

Steve Jobs & Computer Science

情報技術史の研究とそのための手段

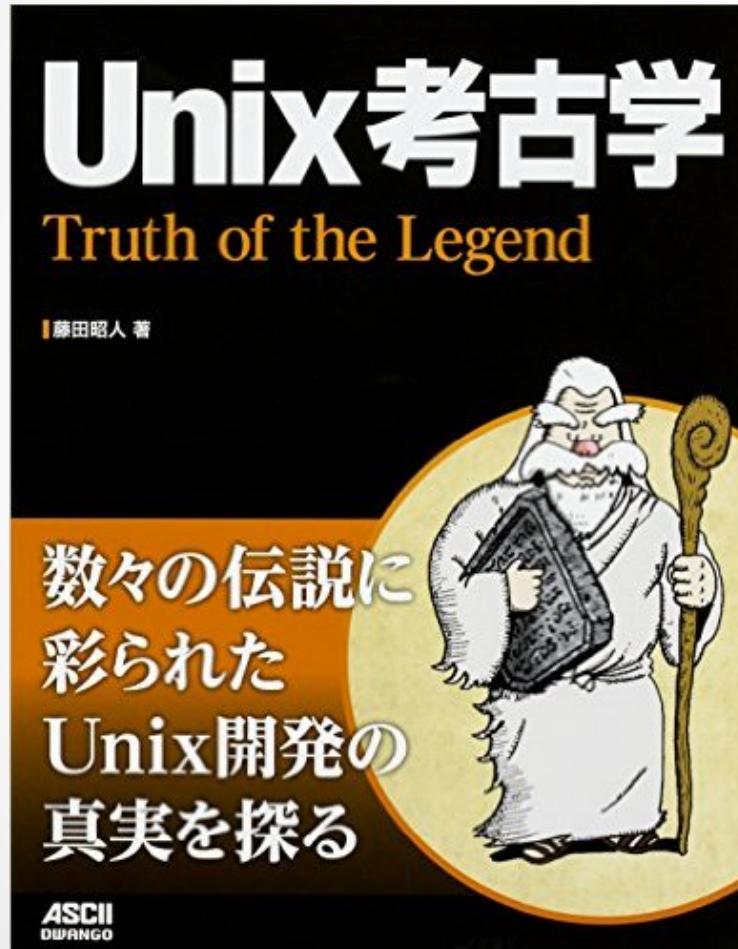
OCU Workshop 2016/12/16 Akito Fujita



自己紹介

- 氏名 藤田昭人
- 現職 I I Jイノベーションインスティテュート
- 略歴
 - 創造都市研究科第2期生（2006年3月修了）
 - その後、博士課程に進学するも退学
 - その後は石橋先生との共同研究をお願いします
- 研究テーマ
 - クラウドコンピューティングのためのプラットホームの研究
 - ソーシャルデータの分析手法の研究

本日僕が呼ばれた理由



- 4月に"Unix考古学"を出版したから
- 発売直後から講演活動
 - 1回目はさくらの横田くんの発案
 - 3ヶ月間で都合6回
- Amazonランキングでは
 - "Unixオペレーティングシステム"
 - 発売当初は1位だったことも・・・
 - 現在は68位にランクダウン
 - 講演をすると30位くらいまで浮上
- インディーズバンドみたいな気分
 - 「ライブをやって手売りだぜえ・・・」
- ということで・・・
 - 旬が終わらないうちに呼ばれた？

情報技術史の研究

- ・ 私が考える情報技術史研究
 - 情報技術を社会学あるいは歴史学の枠組みで捉える
 - 今後の情報技術研究の社会的影響を推察する上での有益な知見
- ・ 私が考える情報技術史の4つの主要テーマ
 - 情報技術の移転：1940年代の欧米間の技術移転
 - 情報技術と軍事：ARPANETの開発
 - 情報技術の民生化：1960～1980年代の商業化の試み
 - 情報技術とメディア：1990～2000年代の情報技術のメディア化
- ・ 今回取り上げるテーマは「情報技術の民生化」の事例

情報技術史研究ための手段

- 近年、情報技術史の文献リポジトリの整備が進んでいる
 - ミネソタ大学の Charles Babbage Institute
 - Oral History Collection: 情報技術で著名な研究者へのインタビュー
 - <http://www.cbi.umn.edu/oh/index.html>
 - bitsavers.org
 - 過去のコンピューターのマニュアル&ソフトウェアをアーカイブ
 - <http://bitsavers.org/>
 - Wikipedia英語版
 - 情報技術に関わる情報が非常に充実している
 - <https://en.wikipedia.org/>
- これらのリポジトリを活用すれば、情報技術史に関わる事実情報は機械的に効率良く収集することができる
 - 「Unix考古学」は文献検索だけで執筆した書籍

情報科学史文書の執筆

Step1.

構想の検討

Step2.

文献の探索

Step3.

文書の執筆

年表と
タイムライン

登場人物
と性格

トピック

情報科学史文書の執筆

- ここでは「Unix考古学」を書いた時の手順を示します
 - Step 1. 構想を練る 執筆対象に対する仮説を考える
 - Step 2. 文献を探す 仮説を裏付ける事実（文献）を探す
 - Step 3. 文書を書く 仮説と事実を組み合わせて文章に落とす
- 最も重要なのは Step 1. です
 - 執筆対象について自分の中に明確なイメージを持つ必要あり
 - Step 1. が固まっているれば Step 3. あまり悩まない
- Step 1. では次の3要素について自分の視点で把握します
 - 略歴（年表）
 - 登場（する可能性のある）人物とその性格
 - 彼らが登場するトピックやエピソード
-

Case Study

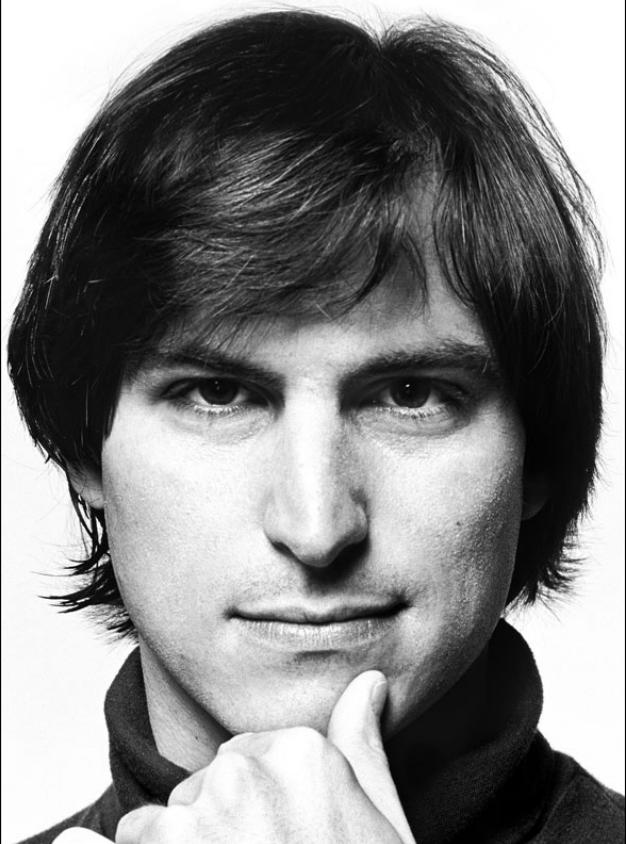
- 本稿では「新しい著作を執筆する」ことを仮定して前述の STEP 1. の作業の具体例に示します
- テーマは “Steve Jobs” です
 - 彼の視点を通して1960年代～1980年代の計算機科学の研究特に “パーソナルコンピュータ” の概念の成立について概観を試みます
- 事例としては以下の 3 つの拠点での研究を中心に紹介
 - Project GENIE (UCB)
 - Xerox Palo Alto Research Center (PARC)
 - Carnegie Mellon School of Computer Science (CMU SCS)
- 私が次の次に書きたいと思っていたテーマです 😊

Case Study

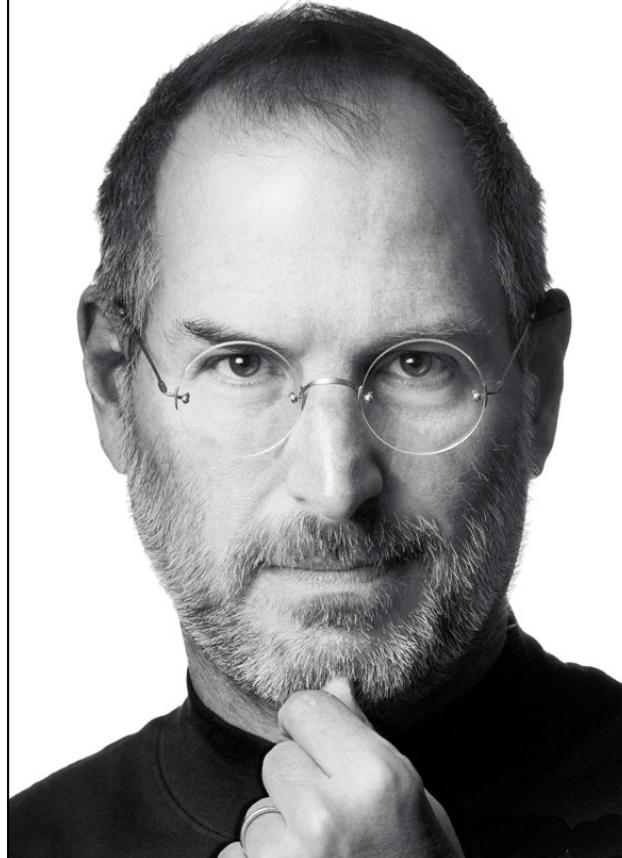
- STEP1.での作業は・・・
 - Wikipedia英語版を中心に情報を収集する
 - 最初に人物を収集する → 登場人物の定義（写真が大事）
 - 次に注目したい成果 → その人物のトピックを理解
 - 年表あるいはタイムラインを作成して前後関係を把握
 - 全体を大まかに把握するには年表
 - 特定の期間の詳細の前後関係を把握するにはタイムライン
 - 時系列に沿って人物のシーケンスを決める
 - 複数回登場する場合は成果とのペアで配置する
 - 読み物を意識して章立て（シーケンスの区切り）を設定
 - 人物や成果を時系列で眺めると自ずと区切りが見えてくる
 - 各章（区切り）毎に短文を書いてその章の概要を定義する
- この作業だけ書籍の企画書相当の内容になります

