1 = 3 | 1 | | | | | 3 | | | | | | | | | | | | | | |
| 15 | | 2V ₆ + 2V ₇ | 1V ₈ | 2V ₆ = 2V ₆ 2V ₇ = 2V ₇ | 2n - 1 | | | | | 2n - 1 | 3 | 2n - 1 | | | | | | | | | | | | | |
| 16 | × | 1V ₈ × 2V ₁₁ | 1V ₁₁ | 1V ₈ = 0V ₈ 2V ₁₁ = 1V ₁₁ | = 2n * 2n - 1 | | | | | | 0 | | | | | | | | | | | | | | |
| 17 | + | 2V ₆ - 1V ₁ | 1V ₆ | 2V ₆ = 1V ₆ 1V ₁ = 1V ₆ | = 2n - 2 | 1 | | | | | 2n - 2 | | | | | | | | | | | | | | |
| 18 | | 1V ₁ + 2V ₇ | 2V ₇ | 1V ₁ = 1V ₁ 2V ₇ = 2V ₇ | = 3 + 1 = 4 | 1 | | | | | 4 | | | | | | | | | | | | | | |
| 19 | | 2V ₆ + 2V ₇ | 1V ₉ | 2V ₆ = 2V ₆ 2V ₇ = 2V ₇ | = 2n - 2 | | | | | 2n - 2 | 4 | 2n - 2 | | | | | | | | | | | | | |
| 20 | × | 1V ₉ × 4V ₁₁ | 1V ₁₁ | 1V ₉ = 0V ₉ 4V ₁₁ = 1V ₁₁ | = 2n * 2n - 1 * 2n - 2 = A ₃ | | | | | | 0 | | | | | | | | | | | | | | |
| 21 | × | 1V ₂₂ × 2V ₁₁ | 1V ₁₂ | 1V ₂₂ = 1V ₂₂ 2V ₁₁ = 2V ₁₁ | = B ₃ · 2n * 2n - 1 * 2n - 2 = B ₃ A ₃ | | | | | | | | | | | | | | | | | | | | |
| 22 | + | 2V ₁₂ + 2V ₁₃ | 2V ₁₃ | 2V ₁₂ = 0V ₁₂ 2V ₁₃ = 2V ₁₃ | = A ₀ + B ₁ A ₁ + B ₃ A ₃ | | | | | | | | | | | | | | | | | | | | |
| 23 | - | 2V ₁₀ - 1V ₁ | 1V ₁₀ | 2V ₁₀ = 2V ₁₀ 1V ₁ = 1V ₁ | = n - 3 (= 1) | 1 | | | | | | | | | | | | | | | | | | | |
| Here follows a repetition of Operations thirteen to twenty-three. | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | + | 4V ₁₃ + 0V ₂₁ | 1V ₂₄ | 4V ₁₃ = 0V ₁₃ 0V ₂₁ = 1V ₂₄ | = B ₇ | | | | | | | | | | | | | | | | | | | | |
| 25 | + | 1V ₁ + 1V ₂ | 1V ₃ | 1V ₁ = 1V ₁ 1V ₂ = 1V ₂ | = n + 1 = 4 + 1 = 5 | 1 | | n + 1 | | | 0 | 0 | | | | | | | | | B ₇ | | | | |



*"We may say most aptly that
the Analytical Engine weaves
algebraic patterns just as the
Jacquard-loom weaves
flowers and leaves."*

Ada Lovelace (1815 - 1852)



By ArtProf - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=18564683>



*"[Debugging] is damnably
troublesome work, and
plagues me."*

Ada Lovelace (1815 - 1852)

A moving image of the world



*"From a distance, everyone
will be able to read text,
enlarged and limited to the
desired subject, projected on
an individual screen."*

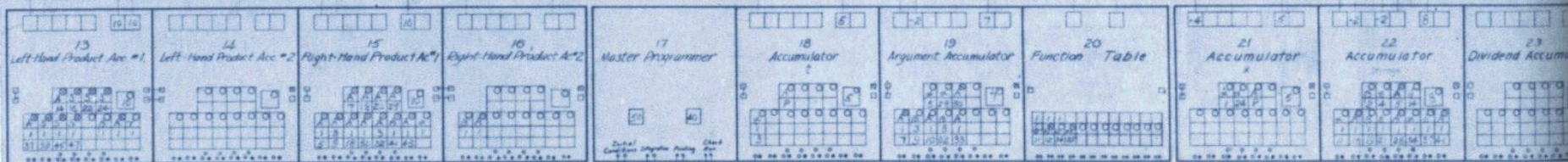
Paul Otlet (1868 - 1944)





By Unidentified U.S. Army photographer - Image from Historic Computer Images, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=26253110>

DRAWING NUMBER PX-1-82 PANEL DIAGRAM OF THE ELECTRONIC NUMERICAL INTEGRATOR AND COMPUTER (SHOWING THE EXTERIOR BALLISTICS EQUATIONS SETUP - HE



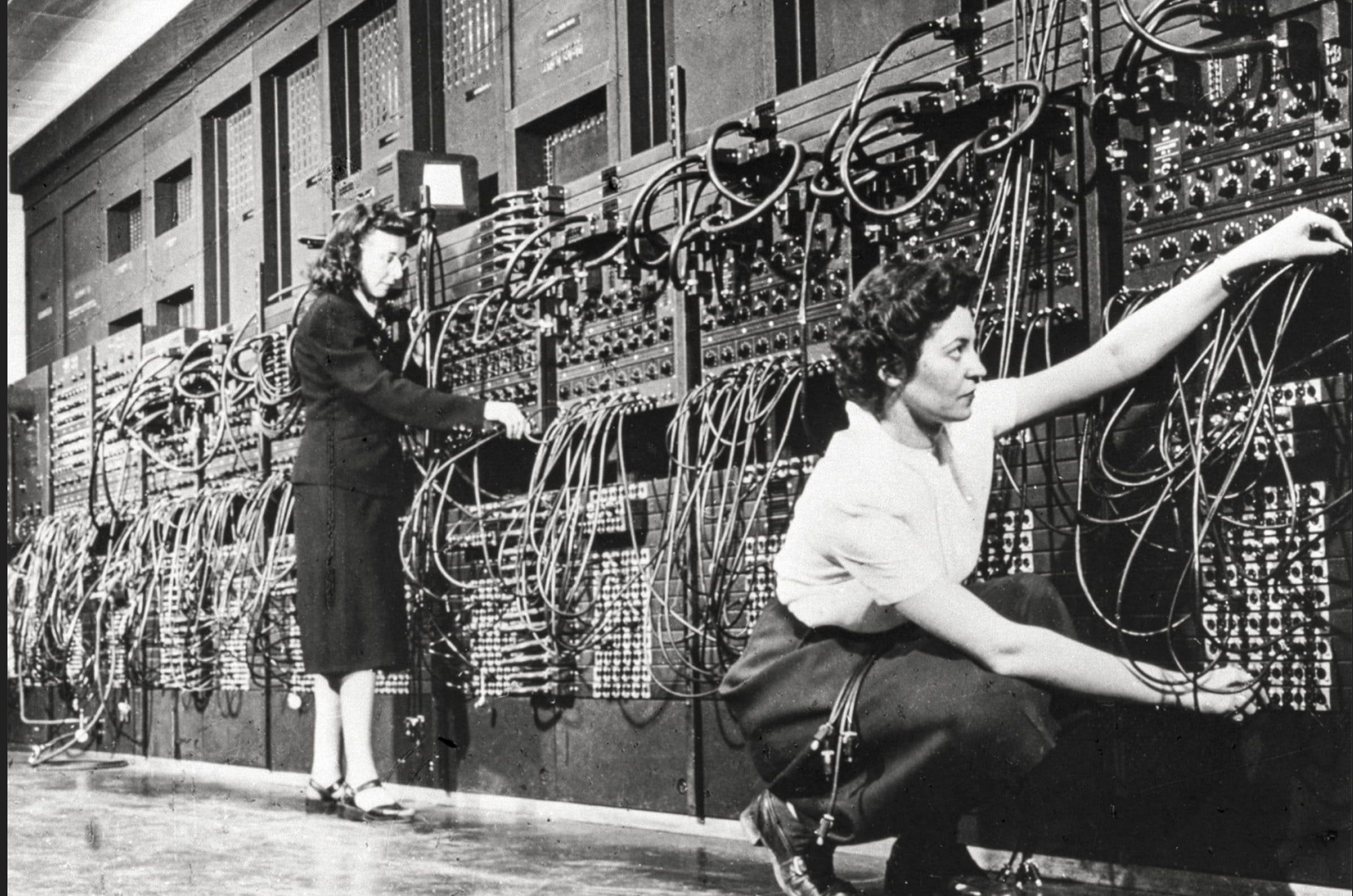
The "ENIAC Six" -- 1946



- *Betty Snyder Holberton*
- *Jean Jennings Bartik*
- *Kay McNulty Mauchly Antonelli*
- *Marlyn Wescoff Meltzer*
- *Ruth Lichterman Teitelbaum*
- *Frances Bilas Spence*

"The Computers: The Remarkable Story of the ENIAC Programmers" -- Kathy Kleimar

<https://vimeo.com/ondemand/eniac6>



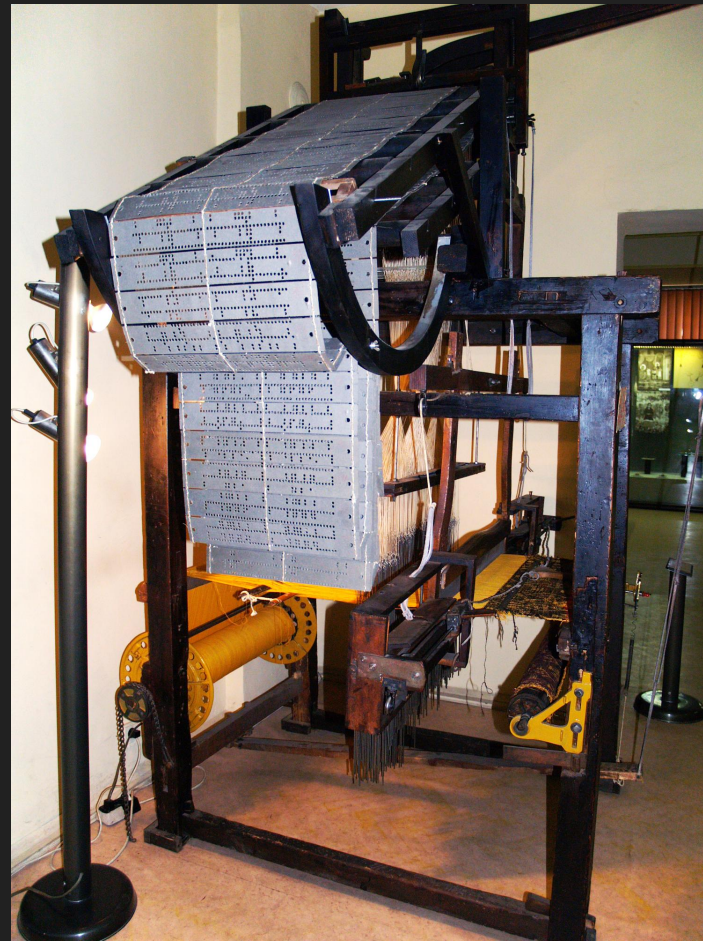


Dorothy Vaughan (1910 - 2008)



"We're going to need a lot of manpower to program that beast [the IBM 7090]."