Steve Jobs

Steve Jobs by Walter Isaacson



Steve Jobs by Walter Isaacson



February 24, 1955 – October 5, 2011

Steve Jobs 1955~1980

年号	年齢	トピック
1955	0	2月24日生まれ
1971	17	高校生の時にスティーブ・ウォズニアックと出会う
1972	18	オレゴン州のリード大学へ進学するが中退
1974	19	アタリ(ビデオゲーム会社)のエンジニアとなる
1976	21	ウォズと製作したApple I を666.66 ドルで発売
1977	22	アップルコンピュータを法人化 Apple IIを販売開始
1979	24	Xerox PARC を訪問
1980	25	アップルコンピュータが株式公開
1984	29	初代Macintosh発売
1985	30	取締役会でジョン・スカリーよりすべての仕事を剥奪される

Steve Jobs 1984~1993

年号	年齢	トピック
1985	30	辞表を提出してNeXTを設立
1985	30	Carnegie Mellon University (CMU) を訪問
1986	31	ルーカスフィルムのコンピュータ関連部門を買収 (ピクサー社)
1987	32	ロス・ペローから2000万ドルの出資を引き出した
1989	34	キヤノンから1億ドルの出資を引き出した
1991	36	ディズニーへのCGアニメーション映画作成の売り込みに成功
1991	36	ローレン・パウエルと結婚 息子のリードが誕生
1993	38	NeXT全社員530人のうち280人をレイオフ (NeXTソフトウェア)

Steve Jobs 1996~2011

年号	年齢	トピック
1996	41	アップルがNeXT社を4億ドルで買収 非常勤顧問で復帰
1998	43	iMac発売 赤字経営だったアップルを黒字へ転換させる
2000	46	アップルCEOに就任
2001	47	Mac OS Xを発売
2001	47	iTunesとiPodによって音楽事業に参入
2006	51	ディズニーがピクサーを買収 個人筆頭株主になり役員に就任
2007	52	iPhone発売
2009	54	膵臓がんの治療に専念するためCEOを休職
2011	56	CEOを辞任 10月5日に死去

Steve Wozniak

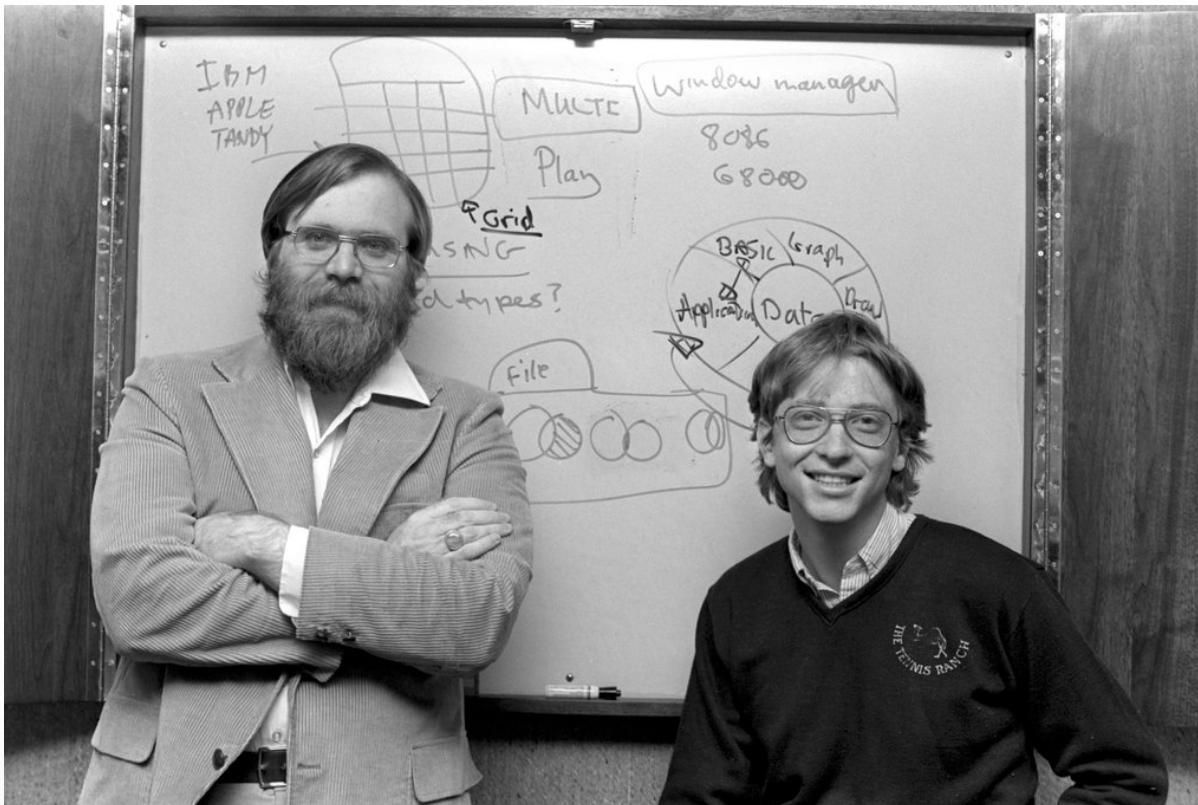


- Jobsと高校時代からの親友
- Apple I, II を独力で設計した
- 機能を落とさずチップ数を減らす天才的なエンジニア
- BASICなどソフトウェアの開発もできる
- Appleの初期の成功は彼に負うところが大きい

Two of Us



Rival



Paul Allen

Bill Gates

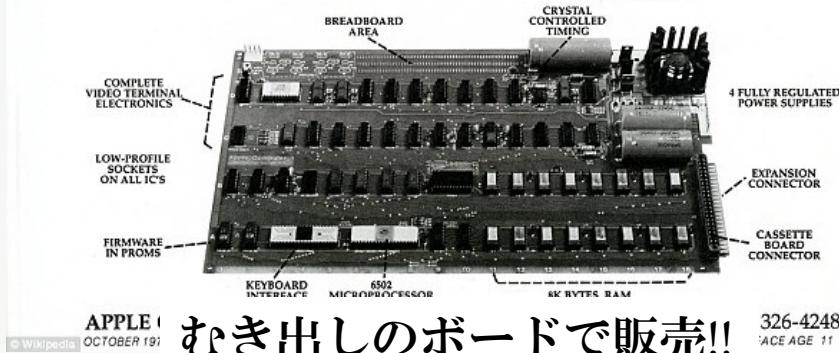
Rival

- 宿敵Microsoftを設立した二人 → 敵役は大事
 - Bill Gates ご存知 Microsoft の創業者
 - Paul Allen Microsoftの共同創業者 Altair BASICの開発者
- この2組のペアは何事につけても対照的
 - Jobs-Wozniakは理想家肌 “世界を変える!!”
 - Gates-Allenは現実主義 “ガツツリ金儲け!!”
- ビジネスアプローチも対照的
 - Jobs-Wozniakはハードウェア指向 垂直統合モデル
 - Gates-Allenはソフトウェア指向 水平分業モデル
- 味方と敵の主要登場人物を定義すると物語感が倍増
-

Apple I, II, III

Apple I

Byte into an Apple \$666.66*



Apple III



調子に乗ってビジネス
ユースを狙うも大失敗

Apple II



プラスチックケースに
収めたのはJobsの仕事

Apple I, II, III

- Apple I (1976) Apple の最初の製品
 - 元はと言えば Wozniak が趣味で作ったコンピュータ
 - おたくコミュニティで鍛えた Wozniak の技が冴える
- Apple II (1977) Apple 初期の大ヒット製品
 - スタイリッシュなケースと静かな電源を用意
 - オタクではなく一般家庭で使われるコンピュータを指向
 - 大手スーパーのおもちゃ売り場の横に陳列させた
 - おたくマシンだけでなく家庭用ゲーム機としても人気を博す
 - 爆発的なヒットにつながる
 - でも仲間内ではボードを作った Wozniak 功績だけが賞賛される
- Apple III (1981) Apple 初期の大失敗作
 - Wozniak なしでビジネスユースを狙うが・・・
 - IBM PC (1981) の登場で大きく空振り



Jobs & Computer Science

...
ジョブスと計算機科学

Apple II の成功により一躍注目を集めようになつた Jobs&Wozniak ですが、古くから彼らを知る仲間から大きな賞賛を集めたのは Wozniak でした。

格安のパーツを工夫して組み立てたボードこそが庶民にも手の届く価格の夢のマシンの核だったからです。Jobs が用意した静かな電源やお洒落なケースはそれほど重要視されませんでした。より本格的なコンピュータを実現するため Jobs は Apple III の商品化に取り込みましたが大失敗。

起死回生の策を求めて革新的な技術を持つ Xerox PARC に Jobs はコンタクトします。そこで見たものは・・・

Xerox Alto



- 暫定Dynabookとして開発された
- パーソナルコンピュータの原型
- 実は既に開発されていたMAXCのボード3枚を修正して、たった3ヶ月で最初のハードウェアを立ち上げた
- Alan Key が獲得したDynabookの研究予算をCharles P. Thackerたちがもぎ取ったとも言われている

Xerox Alto

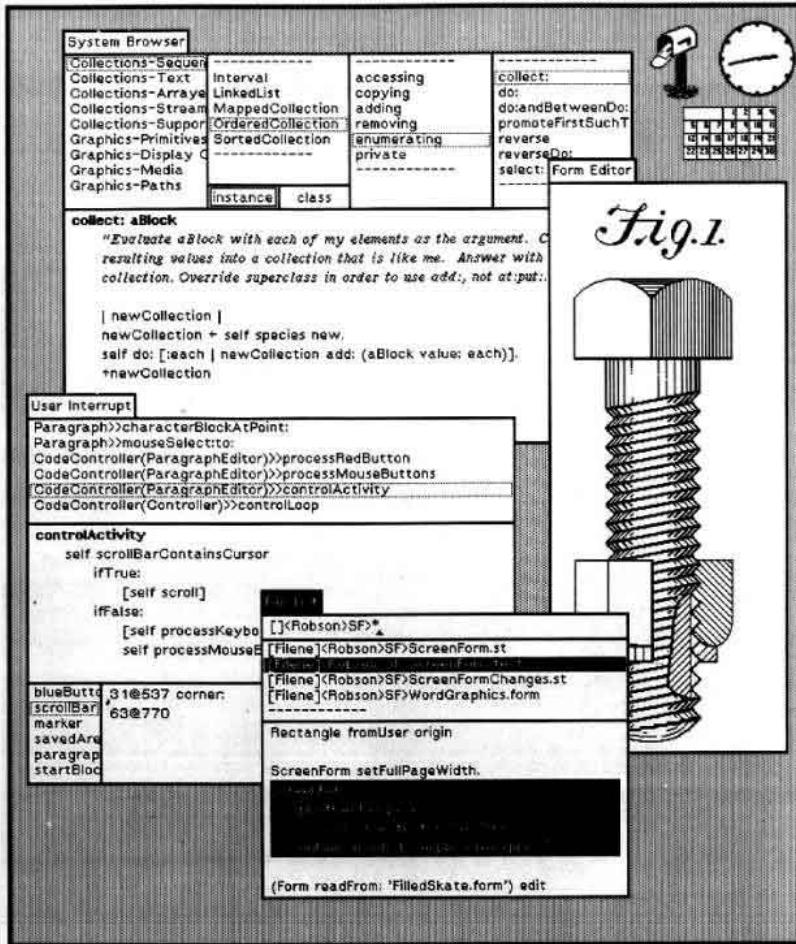


Fig.1

- BitBlitをサポートする
BitmapDisplayとMouseを装備
- システム開発言語は BCPL と Mesa
- Smalltalk とそのIDEをサポート

Altoの技術

“Dynabook” Concept

Graphical
User Interface

Bitmap Display

Bit blit

Mouse

Object-Oriented
Programming

Smalltalk

Distributed
Computing

Ethernet

Alto OS

Hardware

Altoの技術

- Alto はシングルユーザーシステムとして構想された
 - Dynabook コンセプト
- (当時としては) 革新的な技術の集合体
 - グラフィカルユーザーインターフェース
 - Bitmap Display, Bit blit, Mouse
 - オブジェクト指向型プログラミング
 - Smalltalk
 - ローカルエリアネットワークと分散コンピューティング
 - Ethernet
- 研究システムとして製造され市販されなかった
 - 限定品？ 実際に使えたのはごく少数の人たちだけ



behind the Alto

...

Alto の背景



衝撃的な実験システムであったAlto
それは1950年代から続いて対話型コンピュータを想定した
多様な研究成果の集合体でもありました。

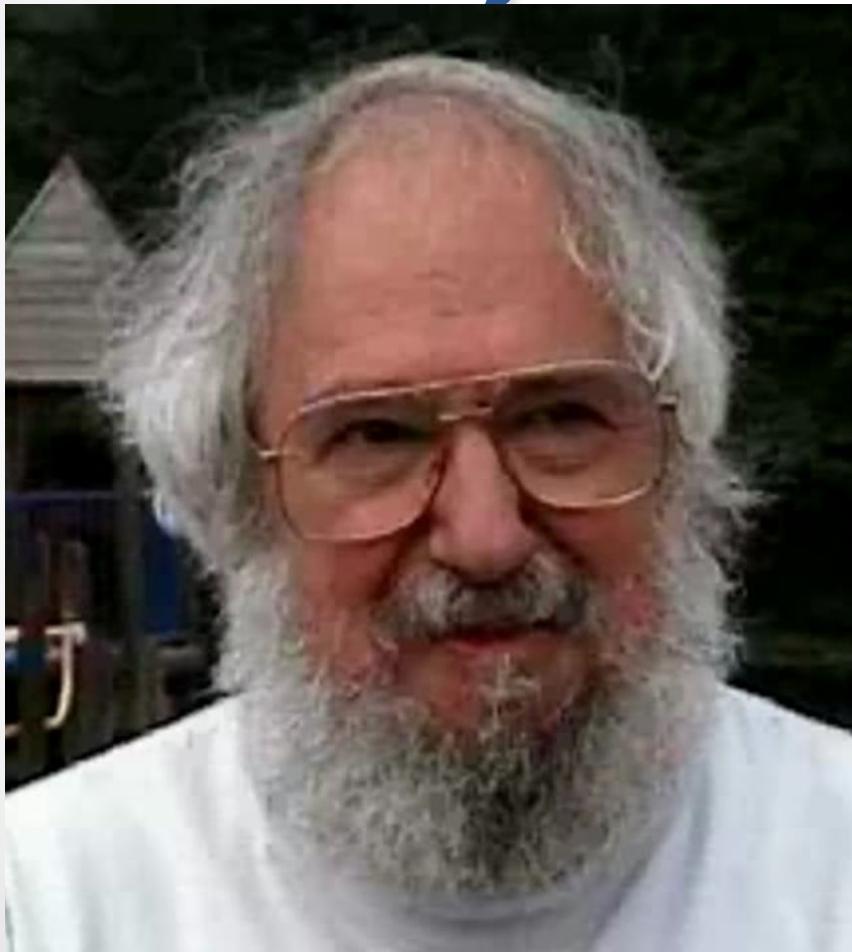


Alan Key



- Dynabook コンセプトの発案者
- Smalltalkの開発を手がける
- そのためのシステム、暫定
Dynabookを開発するためにButler
Lampsonたちに協力を求める
- 2003年 ACM Turing Award

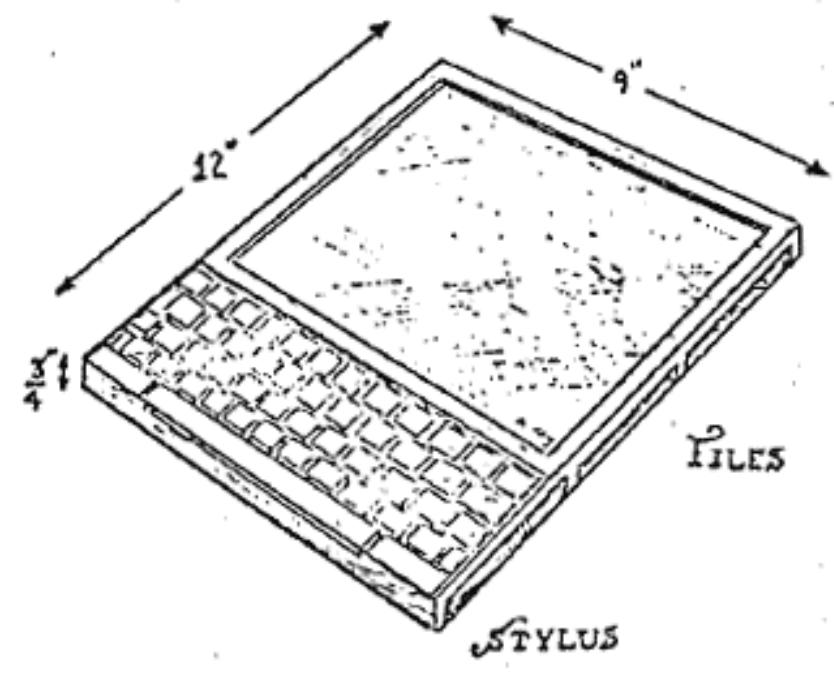
Seymour Papert



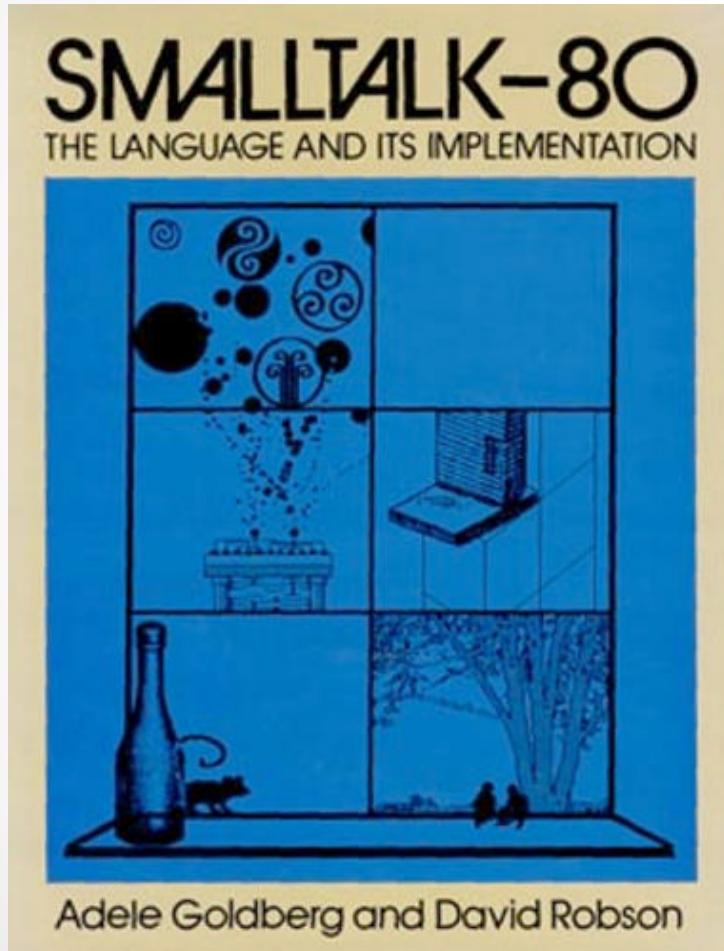
- プログラミング言語Logoの開発者
 - Lispをベースにした子供向けのプログラミング言語
- Alan Kay に大きな影響を与える
 - Dynabook
 - Smalltalk

Dynabook

- A personal computer for children of all age
- 今日の iPad によく似た想像上のコンピュータ
- 今日のパーソナルコンピュータに大きな示唆を与える