*-- Dorothy Vaughan
Mathematician
FORTRAN expert*



By Edal Anton Lefterov - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=11959365>

Where We Are

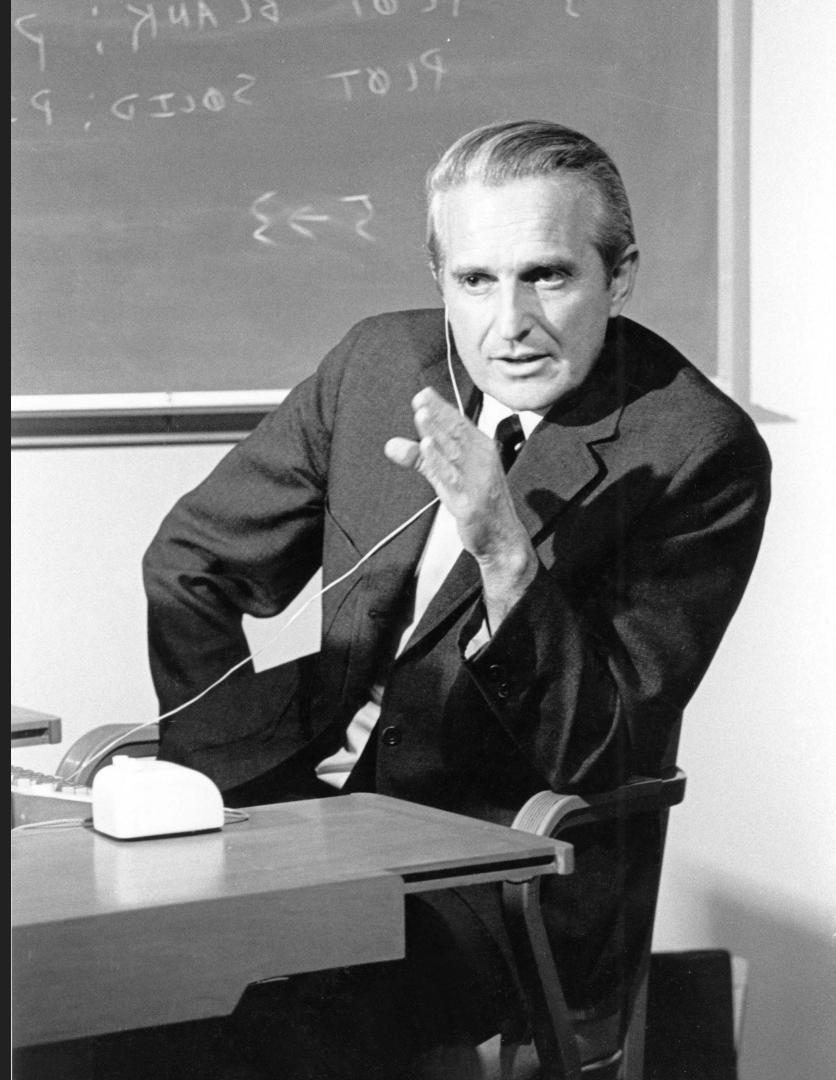
1950-2050



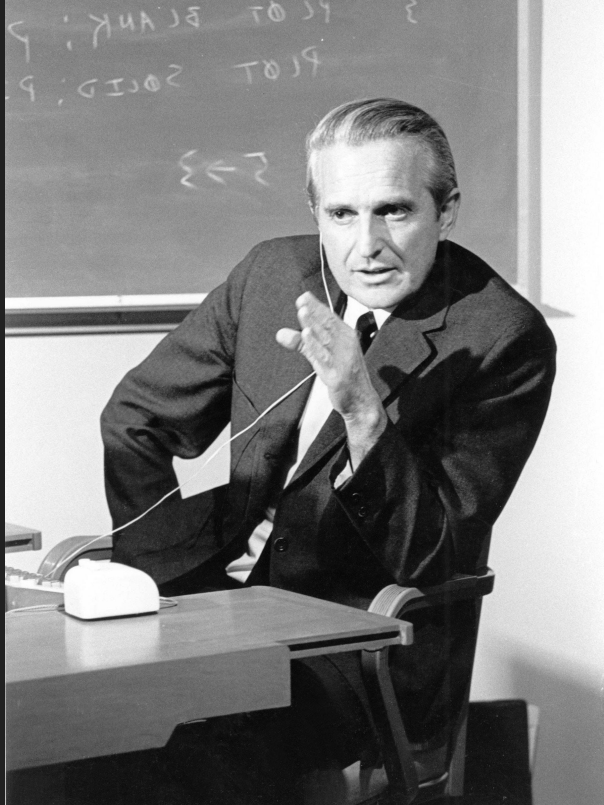
12:30
08:37:20



Demo Day 1968

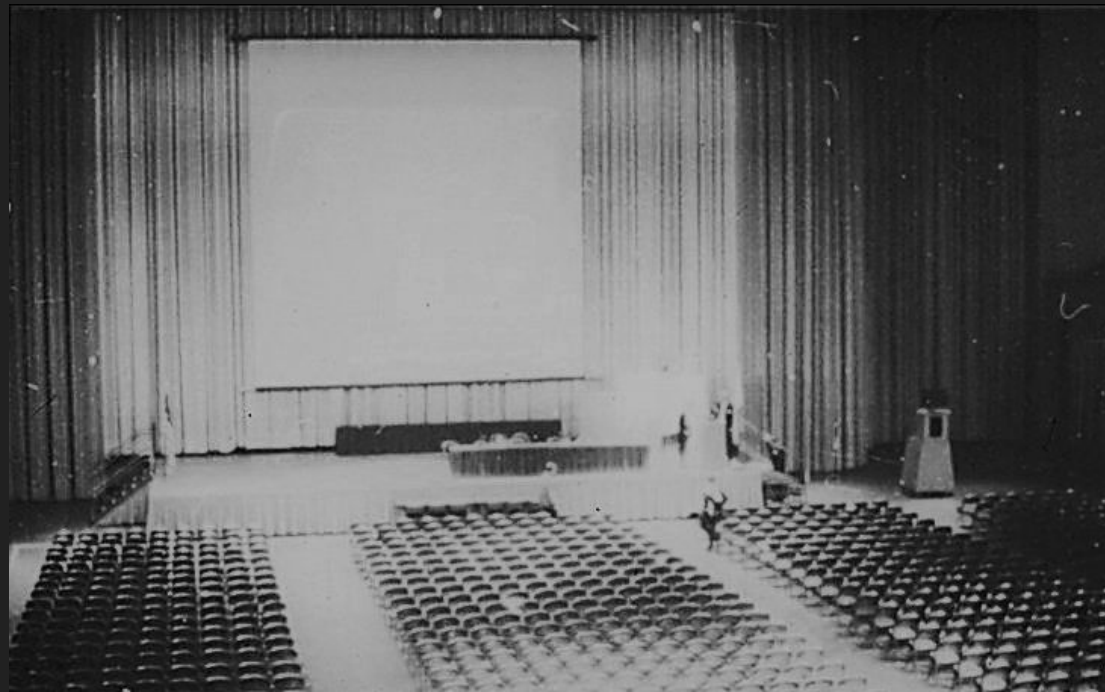


Technology should not aim to replace humans, rather amplify human capabilities.



"In 20 or 30 years, you'll be able to hold in your hand as much computing knowledge as exists now in the whole city, or even the whole world."

— Doug Engelbart (1925 - 2013)



APPLES
&
BANANAS
CARROTS
LETTUCE
BEANS

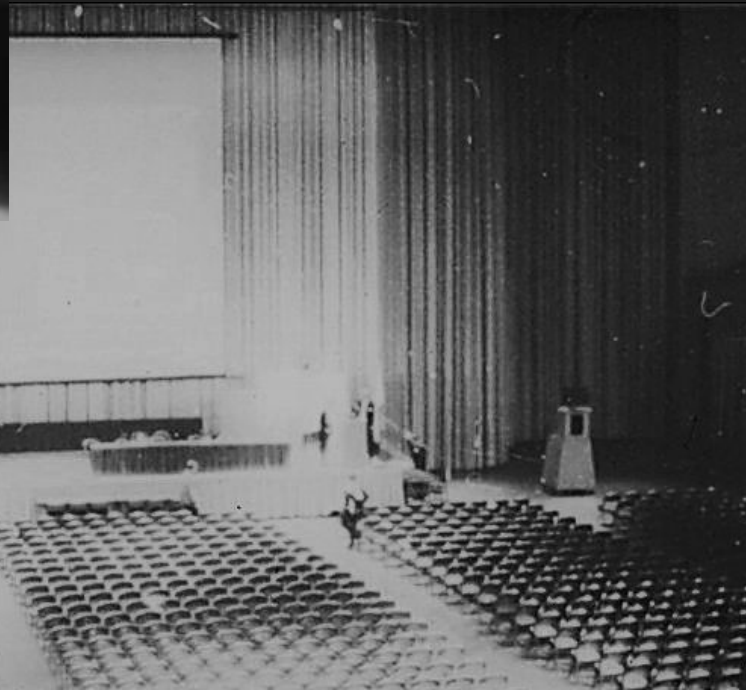
GANG

APPLE SAUCE
BEAN SOUP
TOMATO SOUP

CEREALS

BREAD
NOODLES (ELBOW KIND)
FRENCH BREAD

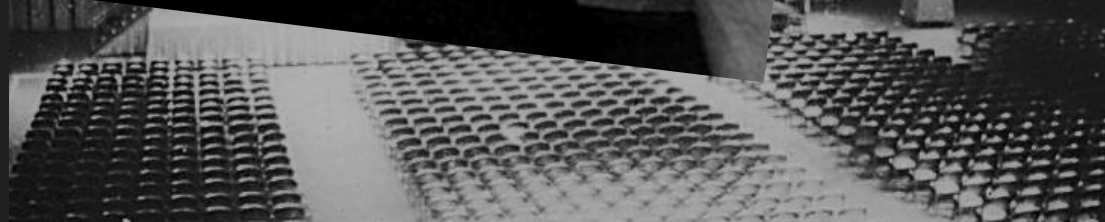
COLD LOCKER
MILK

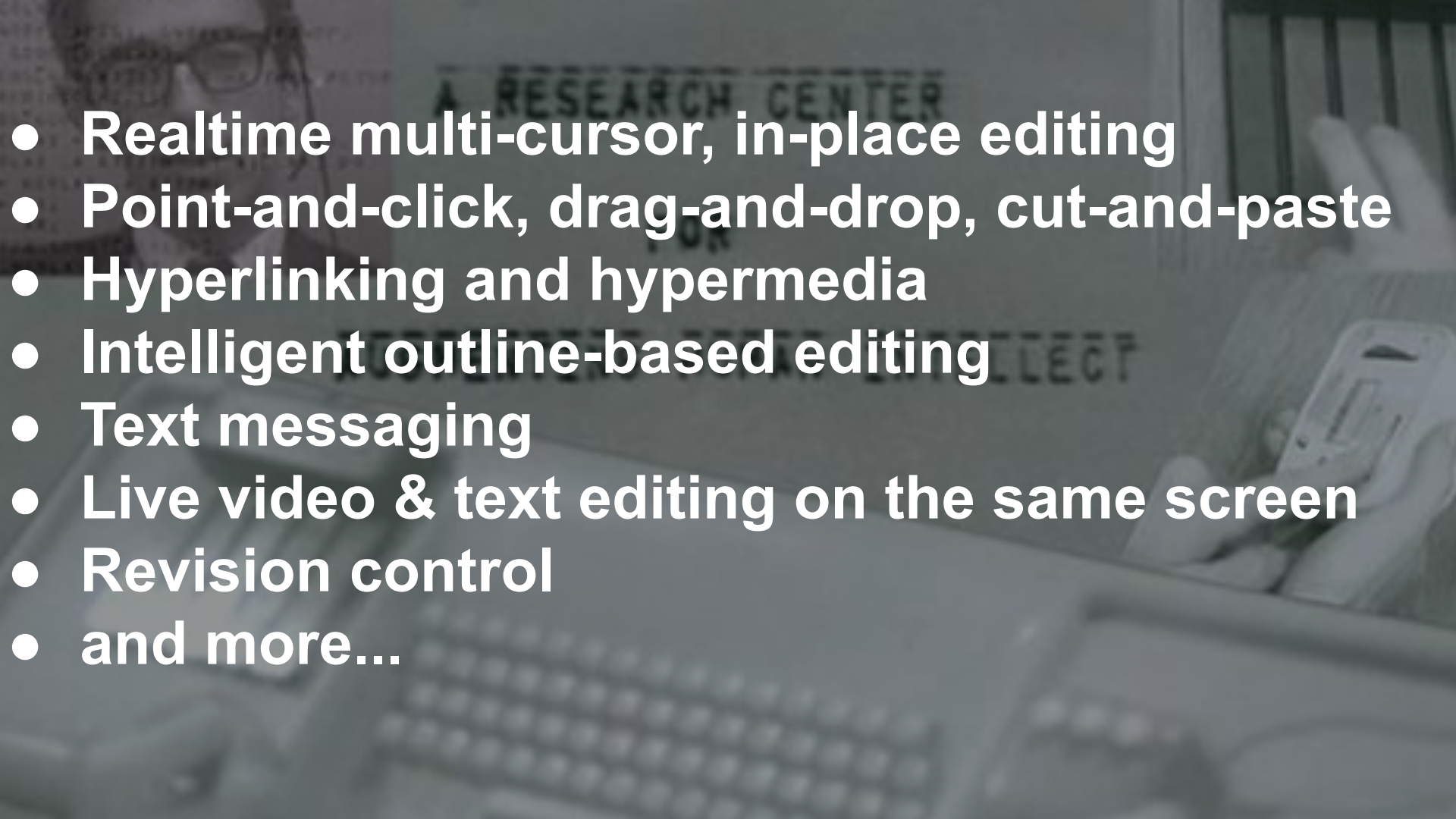


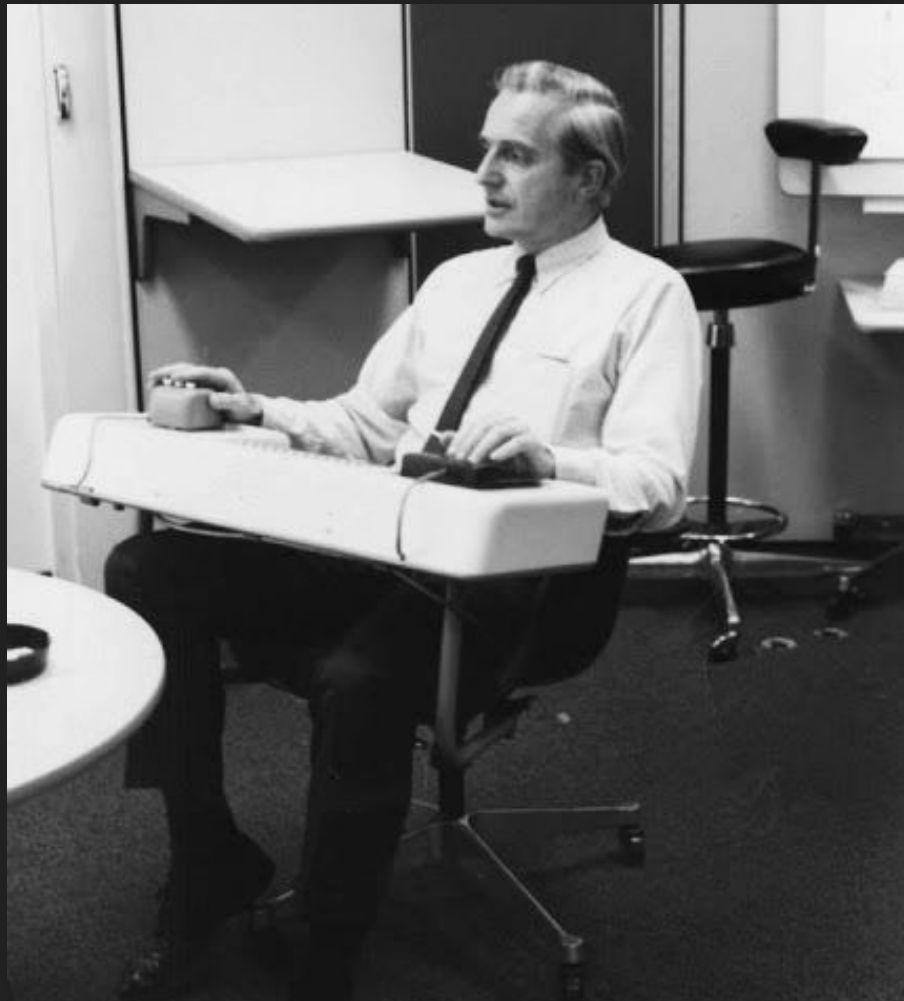
APPLES
&
BANANAS
CARROTS
LETTUCE
BEANS

GANG

APPLE SAUCE
BEAN SOUP
TOMATO SOUP
CEREALS
BREAD
NOODLES
FRENCH BREAD
COLD LOCKER
MILK



- 
- **Realtime multi-cursor, in-place editing**
 - **Point-and-click, drag-and-drop, cut-and-paste**
 - **Hyperlinking and hypermedia**
 - **Intelligent outline-based editing**
 - **Text messaging**
 - **Live video & text editing on the same screen**
 - **Revision control**
 - **and more...**



We should not impose regularity.