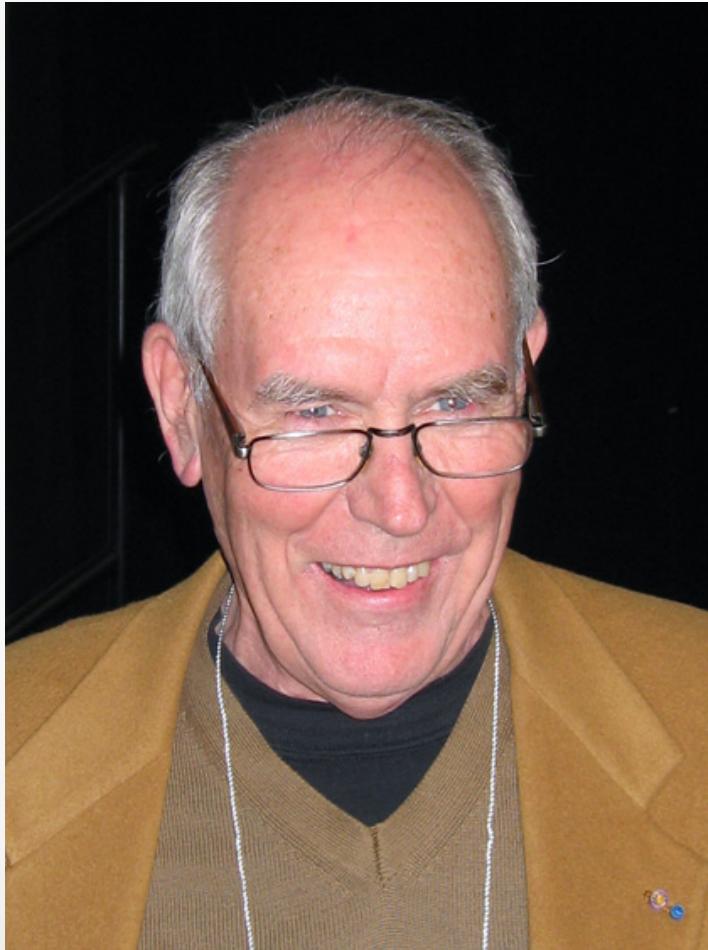


Smalltalk



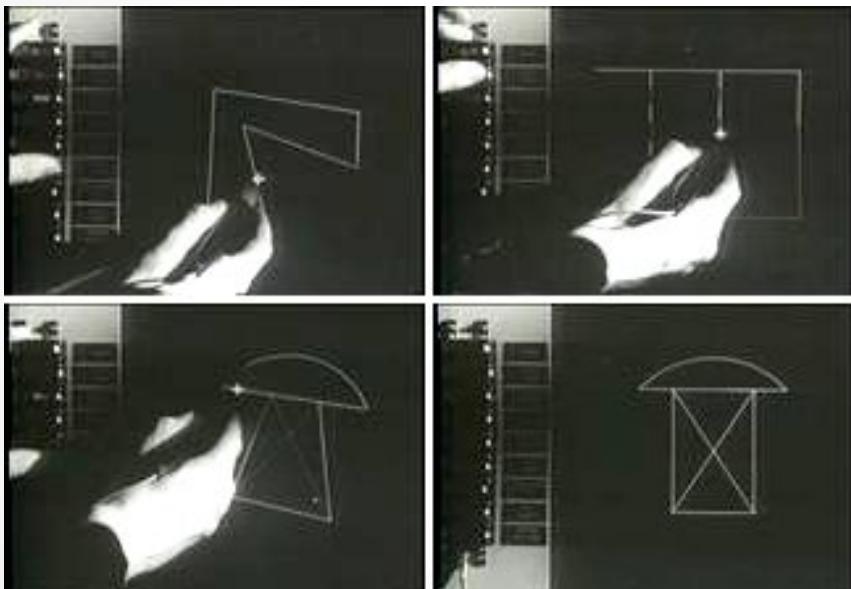
- クラスベースの純粹オブジェクト指向プログラミング言語
 - C++やObjective Cあるいは今日メジャーなスクリプト言語が備えるクラスのルーツとなる言語
- 統合化プログラミング環境の草分

Ivan Sutherland



- Sketchpadの開発者
- 対話型コンピューティングの研究領域を開拓
- 教え子はCG業界の綺羅星のような面々
 - Alan Kay Smalltalk language
 - Henri Gouraud Gouraud shading technique
 - Frank Crow antialiasing methods
 - Edwin Catmull Pixar
- ベンチャ一起業家としても実績が多い
- 1988年ACM Turing Award

Sketchpad



- 元祖グラフィカルユーザーインターフェースのシステム
- より良い対話型コンピューティングを目指す
- Ivan Sutherlandの学位研究

Douglas Engelbart

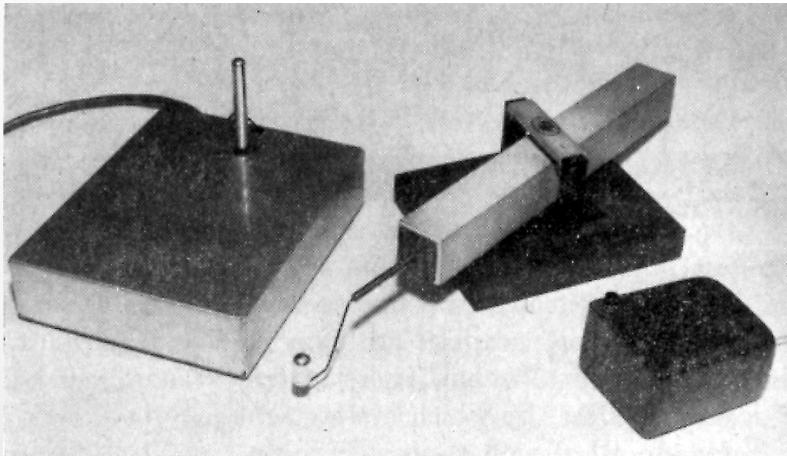


- oN-Line System (NLS) の開発者
- マウスの発明で有名
- [The Mother of All Demos](#)
- 1997 ACM Turing Award

oN-Line System (NLS)



- 対話型コンピューティングを実現するマルチユーザー連携システム
- メディアとしてのコンピュータの可能性を指し示す
- ホストシステムはSDS940



Xerox Palo Alto Research Center



- Xerox がSDS社を買収してコンピュータビジネスに参入する際、研究部門の必要性を感じて設立した研究
- ARPA IPTO の Robert Tylor を雇って先端研究を手がける研究者を集めさせた
- Project GENIEに参加した人々が集まった。Alto を作ったのは彼ら

Robert Taylor



- ARPA IPTOの3代目ディレクタ
- 第2次世界大戦での従軍経験があり、復員者の特典で心理学の修士号を取得
- 自分を抜擢してくれた J. C. R. Licklider に恩義を感じ、終生かれのビジョンの実現に力を注いだ
- 典型的ヤンキー。どんな勝負でも勝たないと気が済まない

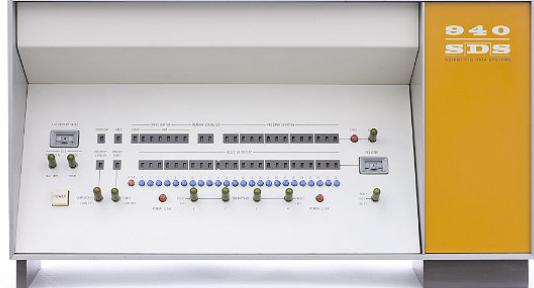
Project Genie



Project Genie researchers (left to right) Professor Paul Morton, Mel Pirtle and Wayne Lichtenberger at UC Berkeley with the SDS 930, the computer they and others on their research team modified to be the first commercially successful timesharing system. The project was one of the seminal pioneering efforts that would transform the computer from elite technology—exclusively serving U.S. industry and the military—to personal computer. With the SDS in the background, Pirtle sits in front of the machine's operator console.

- Project MAC が設立された翌年の1964年にUCBに設立された研究プロジェクト
- 研究テーマは Project MAC と同じタイムシェアリングシステム開発
- Project MAC とは異なりほぼ3年で商用化まで漕ぎ付ける
- ハードウェアはSDS940として販売される？
- ソフトウェアはプロジェクトメンバーが設立したベンチャー Berkeley Computer Corporation (BCC)が提供？

SDS940



- SDS社が販売していたSDS930を改
造して、仮想記憶をサポートした
メインフレーム・コンピュータ
- oN-Line System (NLS) のホスト
システムとしても利用された



Berkeley Timesharing System

Life after Project Genie

Following the 1968 exodus to Berkeley Computer Corporation (BCC), Project Genie alumni continued to make pioneering contributions to the science of computing. This is only a partial list of the major players and their subsequent claims to fame.



L. Peter Deutsch
(Ph.D.'73 CS) went on to Xerox PARC and Sun Microsystems, founded Aladdin Enterprises and is now a composer. He is a fellow of the Association for Computing Machinery (ACM), corecipient of the ACM Software System Award and a distinguished alumnus of Berkeley's computer science program.



Butler Lampson
(Ph.D.'67 EECS) was a founding member of Xerox PARC in 1970, where he was instrumental in developing the Alto, laser printer design, the first WYSIWYG formatting program and Ethernet. Now working for Microsoft Research and as an adjunct professor at MIT, he is a recipient of both the ACM Turing Award and the Charles Stark Draper Prize.



Wayne Lichtenberger
joined the faculty at the University of Hawaii from 1971 to 1981, then left academics for corporate positions at Hewlett-Packard, Ungermann-Bass, Inc. and Cisco Systems. He is now working full time at XKL, LLC, in Redmond, Washington, which he helped start.



Melvin Pirtle
(B.S.'61, M.S.'62, Ph.D.'67) was the impetus behind the BCC and served as its president. He then took on the ILLIAC IV project for DARPA at NASA/Ames in Mountain View. He spent two years sailing the Pacific, then began consulting with various companies. He was living and working in Las Vegas when he died of a heart attack in 2003.



Charles Thacker
(B.A.'67 Physics) moved on to Xerox PARC and worked on the Alto, Ethernet and the first laser printer, among other technologies. He was a founder of the Systems Research Center at Digital Equipment Corporation and joined Microsoft Research in 1997. He is a fellow of the ACM and recipient of the Charles Stark Draper Prize.



Ken Thompson
(B.S.'65, M.S.'66 EE) worked on the MULTICS operating system and joined Bell Labs where, in 1969, he co-created the UNIX operating system. He also co-created hardware and software for the chess computer Belle and now works at Google. He is a recipient of both the Turing Award and the National Medal of Technology.