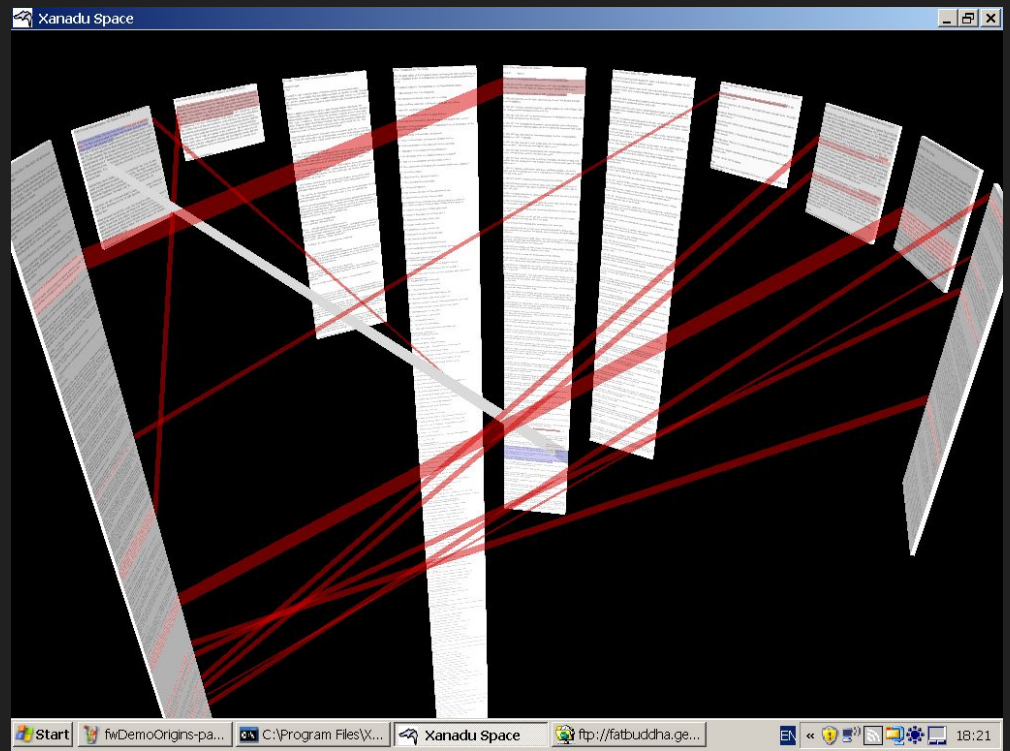
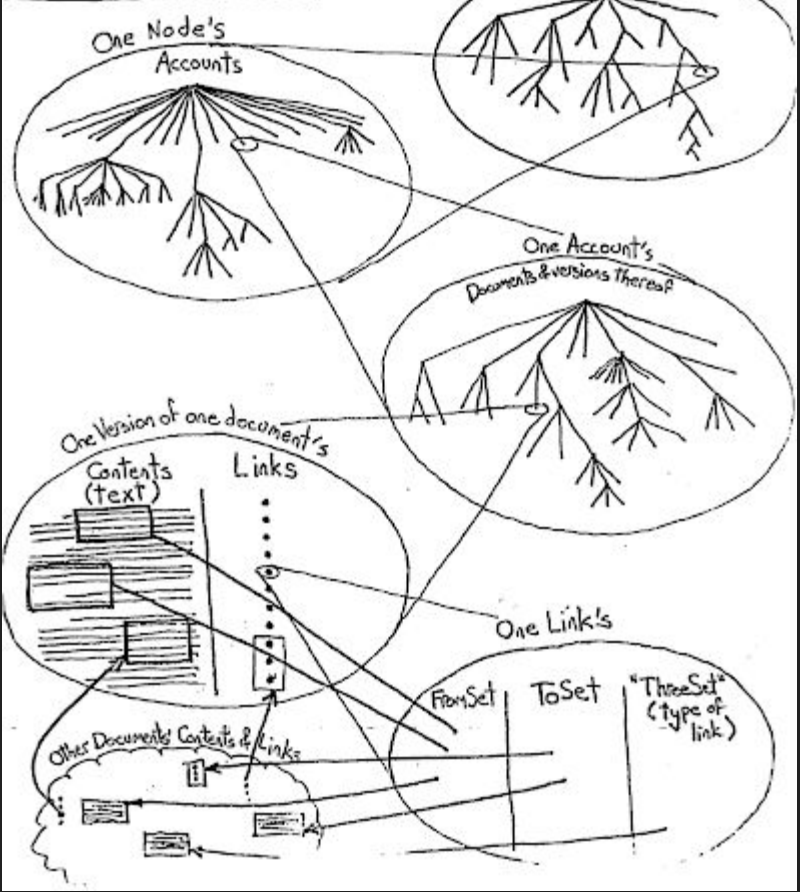


"The point was to be able to have a medium that would record all the connections and all the structures and all the thoughts that paper could not."

-- Ted Nelson (1937 --)

The Structure of the Xanadu™ Docverse

©1983 Steve Wilton



You can and must understand computers NOW.

COMPUTER



DREAM MACHINES



New Freedoms Through Computer Screens
— a Minority Report

This is the flip side of Computer Lib.

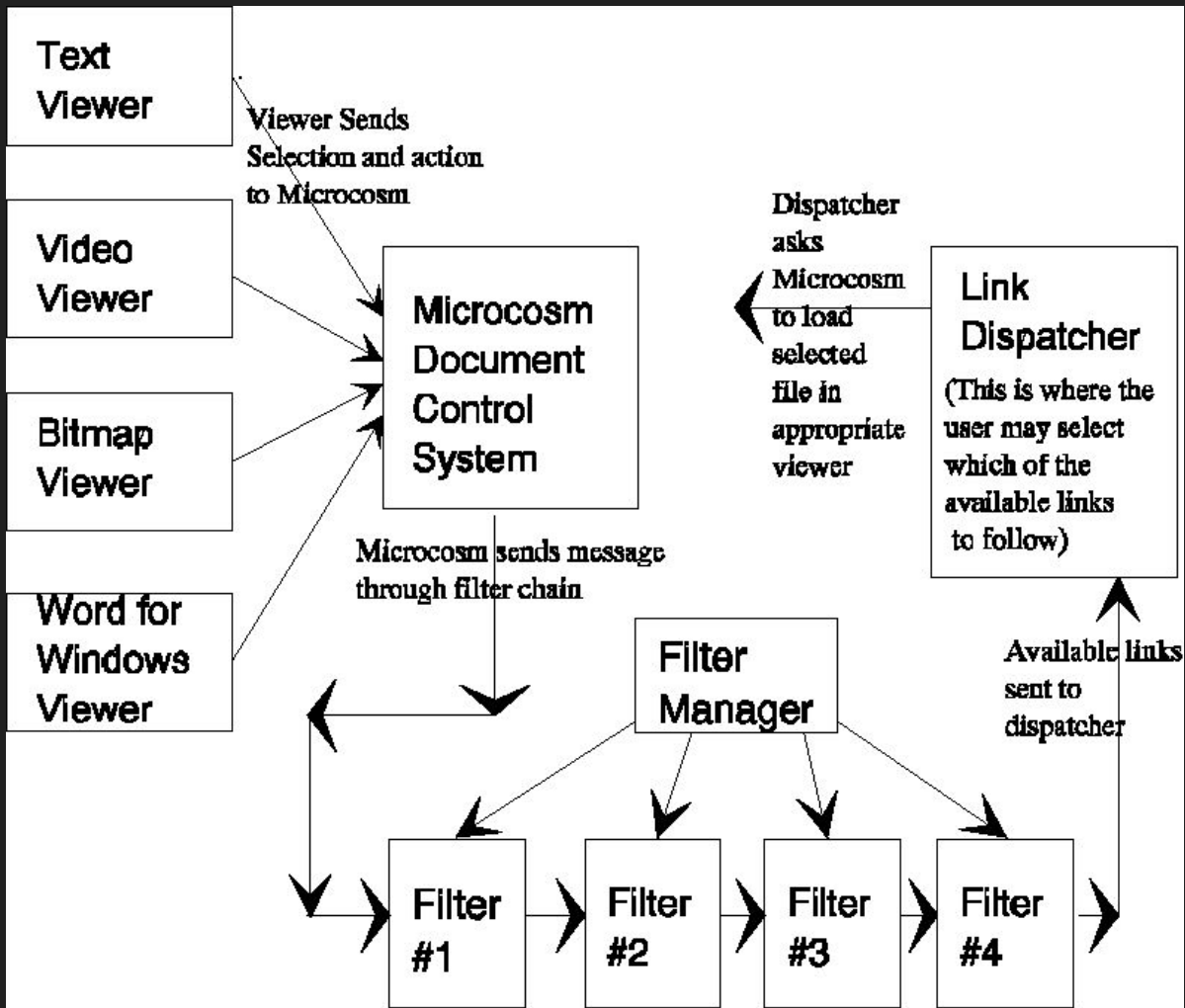
The whole foundation of hypertext is collaborative



*"The strength of the internet is
its global interconnection."*

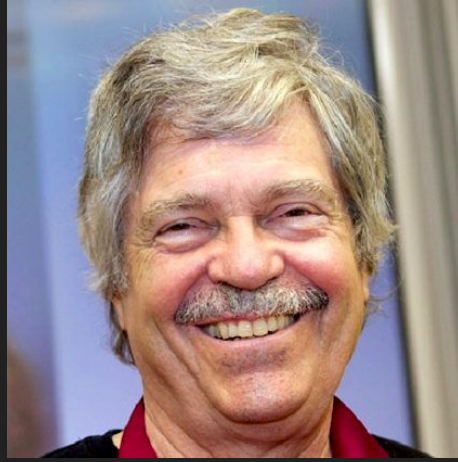
-- Dame Wendy Hall (1952 --)



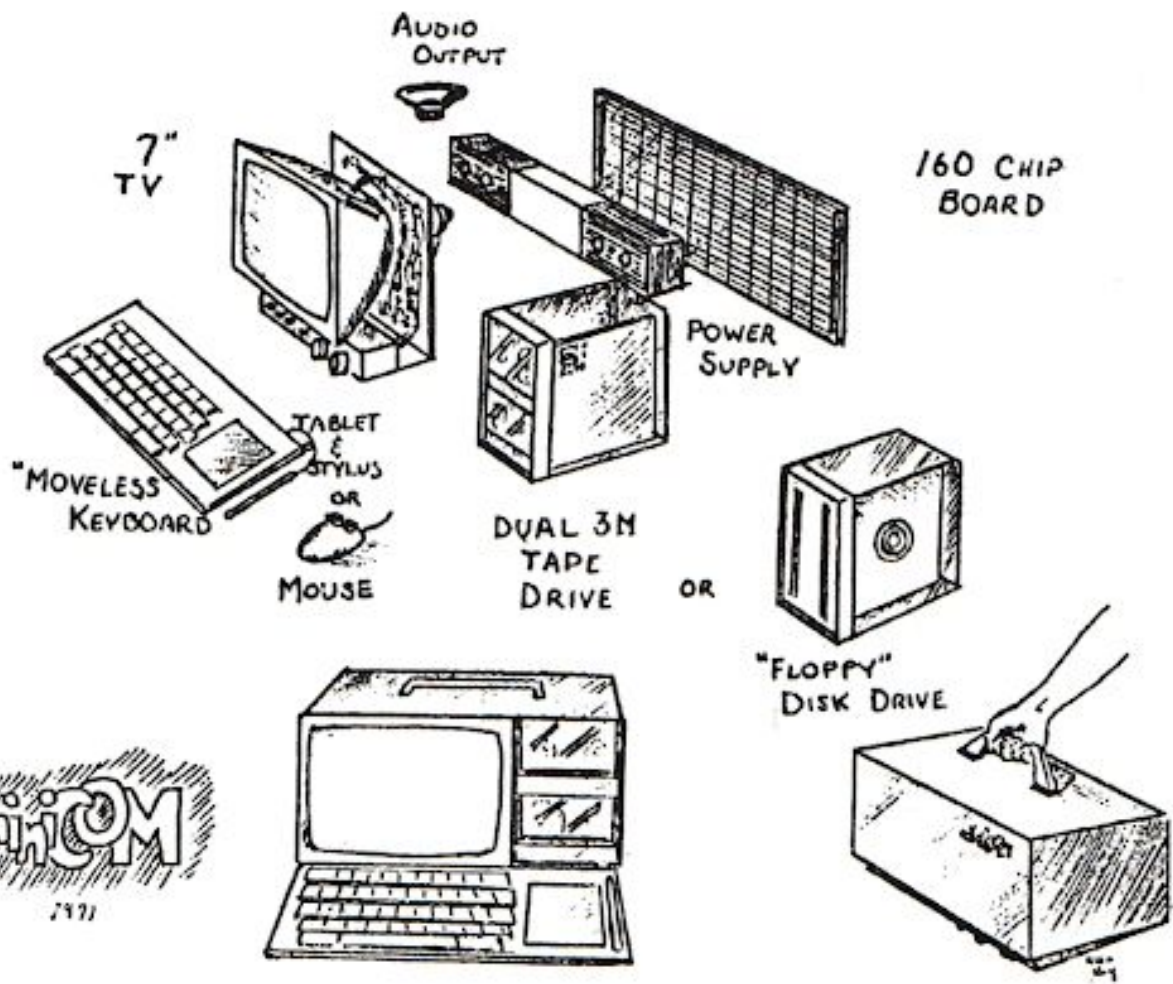




Adele Goldberg & Alan Kay -- 1970s

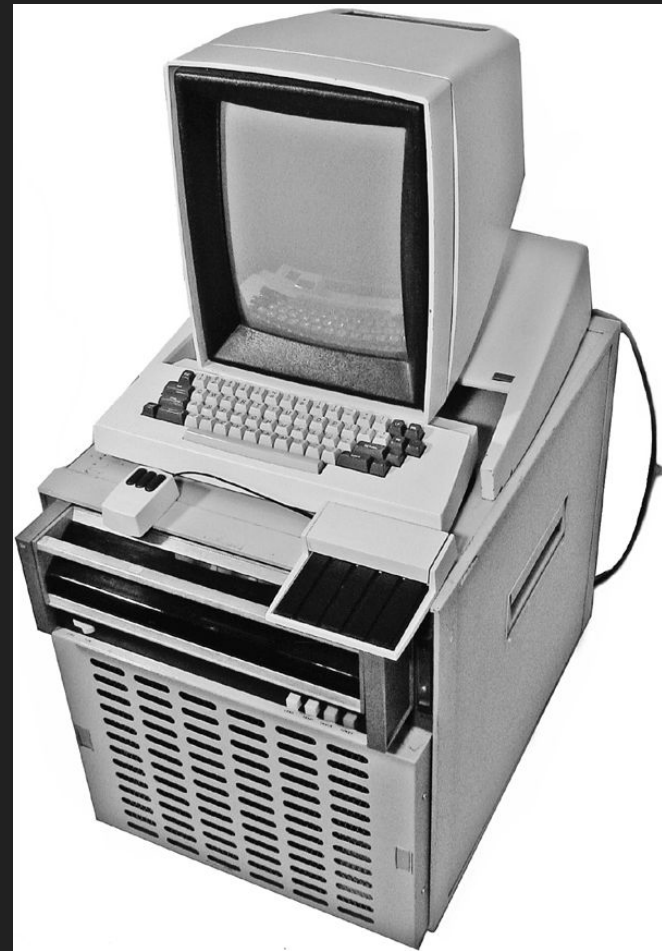
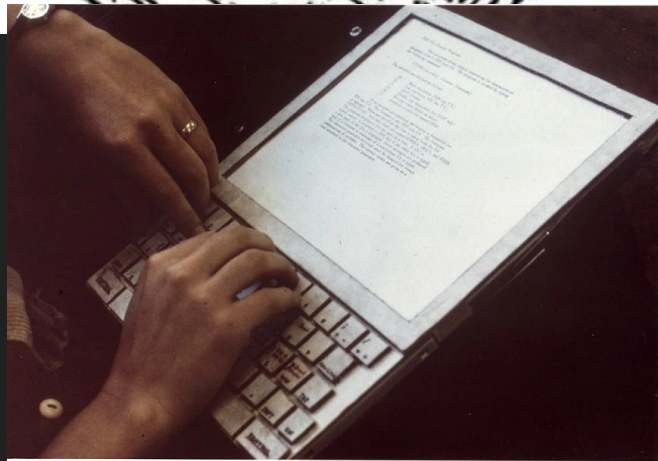


- *XEROX Parc*
- *Smalltalk Language*
- *Object-Oriented Programming*
- *Dynabook*



minicom

2971



```

5 with Text_IO; use Text_IO;
6 with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
7
8 procedure bernoulli is
9   Bern: Integer:= 4;
10
11 begin
12   Put_Line("Which B(n) would you like to compute:");
13   Get(Bern);
14
15   declare
16     N: array(1..Bern+1) of Integer;
17     D: array(1..Bern+1) of Integer;
18     new_denom: Integer;
19     gcd_result: Integer;
20
21     function GCD (X, Y: Integer) return Integer is
22
23       X1: Integer:= X;
24       Y1: Integer:= Y;
25       Old_X : Integer;
26
27       begin
28
29         while (Y1 /= 0) loop
30           -- x, y := y, x mod y
31           Old_X := X1;
32           X1 := Y1;
33           Y1 := Old_X mod Y1;
34         end loop;
35
36         return X1;
37
38       end GCD;
39


```


Wikipedia, the free e... x +

en.wikipedia.org/wiki/Main_Page

New features Log in / create account

Main Page Discussion Read View source View history Search


WIKIPEDIA
 The Free Encyclopedia

Main page
 Contents
 Featured content
 Current events
 Random article

Interaction
 About Wikipedia
 Community portal
 Recent changes
 Contact Wikipedia
 Donate to Wikipedia
 Help


Toolbox
 Print/export

Languages
 Simple English
 العربية
 Bahasa Indonesia
 Bahasa Melayu
 Български
 Català
 Česky
 Dansk
 Deutsch
 Eesti
 Ελληνικά
 Español

Welcome to Wikipedia,
 the free encyclopedia that anyone can edit.
 3,421,206 articles in English

- Arts
- History
- Society
- Biography
- Mathematics
- Technology
- Geography
- Science
- All portals

Today's featured article


The SR Leader class was a class of experimental 0-6-6-0 articulated steam locomotive, produced to the design of the innovative engineer **Oliver Bulleid**. Intended as a replacement for the ageing fleet of **M7 class**, the Leader was an attempt to extend the life of steam traction on the **Southern Railway** by eliminating many of the operational drawbacks associated with existing steam locomotives. Design work began in 1946, and development continued after the nationalisation of the railways in 1948, under the auspices of **British Railways**. The Leader project was part of Bulleid's desire to modernise the steam locomotive based on experience gained with the Southern Railway's fleet of electric stock. The design incorporated many novel features, such as the use of **thermic siphons**, **bogies**, and cabs at either end of the locomotive, resulting in its unique appearance. Several of its innovations proved to be unsuccessful however, partly accounting for the project's cancellation in the early 1950s. Five Leader locomotives were begun, although only one was completed. Problems with the design, indifferent reports on performance, and political pressure surrounding spiralling development costs, led to all locomotives of the class being scrapped by 1951. **(more...)**


Recently featured: *Sherlock Holmes Baffled* – Australian Magpie – Terry Fox

[Archive](#) – [By email](#) – [More featured articles...](#)

Did you know...

From Wikipedia's newest articles:

- ... that the Army general Maxwell Woodhull donated his **family home** (*pictured*) to George Washington University in **Washington, D.C.**?



In the news



- Italian authorities seize €23 million in assets from the **Institute for Works of Religion** in Vatican City amidst an investigation of alleged money laundering.
- The King's Speech***, a historical drama film starring Colin Firth (*pictured*), wins the People's Choice Award at the 2010 Toronto International Film Festival.
- In the **Swedish general election**, the centre-right Alliance wins a plurality, while the far-right Sweden Democrats hold the balance of power.
- In Gaelic football, **Cork defeat Down** to win their seventh All-Ireland Senior Football Championship.
- French athlete **Philippe Croizon** becomes the first quadruple amputee to swim the English Channel, completing the challenge in less than 14 hours.
- In cycling, Italian Vincenzo Nibali **wins** the Vuelta a España.

Wikinews – Recent deaths – More current events...

On this day...

September 22: Mid-Autumn Festival in the Chinese lunar calendar (2010); **Sukkot** begins at sunset (Judaism, 2010); Independence Day in **Bulgaria** (1908) and **Mali** (1960); **Car-Free Day** in Europe and Canada; **OneWebDay**

- 1792 – The epoch of the **French Republican Calendar** occurred, marking the first full day of





2:34PM
SUNNY 71 .6F/22C
GPS / WI-FI

ROAMING OFF ON

1 2 3 4 5 6 7 8 9 0 DELETE 7 8 9
Q W E R T Y U I O P ENTER 4 5 6
S D F G H J K L SHIFT 1 2 3
Z X C V B N M CAPSLOCK CTRL SYMBOLS
SPACE FUNCTION

SERVICES
PING
ING
RITY
TRACKING
ORKPLACE

Where We Will Be

2050 - ??

Let's Review...