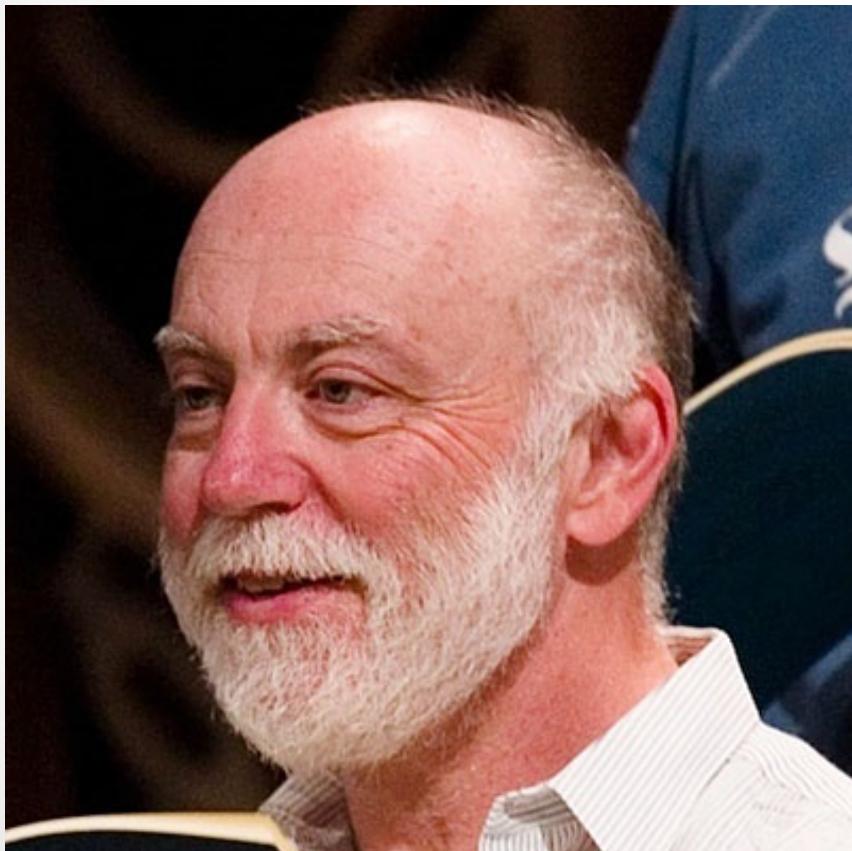
- SDS940で稼働したタイムシェアリングシステム
- 当初はBerkeley Computer Corporation (BCC)が提供したが、XeroxによるSDS買収時にXeroxの製品になった？
- Unixの開発社である Ken Thompson がUCB在学中に参加していた

Butler Lampson



- Berkeley Timesharing Systemの開発者の人一人
- Altoの開発者でもある
- 情報技術の様々な分野で研究実績があり、論文を多数発表している
- 1992年にACM Turing Award

L Peter Deutsch



- 若干17歳でPDP-1 LISPを実装した
- Berkeley Timesharing Systemの開発者の一人
- Altoの開発者でもある
- Xerox の D-Machines 向けの InterLispを実装した
- Ghostscript の開発者でもある

Charles P. Thacker



- コンピュータ・ハードウェアの設計者
- Berkeley Timesharing Systemの開発者の人一人
- Altoの開発者でもある
- 2009年にACM Turing Award

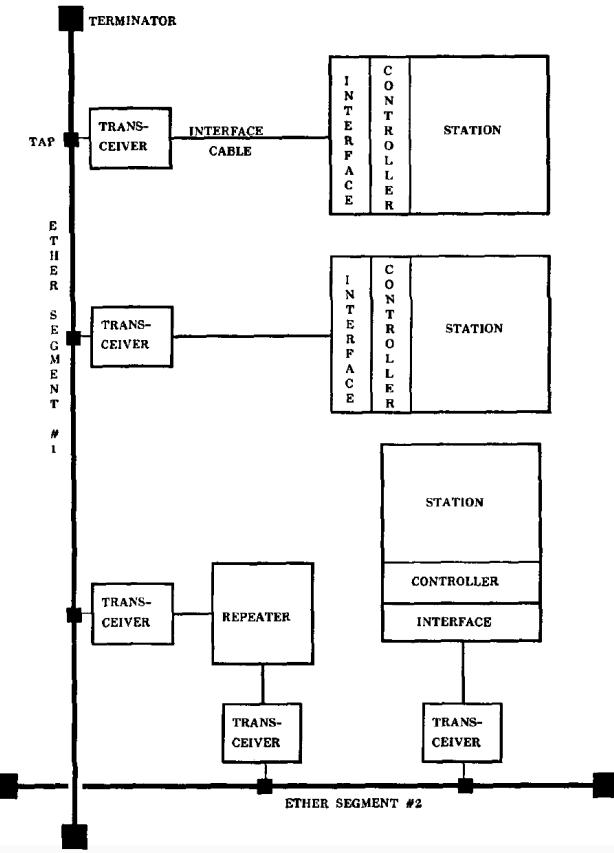
Robert Metcalfe



- Ethernet の開発者
- ネットワーク企業の3COMを設立

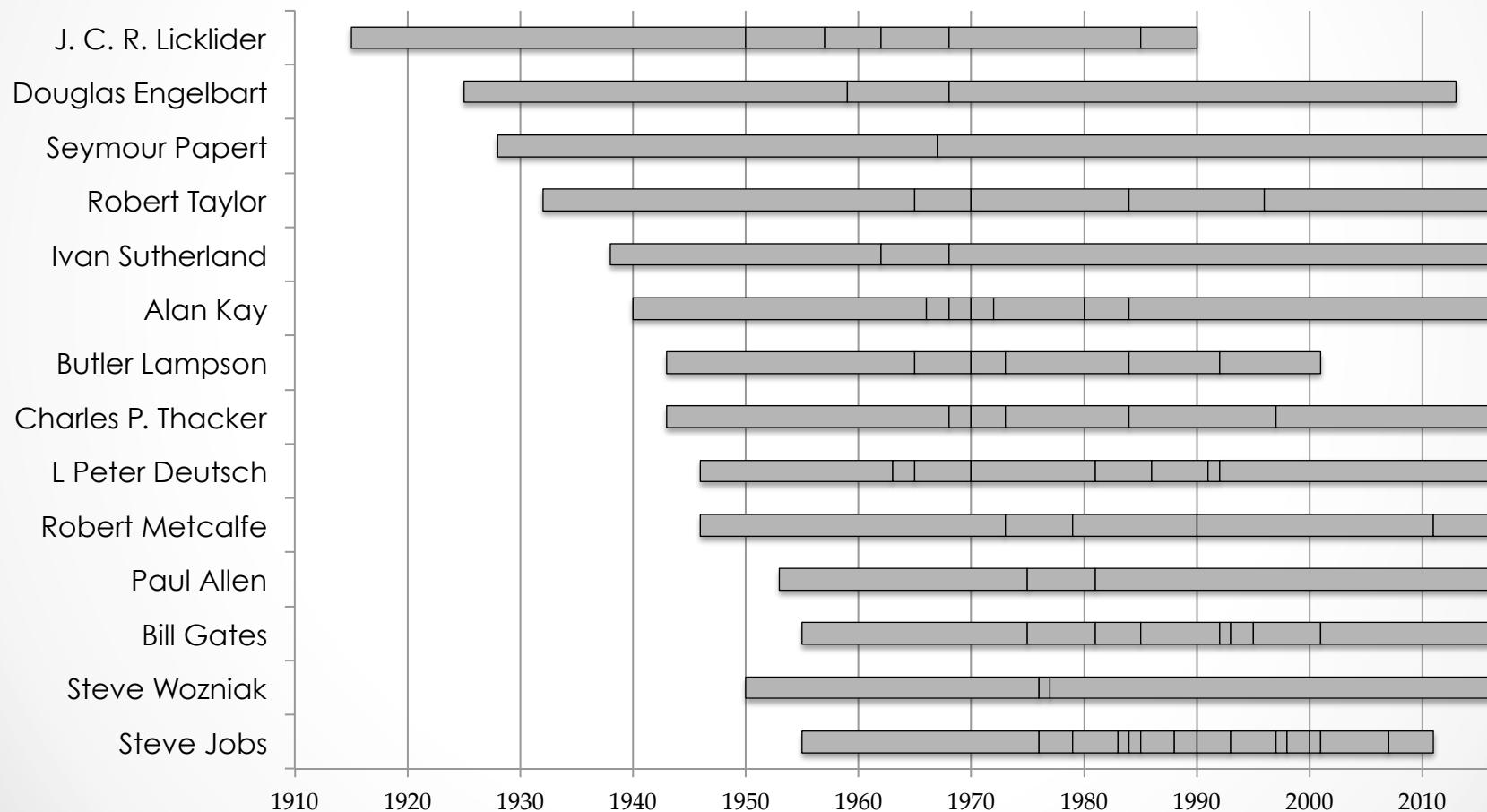
Ethernet

Fig. 1. A two-segment Ethernet.



- ローカルエリアネットワーク (LAN) の通信技術
- [CSMA/CD](#)
- ハワイ大学の無線ネットワークであるALOHA.netの通信方式を参考に開発した

Altoまでの計算機技術の系譜





Jobs vs Gates

...

ジョブス vs ゲイツ



Xerox PARCで見たAltoに魅了されたJobsは直ちにAppleが作るマイクロプロセッサによるハードウェアを使って、再現を試みます。こうして生まれたのがLisaとMacintosh。しかしAltoのソフトウェアを動かすには当時のマイクロプロセッサは非力すぎました。

さらに・・・

彼の野心的な試みをすぐ後ろをピッタリ付いてくる存在もありました。



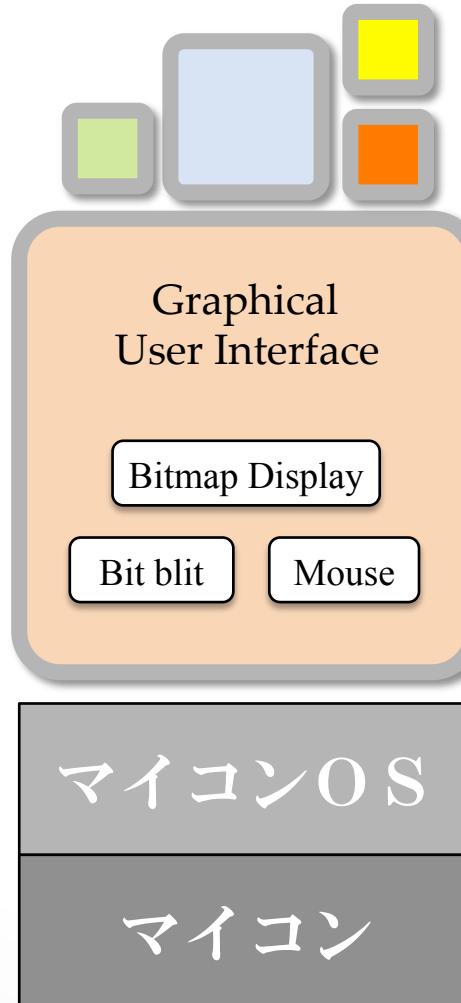
Lisa & Macintosh



Lisa & Macintosh, Windows

- Lisa (1983)
 - 1983年に発売されたが\$9,995という高額な価格が商業的には失敗
 - 1985年には生産停止。
- Macintosh (1984)
 - 1979年にJef Raskinが主導して開発プロジェクトが開始
 - 1981年にJobsの介入によりRaskinはプロジェクトから離脱
 - 1985年の追放によりJobsもプロジェクトから離脱
 - その後DTP市場等での成功によりデザイン分野などでは定着
- Microsoft Windows (1985)
 - 1983年のLisa出荷後に開発をアナウンスをした
 - 続くWindows 2.0 (1987) Windows 2.1x (1988) もポンコツ
 - Windows 3.0 (1990) になってようやく何とか使えるように

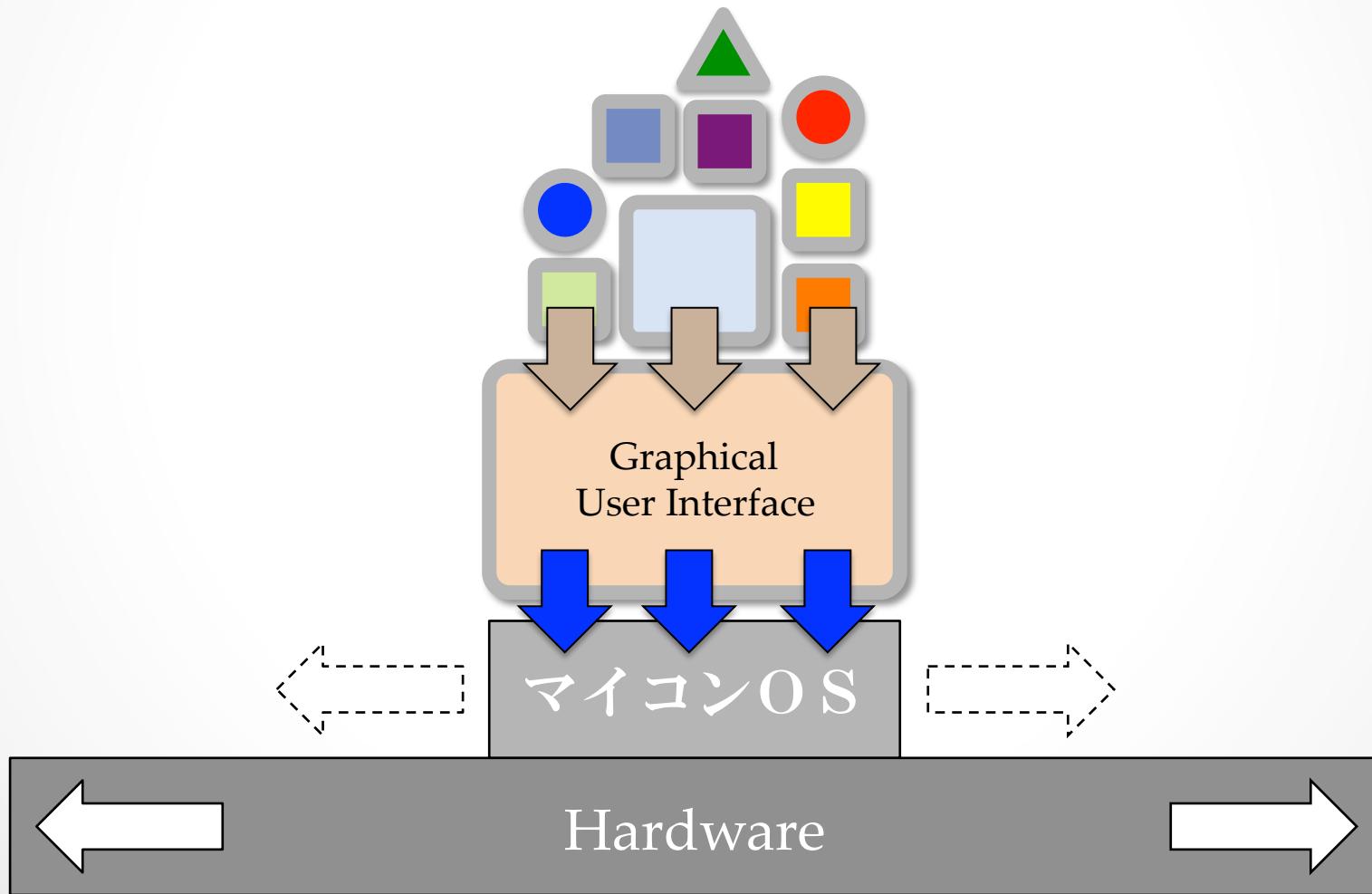
Xeroxの技術を模倣？



Xeroxの技術を模倣？

- 安直に従来のマイコンOSの上にGUIのみを上積み
 - 当時のハードウェアの能力が低すぎた
 - GUIにばかり目を奪われすぎた
- 廉価版だから劣化コピーで許してもらえる?
 - “Dynabook”コンセプトはそっちのけ
 - オブジェクト指向はサードパーティ任せ
 - 分散コンピューティングは忘れたふり
- Microsoftは当初Macを忠実に模倣してWindowsを開発
 - 双方ともサードパーティが撃破りを繰り返して
 - システムとしての破綻に近づいていく

Mac and Windows



Mac and Windows

- Windows 3.1x (1992) 登場により商業的形勢は逆転
 - Mac だけでなくその他のワークステーションも淘汰の対象
- しかし OS Kernel の脆弱性が徐々に顕在化
 - 複数のサードパーティのソフトウェアを使うとトラブル続出
 - PCのハードウェア能力は Alto を凌駕していたが
 - アプリケーションの重みのため OS Kernel は追従できず
- 資金力のあるMicrosoftは（迷走するも）次々に手を打つ
 - Windows NT (1993) Windows 9x (1995)
- 打開策が見出せず迷走したJobsなきAppleは倒産の危機
 - Classic Mac OS
- 結局、ジョブズに千載一遇のチャンスが・・・
-



“Workstation”, Alto’s Successors

…

ワークステーション、
Alto の継承者

Jobs は Lisa を発表した1983年にはAppleのハードウェアが Altoを GUI を完璧に再現するには非力すぎることに気づいていました。そこで密かに Big Mac なるプロジェクトを立ち上げて、よりパワーのあるコンピュータの開発に着手しました。これは、今日ワークステーションと呼ばれているカテゴリのコンピュータだったようです。

1985年のJobsの追放によりこのプロジェクトはAppleでは頓挫しますがNeXTという新たな会社において開発は継続されます。Appleの軛から解放されたJobsが向かったのは Carnegie Mellon University (CMU) でした。その理由は彼が居たからです。

Carnegie Mellon School of Computer Science



Richard Rashid



Mach

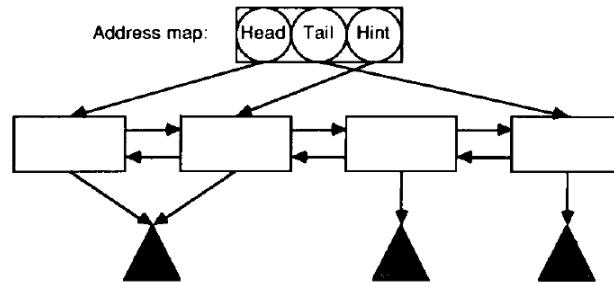


Fig. 1. A simple address map.

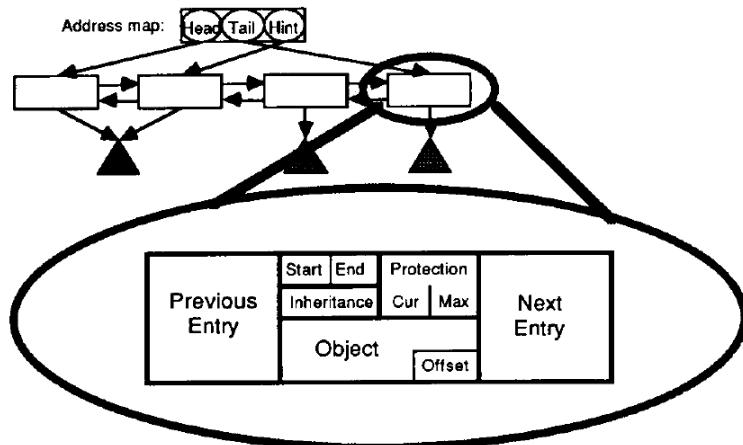


Fig. 2. An address map entry.

Accent

Matchmaker:
The Accent Remote Call Procedure Language



August 1984

Copyright © 1983

PERQ Systems Corporation
2600 Liberty Avenue
P.O. Box 2600
Pittsburgh, PA 15230
(412) 355-0900

Three Rivers PERQ



RIG & Aleph kernel

- Rochester Intelligent Gateway (RIG)

RIG, Rochester's Intelligent Gateway:

System Overview

by

E. Ball, J. Feldman, J. Low, R. Rashid, and P. Rovner

Computer Science Department

University of Rochester

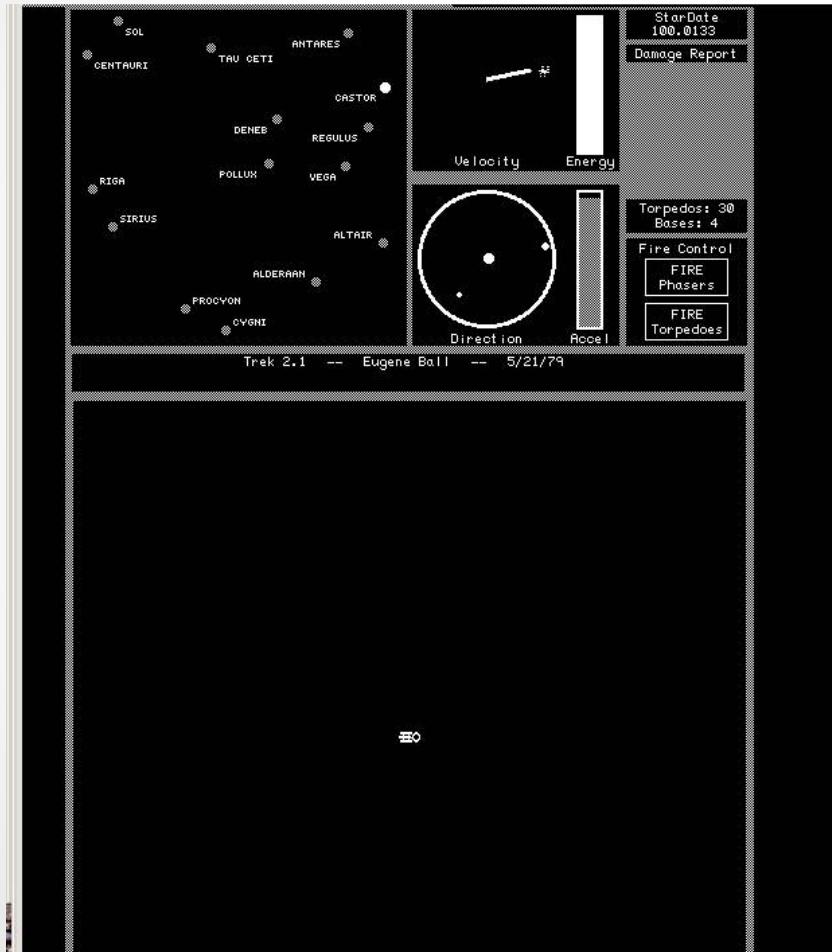
TR5

The RIG system provides convenient access to a wide range of computing facilities. The system includes five large mini-computers in a very fast internal network, disk and tape storage, a printer/plotter and a number of display terminals. These are connected to larger campus machines (IBM 360/65 and DEC KL10) and to the ARPANET. The operating system and other software support for such a system present some interesting design problems. This paper contains a high level technical discussion of the software designs, many of which will be treated in more detail in subsequent reports.