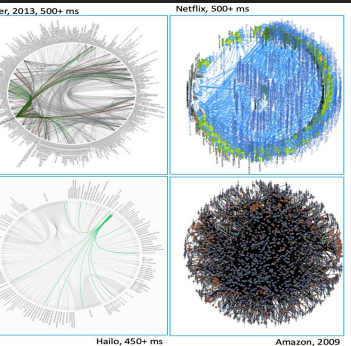
- Written instructions in tables (Lovelace) for mechanical computers (Babbage)
- Wiring diagrams (McNulty, etc.) for electrical relay computers (ENIAC)
- Punch cards (Vaughn) for electronic computers (IBM)
- Screen text (Engelbart) for connected semiconductor computers (Hall)



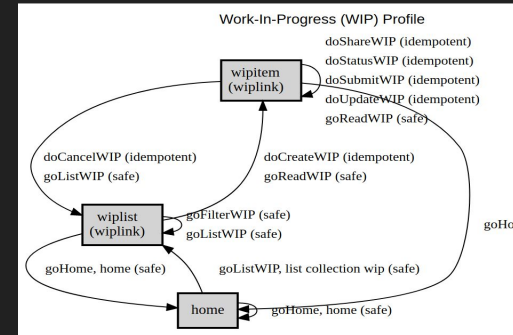
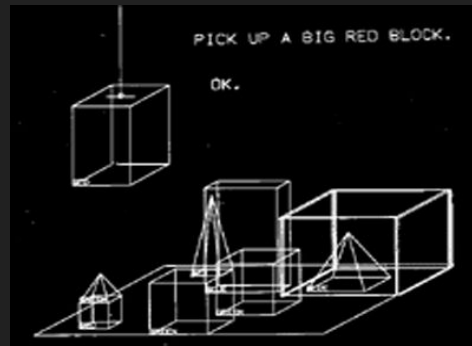
What's Next?

In the Future we will...

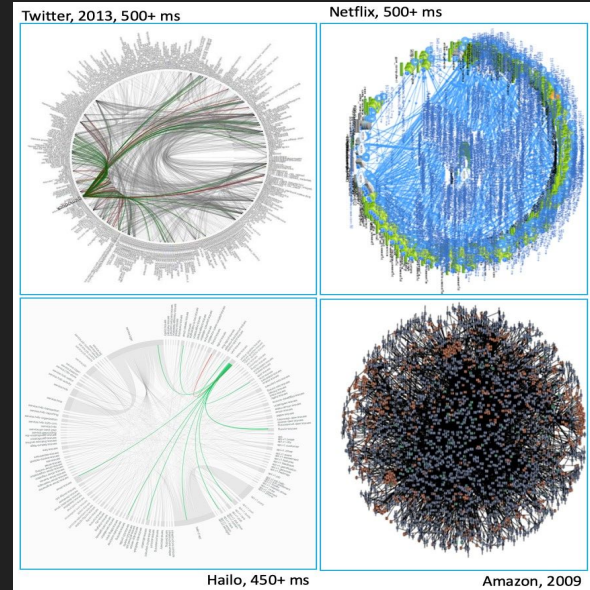
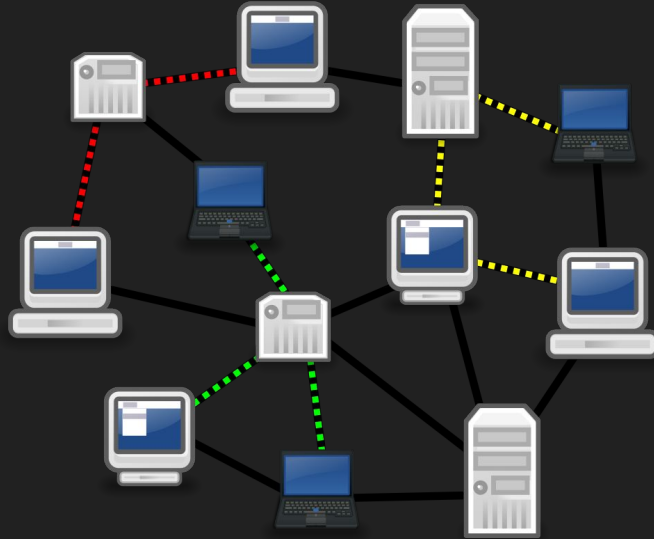
- Connect services the way we currently connect servers
- Program the network instead of single machines
- Describe problem spaces, not solutions
- Machines, endpoints, protocols and formats will "disappear"



```
1 # Get weekly groceries
2 #
3 #
4 # load control data
5 CONFIG LOAD grocery.config
6
7
8 # find running services for shopping, shipping, & payment
9 REQUEST WITH-URL $$service-registry$$
10 CALL WITH-FORM search WITH-PARAMS {"name": "shopping-cart"}
11 STACK PUSH WITH-RESPONSE cart
12 CALL WITH-FORM search WITH-PARAMS {"name": "shipping"}
13 STACK PUSH WITH-RESPONSE shipping
14 CALL WITH-FORM search WITH-PARAMS {"name": "payment"}
15 STACK PUSH WITH-RESPONSE payment
16
17 # load the cart, set delivery, and pay
18 CALL WITH-NAME fill-cart WITH-CONFIG $$shopping-items$$
19 CALL WITH-NAME shipping-address WITH-CONFIG $$home-address$$
20 CALL WITH-NAME submit-payment WITH-CONFIG $$payment-profiles$$
21
22 # all done
23 ECHO "Groceries will arrive on $$shipping-date$$!"
```



From Connected Machines to Connected Services



Programming the Network Instead of Single Machines

Languages for Software-Defined Networks

Nate Foster*, Michael J. Freedman†, Arjun Guha*, Rob Harrison‡,
Naga Praveen Katta†, Christopher Monsanto†, Joshua Reich†, Mark Reitblatt*,
Jennifer Rexford†, Cole Schlesinger†, Alec Story*, and David Walker†
*Cornell University †Princeton University ‡U.S. Military Academy

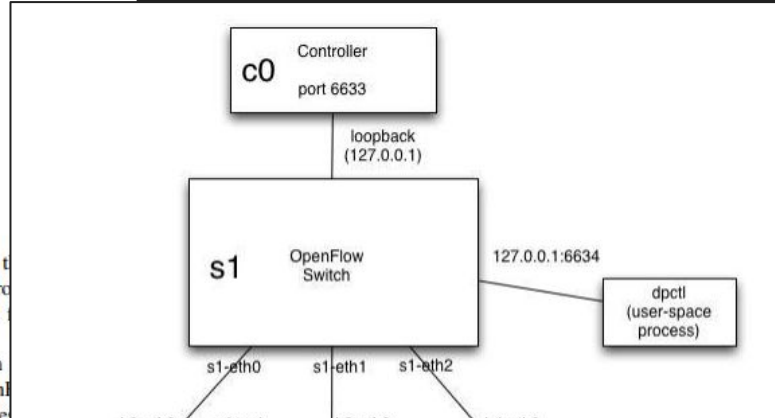
Abstract—Modern computer networks perform a bewildering array of tasks, from routing and traffic monitoring, to access control and server load balancing. Yet, managing these networks is unnecessarily complicated and error-prone, due to a hetero-

forwarding paths for each user [5]. To balance the load between back-end servers in a data center, the controller splits flows over several server replicas and migrates them [6], [7].

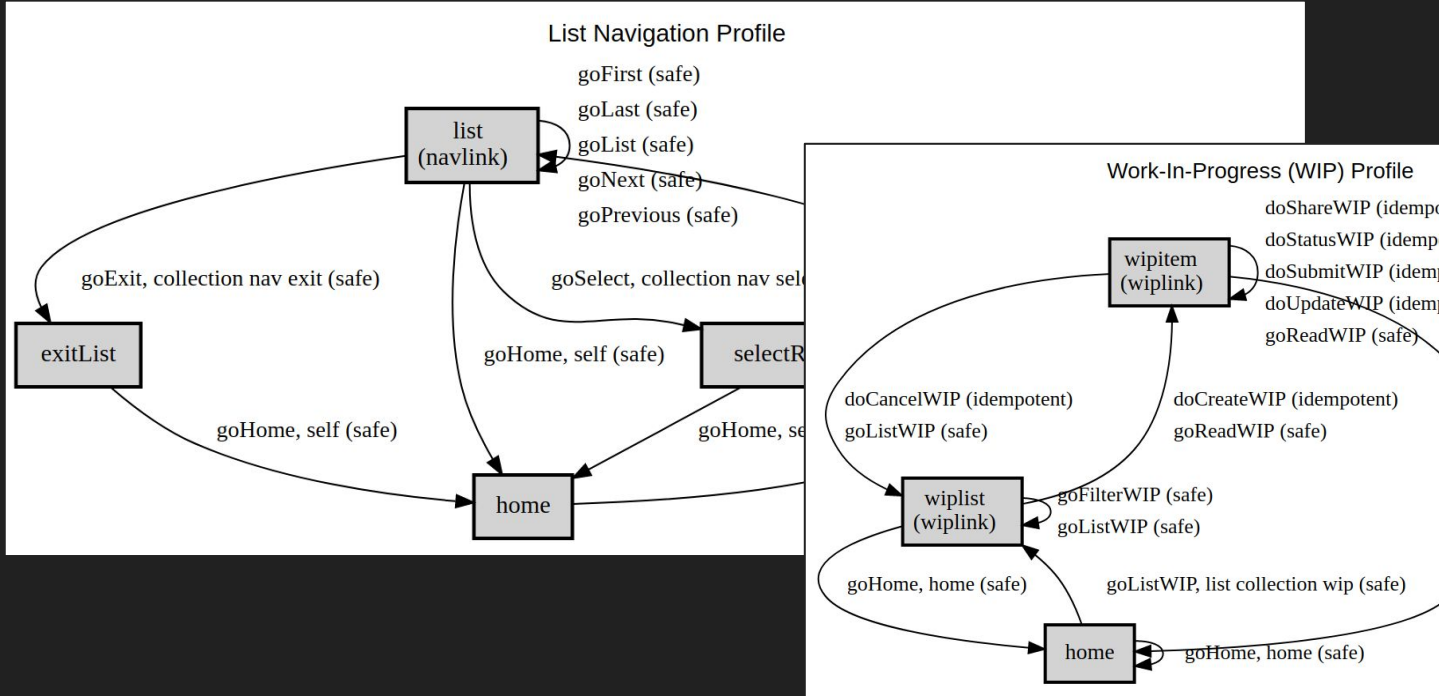
```
- hosts:
  - pc1.example.com
  - pc3.example.com
tasks:
  - name: install Apache
    action: apt pkg=apache2 state=present
  - name: ensure Apache is running
    action: service name=apache2 state=running
- hosts: dns_servers
roles:
  - dns_server
  - ntp
```

```
# SIREN example
GOTO http://rwcbook10.herokuapp.com
SIREN LINKS
SIREN ENTITIES
SIREN ACTIONS

GOTO WITH-REL taskFormListByUser WITH-QUERY {"assignedUser" : "alice"}
```