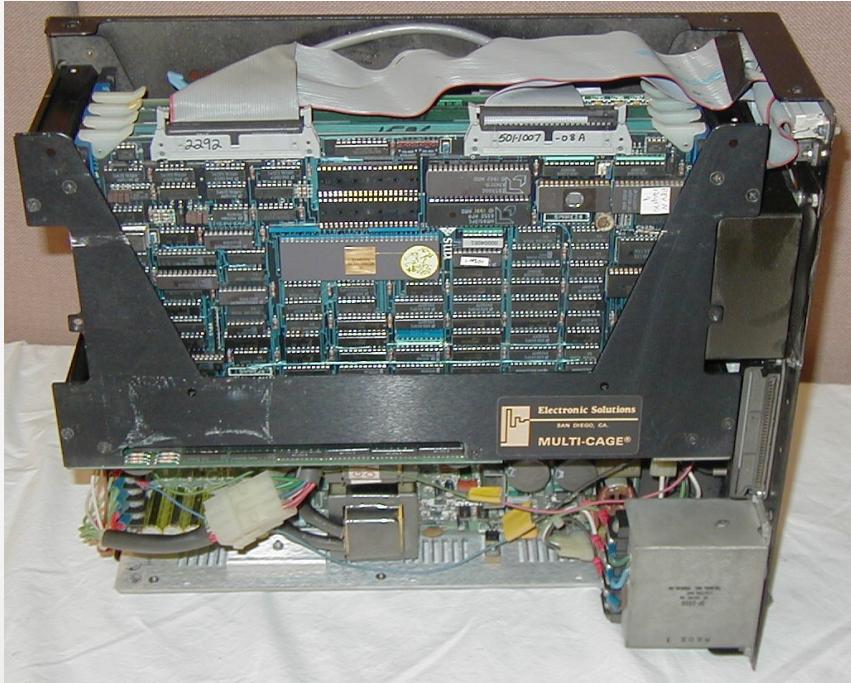
50 Altos Donation



Alto Trek



SUN workstation





NeXT Inc.

• • •

Carnegie Mellon University の技術を導入することにより
"Big Mac" は徐々に形になっていきます。そこでは CMU
からリクルートしてきた彼の貢献が大きかった。

のちに彼は Jobs の Apple 凱旋の技術的裏付けを担うので
すが、彼が NeXT に参加した 1987 年の段階では、それは誰
にもわからないことでした。



Avie Tevanian



NeXT



NeXT Cube



NEXTSTEP



NEXTSTEPの技術

“bicycle for our minds” Concept

Graphical
User Interface

Bitmap Display

Bit blit

Mouse

Object-Oriented
Programming

Objective C

Interface Builder

Distributed
Computing

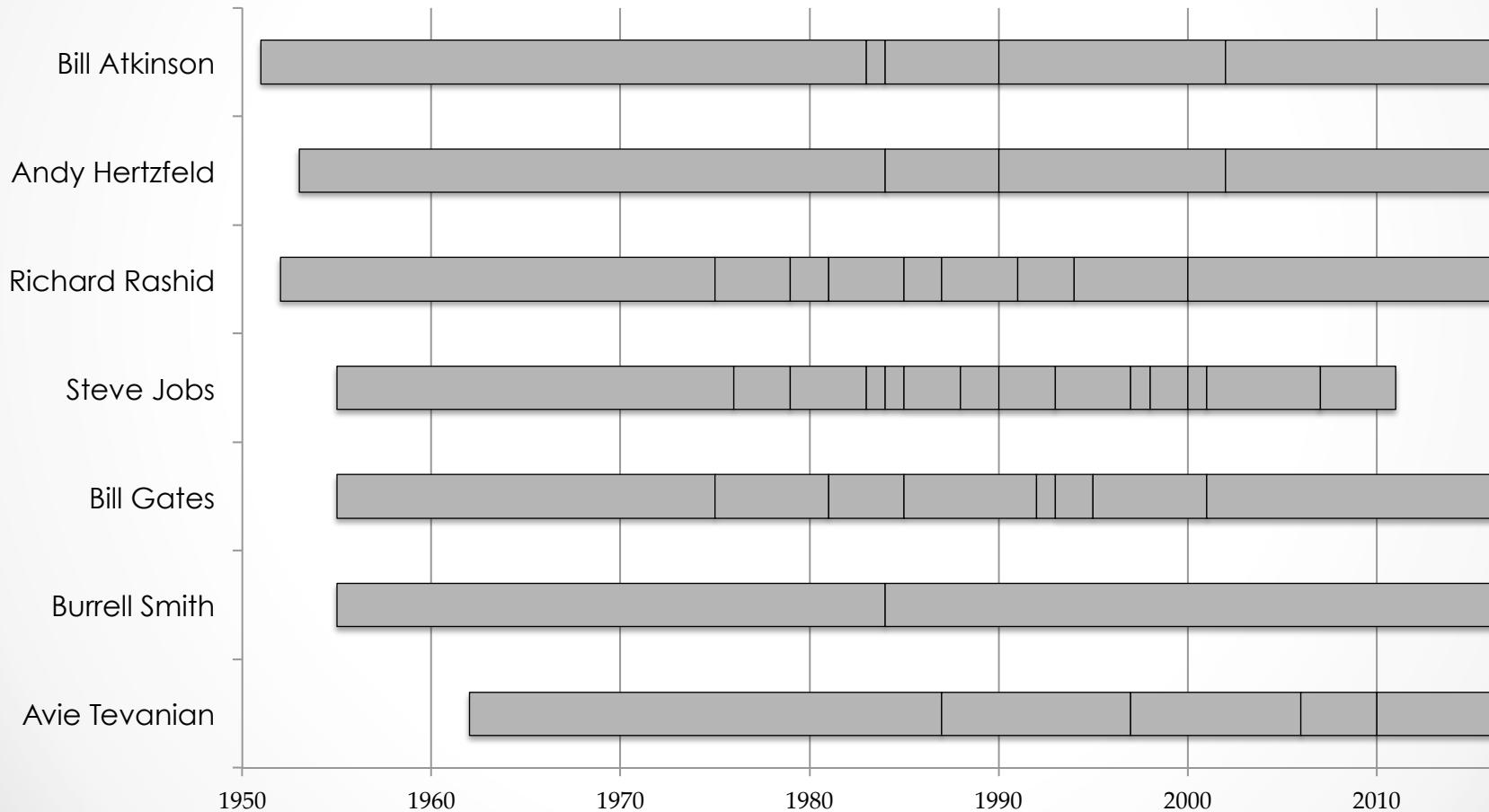
Ethernet

Portable Distributed Objects

Mach Kernel

Hardware

Altoからの計算機技術の系譜





Apple “Interim” CEO

...

Appleへの凱旋



理想としたハードウェアと一体の垂直統合モデルによるビジネスを断念したものの、金融業界などの固定的ファンを獲得したNEXTSTEPにより経営が安定した NeXT Inc.。さらに Pixar の成功により事業家としての地位を掴み取った Jobs。

しかし1996年、思いがけない話が飛び込んできます。Apple が自社でのOS開発を断念し社外からの調達に動いている・・・業績低迷が続き既に瀕死の状態に陥っている Apple。結局、Jobs は敢えて火中の栗を拾う選択をします。

今日、みなさんが良く知るJobsの復活劇の始まりでした。

Gil Amelio



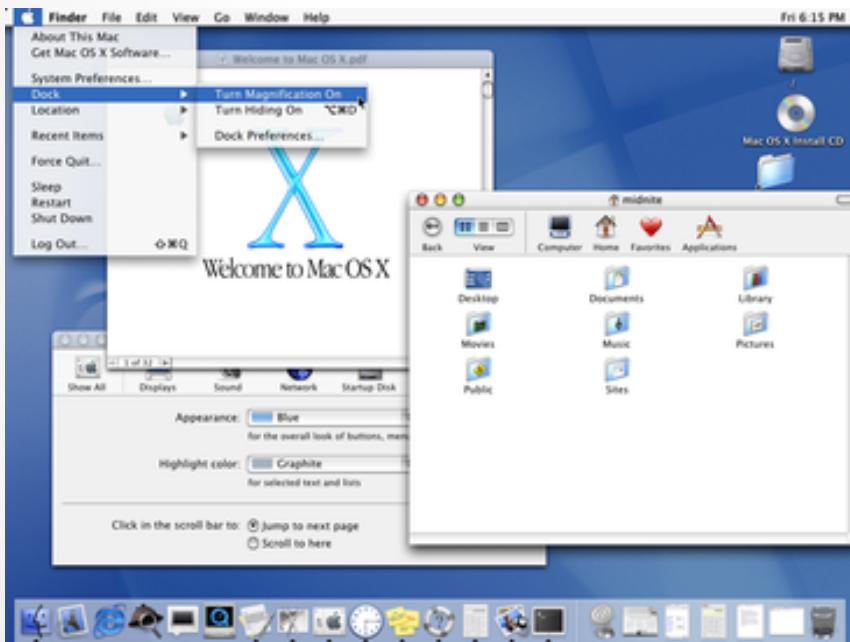
Jean-Louis Gassée



iMac G3



Mac OS X 10.0





Jobs' Role at CS

...

計算機科学におけるジョブスの役割



Macintoshの開発においてJobsが取り入れた技術は数多くの計算機科学の研究成果でした。そこでは多数の研究者が関わっていたことがわかつてもらえたと思います。

では、これらの研究者が彼ら自身の研究を始めた理由とはなんだったのでしょうか？実は彼らの研究を理解して資金を提供しつづけた存在がありました。



ARPA IPTO



J. C. R. Licklider



Man-Computer Symbiosis (1960)

Man-computer symbiosis is a subclass of man-machine systems. There are many man-machine systems. At present, however, there are no man-computer symbioses. The purposes of this paper are to present the concept and, hopefully, to foster the development of man-computer symbiosis by analyzing some problems of interaction between men and computing machines, calling attention to applicable principles of man-machine engineering, and pointing out a few questions to which research answers are needed. The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today.

「人間とコンピュータの共生」は、マン・マシン・システムの1つの研究分野である。マン・マシン・システムは数多くあるが、人間とコンピュータの共生は今のところ存在しないといつてよい。本論のねらいは、人間とコンピュータの共生の概念を示し、願わくばその発展を促すことだ。そのためには人間とコンピュータの相互作用における問題の幾つかを分析し、マン・マシン・エンジニアリングの原理の適応可能な領域を考察し、研究によって解答が必要とされる新たな問題を提起しよう。近い将来、人間の頭脳とコンピュータが緊密に結合して、その結果得られた共生関係によって、これまでいかなる人間の頭脳でもなされなかつた思考を行い、現在知られているどの情報処理装置にも試みられていない方法でデータを処理するとといったことが期待されているのだ。

bicycle for our minds (1990)

I think one of the things that really separates us from the high primates is that we're tool builders. I read a study that measured the efficiency of locomotion for various species on the planet. The condor used the least energy to move a kilometer. And, humans came in with a rather unimpressive showing, about a third of the way down the list. It was not too proud a showing for the crown of creation. So, that didn't look so good. But, then somebody at Scientific American had the insight to test the efficiency of locomotion for a man on a bicycle. And, a man on a bicycle, a human on a bicycle, blew the condor away, completely off the top of the charts.

And that's what a computer is to me. What a computer is to me is it's the most remarkable tool that we've ever come up with, and it's the equivalent of a **bicycle for our minds**.

私たちを霊長類から本当に切り離す1つの例は、私たちが道具の作り手だということです。私は以前、地球上の様々な種の移動効率を測定した研究を読んだことがあります。1キロメートル移動するために使用するエネルギーは、コンドルが最低で、人間はリストの下の3分の1のあたり、あまり印象的ではなかったように見えて、それほど誇らしいことではありませんでした。しかし、サイエンティフィック・アメリカンの誰かは、自転車に乗っている人の歩行効率をテストする洞察力を持っていました。自転車に乗った人間だとコンドルを吹き飛ばし、チャートのトップを完全に突き破ります。

コンピュータは私にはそういう存在なのです。私にとってコンピュータとは、それがこれまでに思いついたなかでも最も注目すべきなツールであり、**私たちの思考のための自転車**に相当するものです。



<https://www.brainpickings.org/2011/12/21/steve-jobs-bicycle-for-the-mind-1990/>



Licklider & Jobs

- “Man-Computer Symbiosis”は壮大な問いかけであった
 - 1960年には [Jules Verne](#) の物語のような話だった
 - 計算機科学者への Licklider のとんでもない挑発でもあった
- “bicycle for our minds”は最もエレガントな回答だった
 - Licklider の問いに答えようとした人は無数にいたが・・・
 - Jobs ほどオシャレかつ具体的に回答を提示した者はいない
 - その証拠は・・・ Licklider が夢想し Jobs が具現化したデバイスを使って、今、僕はみなさんにプレゼンテーションをしている
- 僕が考えること
 - 二人とも「計算機科学者ではない」ことは偶然ではない
 - 1960年～1990年はコンピュータが最もイノベティブな時代

で、Jobsの物語の結論は？

僕は構想段階では結論を用意しません。

STEP 2. の文献探索で新事実を発見する可能性があるから、
STEP 3. の文章執筆が全て終えてから結論を考えます。

でも・・・この Case Study をみると結論は決まっている
ようにも見えますね😊





Method & Tools

…

情報技術史の研究のための手段

STEP 2. 文献探索

- 実はこの工程が最も長く苦痛な作業だったりします
 - ジャーナリストでも作家でもない僕には検索エンジンだのみ
 - 「Unix考古学」の執筆でも期間の2/3はこの作業
 - 正味10日間で「記事に使える論文を10本」は苦行です
 - ハズレ文献が多すぎるし、概要把握にも時間が掛かる→徒労感
- 現在、この工程を機械化する方策を考えています
 - 僕が“Drilldown Search”と呼んでいる技術です
 - 僕が身に付けてる属人的ノウハウを機械に教える試みです
 - 現在の統計的自然処理技術を使うとかなりイケそうな・・・
- この方法についてエッセイ風文章を書いているので、
• 興味のある方はお持ちください（限定20部）

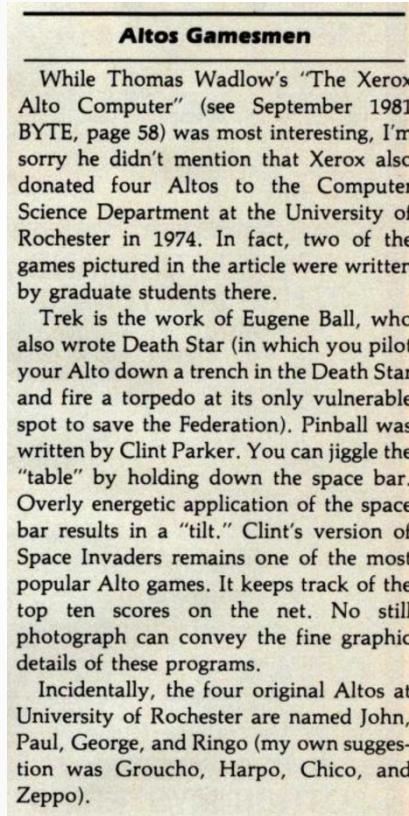
50 Altos Donation

The Xerox Alto Computer
September 1981 BYTE publications Inc.

In 1978, Xerox donated a total of fifty Altos to Stanford, Carnegie-Mellon, and MIT (Massachusetts Institute of Technology). These machines were quickly assimilated into the research community and rapidly became the standard against which other personal computers were judged.

- Wikipedia の “Xerox Alto” のページに Alto を幾つかの大学に寄贈したとの記述があったので、その裏取りを試みるも・・・
- Stanford, CMU, MIT への寄贈に言及した記事は見つけた
- でも Richard Rashid が在籍した University of Rochester への寄贈には言及されてない

50 Altos Donation



Altos Gamesmen
February 1982 BYTE publications Inc.

- さらに検索しまくったところ、Alto のゲームに関する記事を見つけた
- この記事で「数人の院生が作った」とされる Alto Trek について Wikipedia で調べてみると・・・
- 開発者に "Rick Rashid" の名前を見つけた
- もう一人の開発者である "Gene Ball" はゲーム作家として著名
- 初のネットワーク対戦ゲームである Alto Trek を開発するにあたり、分散システムの専門家である Rick に協力を求めたのではないか？
- これで Richard Rashid が Alto のソフトウェアに習熟していたことの裏取りは完了

言語リソースとしてのWikipedia

- ・ インターネットリソースを活用した文章執筆
 - Search Engineだけで書籍の執筆は可能
 - 文献探索の機械化が執筆作業の効率化につながる
- ・ Drilldown Searchによる文献探索
 - 検索結果として見つかった文献から更なる検索条件を抽出
 - 時制情報や固有名詞に着目すると検索精度が上がる
- ・ 固有表現認識（NER）
 - 今日の統計的自然言語処理では時制・固有名詞は抽出可能
 - Python NLTK を使うと比較的簡単に検証できる

言語リソースとしてのWikipedia

- NEコーパスの作成はコスト大
 - 文献に固有表現タグを付与する作業は人手
 - フリーのNEコーパスは滅多にない
 - NLTKのNERに付属するのはチャンカーのみ
- WikipediaからのNEコーパス生成
 - Wikipediaは記事毎に固有表現タグが設定可能
 - タグ付き記事へのリンクを含む記事を文例として使用可能
 - WikipediaリソースだけでNEコーパスが生成できる
 - しかしコーパス生成の完全自動化には課題も多い

STEP 3. 文書執筆

- ・もちろん属人性の極致なので機械化は難しいが・・・
- ・今ならなんらかの機械化方策がみつかるのではないか?
 - STEP 2. の文献検索が Drilldown Search で機械化できると仮定
 - 少なくとも半自動的機能は作れるのでは?
- ・文書自動生成技術は最近需要が高まっているらしい
 - 例えば DeNA のキュレーション・サイトの閉鎖とか・・・
 - 向こう半年内に何らかの技術的解が提示できれば・・・
- ・すいません、まだアイデアはまとまってません
 - 石橋先生にはポロポロ漏らしてるけど、多分全く伝わってない
 - 言ってる本人がまとめた形で説明できないぐらいだから
- ・

まとめ

- 情報科学史文書の執筆は3つの工程に分けられる
 - Step 1. 構想を練る 執筆対象に対する仮説を考える
 - Step 2. 文献を探す 仮説を裏付ける事実（文献）を探す
 - Step 3. 文書を書く 仮説と事実を組み合わせて文章に落とす
- Step 2. 3. に関しては最近の自然言語処理技術を活用して機械化が可能（な気がする）
 - 固有表現認識（NER）がキーテクノロジー
 - そのためのコーパス生成が難題
- これからクラスターをぶん回して頑張ってみます
 - 答えがでたらまた報告します
-



Epilogue

...

Jobs' Speech at Stanford University

“Stay hungry, Stay foolish”

Your time is limited,
so don't waste it living someone else's life.
Don't be trapped by dogma —
which is living with the results of other people's thinking.
Don't let the noise of others' opinions
drown out your own inner voice.
And most important,
have the courage to follow your heart and intuition.
They somehow already know
what you truly want to become.
Everything else is secondary.



“Stay hungry, Stay foolish”

君たちの時間は限られている。
だから他の誰かの人生を生きて
時間を無駄にしてはいけない。
ドグマに囚われてはならない —
他の人々の思考に従属して生きることになるからだ。
他人の意見の騒音で
自分の内なる声をかき消してはならない。
そして最も大事なことは、
自分の心や直感に従う勇気を持つことだ。
元より君の心や直感は
君が本当になりたいものを知っている。
それ以外の事は二の次だ。