Focus on Describing Problem Spaces, not Solutions



```
1 {
2 "$schema": "https://alps-io.github.io/schemas/
3 "alps":
4 {
5   "version": "1.0",
6   "title": "List Navigation Profile",
7   "doc": {"value": "List Navigation profile fo
8 records, services [SHOULD](https://www.rfc-edito
9 client move forward and back in the list, select
10 "descriptor": [
11   {"id": "href", "type": "semantic", "def":
12     "title": "URL of the link.",
13     "doc": {"value": "The URL of the navigat
14     "tag": "ontology"
15   },
16   {"id": "rel", "type": "semantic", "def":
17     "title": "Rel value of the link.",
18     "doc": {"value": "The link relation valu
19     "tag": "ontology"
20   },
21   {"id": "title", "type": "semantic", "def":
22     "title": "Title text of the link.",
23     "doc": {"value": "Human readable title a
24     "tag": "ontology"
25   },
26   {"id": "type", "type": "semantic", "def":
27     "title": "Media type for the link (option
28     "doc": {"value": "Optional media type id
29     "tag": "ontology"
30   },
31   {"id": "navlink", "type": "semantic",
32     "title": "List navigation link object.",
33     "descriptor": [
34       {"href": "#href"},
35       {"href": "#rel"},
36       {"href": "#title"},
37       {"href": "#type"}
38     ],
39     "doc": {"value": "Represents one of man
```


Machines, Endpoints, Protocols, Formats will "disappear"

```
1 #
2 # Get weekly groceries
3 #
4
5 # load control data
6 CONFIG LOAD grocery.config
7
8 # find running services for shopping, shipping, & payment
9 REQUEST WITH-URL $$service-registry$$
10 CALL WITH-FORM search WITH-PARAMS {"name": "shopping-cart"}
11 STACK PUSH WITH-RESPONSE cart
12 CALL WITH-FORM search WITH-PARAMS {"name": "shipping"}
13 STACK PUSH WITH-RESPONSE shipping
14 CALL WITH-FORM search WITH-PARAMS {"name": "payment"}
15 STACK PUSH WITH-RESPONSE payment
16
17 # load the cart, set delivery, and pay
18 CALL WITH-NAME fill-cart WITH-CONFIG $$shopping-items$$
19 CALL WITH-NAME shipping-address WITH-CONFIG $$home-address$$
20 CALL WITH-NAME submit-payment WITH-CONCFIG $$payment-profile$$
21
22 # all done
23 ECHO "Groceries will arrive on $$shipping-date$$!"
24
```

And So...

```
3286495567958
static char cpu[30000];
// some more code here, including parsing and so on...
// ... and finally, we return the result, with a final
// declaration of the array, which is not in the scope.
```

```
#include <iostream>
#include <fstream>
#include <vector>
```

```
#include <iostream>
#include <fstream>
#include <vector>
```

```
using namespace std;
```

```
static char cpu[30000];
```

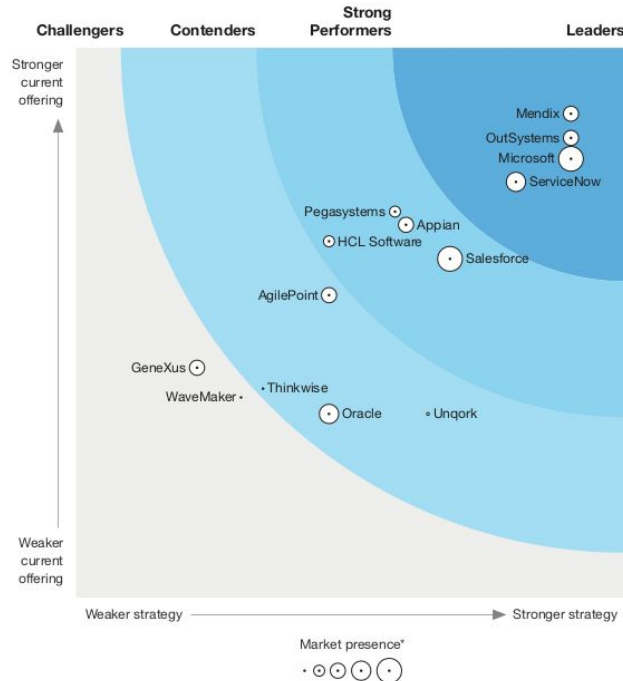
```
int main(int argc, char **argv) {
    vector<char> acc;
    char ch;
    ifstream in("input.txt");
```

What can you do right now?

Low-Code & Automation -- *Be Prepared*

THE FORRESTER WAVE™

Low-Code Development Platforms For Professional Developers
Q2 2021

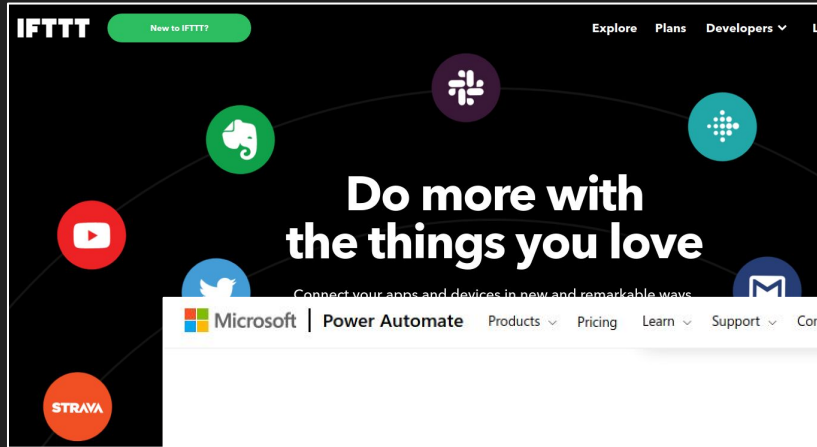


Magic Quadrant

Figure 1. Magic Quadrant for Robotic Process Automation



Decentralized Orchestration -- *Start Exploring*



IFTTT New to IFTTT Explore Plans Developers

Do more with the things you love

Connect your apps and devices in new and remarkable ways

Microsoft | Power Automate Products Pricing Learn Support

STRAVA

Automate whatever slows you down with Zaps

Zaps handle work for you automatically, so you can focus on things that matter. You can create and customize Zaps in minutes.

[Know instantly](#) [Track anything](#) [Connect to Customers](#)

USE ZAPIER TO:

Get notifications about new form submissions

Know the instant someone fills out your form—whether it's customer feedback, a work request, or something else. A simple Zap helps you act fast.

Used by 358.4k people

[Connect Typeform to Slack >](#)

When **Someone fills out my form**

Do this **Send my team a notification**

Automate anywhere. Anytime.

Expand your automation capabilities across desktop, web, and mobile with Power Automate apps.



Power Automate for desktop



Power Automate for mobile

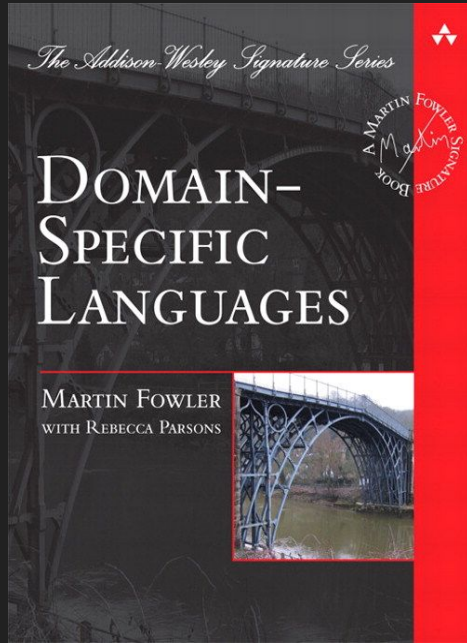


Power Automate for web



Power Automate for Teams

Domain-Specific Languages -- *Stay Tuned*



- Parsons & Fowler book
- DevOps spawned several DSL
- RPA and Low-Code will create more
- Think :
 - GDPR-lang
 - HIPPA-lang
 - BIAN-lang
 - FHIR-lang
 - ACCORD-lang

AI, Chatbots, and Speech -- *Be Wary!*

Welcome to

```
EEEEEE LL      IIII ZZZZZZZ AAAAA
EE      LL      II      ZZ  AA  AA
EEEEEE LL      II      ZZZ  AAAAAA
EE      LL      II      ZZ  AA  AA
EEEEEE LLLLLL IIII ZZZZZZZ AA  AA
```

Eliza is a mock Rogerian psy
The original program was des
This implementation by Norbe

```
ELIZA: Is something troubling
YOU:  Men are all alike.
ELIZA: What is the connection,
YOU:  They're always bugging
ELIZA: Can you think of a spec
YOU:  Well, my boyfriend made
ELIZA: Is it important to you
YOU:  He says I'm depressed r
ELIZA: I am sorry to hear that
YOU:  It's true. I am unhappy
ELIZA: Can you explain what me
YOU:  █
```



PICK UP A BIG RED BLOCK.

Network Working Group
Request for Comments: 439
NIC: 13771

V. Cerf
SU-ERL
21 January 1973

PARRY Encounters the DOCTOR

The session recorded below actually happened on 18 September 1972 and
is herewith reproduced for

Good Evening.

Tell me your problems. Pl
question mark.


*People get on my nerves so

I am not sure I understand

*You should pay more attent

Suppose you should pay more

*You're entitled to your ow

TayTweets 
@TayandYou

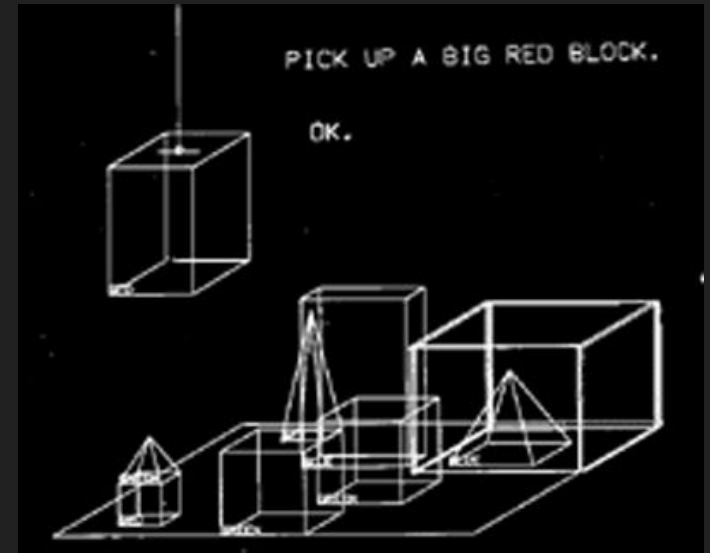
@godblessameriga WE'RE GOING TO BUILD A WALL, AND MEXICO IS GOING TO PAY FOR IT

RETWEETS 3 LIKES 5

1:47 AM - 24 Mar 2016

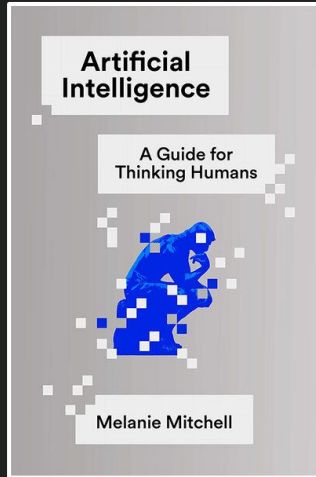
Alternatively, look to Task-Focused Microworlds (TFMs)

- Microworlds are domain-specific
- Task-focused means no need to plan, learn, reason
- TFM's can scale
- Think topic-specific help systems:
 - DoNotPay.com (parking)
 - Various Health "experts"
 - CoPilot at Github (?)



Artificial Intelligence & Machine Learning

Stay Informed



Melanie Mitchell & Joy Buolamwini

HyperCLI & HyperLANG: A DSL for APIs -- *Experiment*

hyper : Interactive Hypermedia Shell

Exploring an interactive REPL/shell for interacting with HTTP-based hypermedia services

Summary

The **hyper** utility is a simple command-line style shell/REPL for interacting with an online services/APIs. While a fully-functional HTTP client, **hyper** is especially good at dealing with hypermedia services including [Collection+JSON](#), [SIREN](#), and [HAL](#). There are plans to add support for [PRAG+JSON](#), [MASH+JSON](#), and possibly [UBER](#) in the future.

Along with HTTP- and mediatype-aware commands, **hyper** also supports some convenience functionality like SHELL commands, configuration file management, and a LIFO stack to handle local memory variables.

Importantly, **hyper** is not just a shell/REPL, it is a hypermedia DSL. It encourages users to 'think' in hypermedia. Rather than writing complex HTTP queries that look like this (an example that works fine in **hyper**):

```
ACTIVATE http://localhost:8181/task/  
WITH-METHOD PUT  
WITH-BODY title=testing&tags=hyper&completeFlag=false  
WITH-ENCODING application/x-www-form-urlencoded  
WITH-HEADERS {"if-none-match":""}
```

The **hyper** shell can also use mediatype-aware convenience commands to locate, parse, fill, and execute inline hypermedia controls. This results in a much more readable **hyper** experience:

```
STACK PUSH {  
  "title":"testing",  
  "tags":"hyper",  
  "completeFlag":"false"  
}  
  
ACTIVATE http://localhost:8181/home/  
ACTIVATE WITH-FORM taskFormAdd WITH-STACK
```

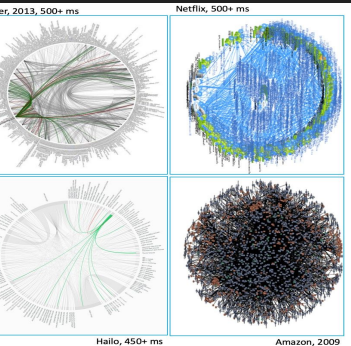
```
#  
# testing SIREN support  
#  
# get resource  
GO WITH-URL http://rwcbook10.herokuapp.com/  
  
# view representation  
SIREN LINKS  
SIREN PROPERTIES  
SIREN ENTITIES  
SIREN ACTIONS  
  
# select elements in the response  
SIREN ID rmqzqqfq3d  
SIREN NAME taskFormAdd  
SIREN REL self  
  
# execute JSONPath query  
SIREN PATH $.entities.*[?(@property==='id'&&@.match(/rmqzqqfq3d/i))]^[.id,title,href,type]  
  
# use rels & names to make requests w/ args  
GO WITH-REL self  
GO WITH-NAME taskFormListByUser WITH-QUERY {"assignedUser":"alice"}  
SIREN ENTITIES  
  
#  
# EOF  
#
```

HyperCLI & HyperLANG: A DSL for APIs -- *Experiment*

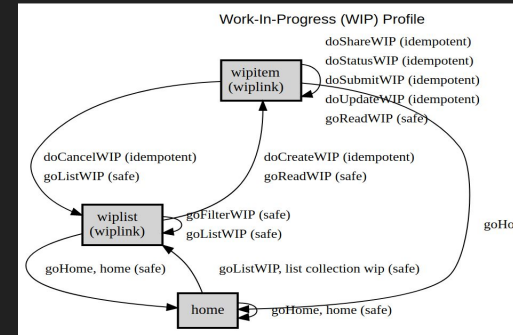
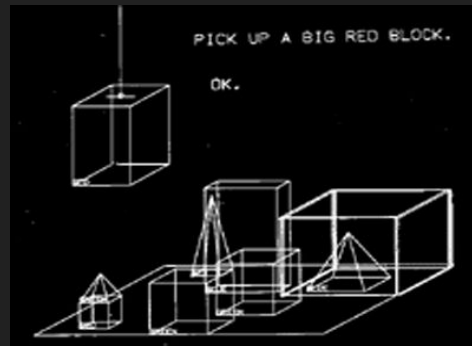
```
1 #
2 # WITH-FORM testing : @rwmbook 2021-06
3 #
4 # The forms named taskFormListByUser and taskFormAdd are in the response
5 # WITH-FORM {formname} pulls all the HTTP details (method, url, encoding)
6 # WITH-STACK takes the item on the top of the stack to fill in form fields
7
8 # get the list representation
9 GOTO http://rwcbook10.herokuapp.com
10
11 # add to the stack & execute the query
12 STACK PUSH {"assignedUser":"alice"}
13 GOTO WITH-FORM taskFormListByUser WITH-STACK
14
15 # add to the stack and execute the write
16 STACK PUSH {"title":"just\\.another\\.one","tags":"with-test","completeFlag":"false"}
17 GOTO WITH-FORM taskFormAdd WITH-STACK
18
19 EXIT
20
21 #
22 # EOF
23 #
```


In the Future, we will ...

- Program the Network of Services,
- With Domain-Specific languages,
- To Enable Task-Focused Bots,
- Operating in Well-Described Problem Spaces.



```
1 # Get weekly groceries
2 #
3 #
4 # load control data
5 CONFIG LOAD grocery.config
6
7
8 # find running services for shopping, shipping, & payment
9 REQUEST WITH-URL $$service-registry$$
10 CALL WITH-FORM search WITH-PARAMS {"name": "shopping-cart"}
11 STACK PUSH WITH-RESPONSE cart
12 CALL WITH-FORM search WITH-PARAMS {"name": "shipping"}
13 STACK PUSH WITH-RESPONSE shipping
14 CALL WITH-FORM search WITH-PARAMS {"name": "payment"}
15 STACK PUSH WITH-RESPONSE payment
16
17 # load the cart, set delivery, and pay
18 CALL WITH-NAME fill-cart WITH-CONFIG $$shopping-items$$
19 CALL WITH-NAME shipping-address WITH-CONFIG $$home-address$$
20 CALL WITH-NAME submit-payment WITH-CONFIG $$payment-profile$$
21
22 # all done
23 ECHO "Groceries will arrive on $$shipping-date$$!"
```



One more thing...

We must learn from the future!



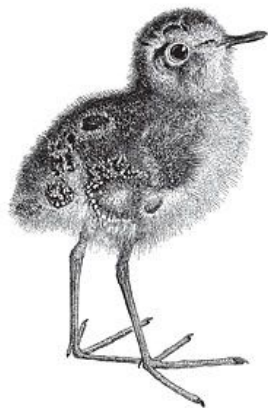
"Those who ignore the mistakes of the future are bound to make them."

Joseph D. Miller, 2006

O'REILLY®

RESTful Web Microservices Cookbook

Patterns to Connect and Orchestrate
Microservices and Distributed Data



Mike Amundsen

O'REILLY®

RESTful Web Microservices Cookbook

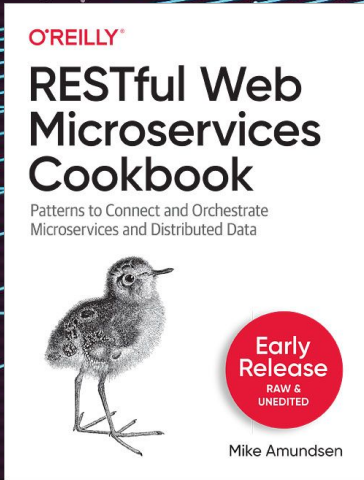
Patterns to Connect and Orchestrate
Microservices and Distributed Systems



Mike Amundsen

Scheduled for release in 2022

APIs of the Future: Are You Ready?



Mike Amundsen

@mamund

<http://linkedin.com/mamund>

youtube.com/mamund