The Power Of The Context

Alan Kay

[Remarks upon being awarded — with Bob Taylor, Butler Lampson and Chuck Thacker — the Charles Stark Draper Prize of the National Academy of Engineering, February 24, 2004]

VPRI Memo M-2004-001

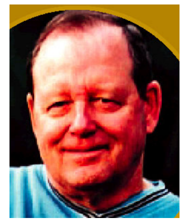
Viewpoints Research Institute, 1209 Grand Central Avenue, Glendale, CA 91201 t: (818) 332-3001 f: (818) 244-9761

The Power Of The Context

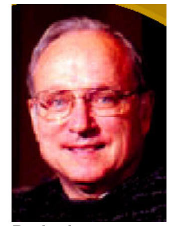
Remarks upon being awarded — with Bob Taylor, Butler Lampson and Chuck Thacker — the Charles Stark Draper Prize of the National Academy of Engineering, February 24, 2004

by Alan Kay

When Bill Wulf called to say that the four of us had been awarded this year's Draper Prize, I was floored because even the possibility was not in my mind. Given the amazing feats of engineering in the 20th century, the previous laureates, and that this is just the 10th awarding of the prize, it seems unbelievable to have been chosen. Of course, every engineer, mathematician and scientist — every artist — knows that the greatest privilege is being able to do the work, and the greatest joy is to actually turn yearnings into reality. So we were already abundantly rewarded many years ago when this work came together to create a new genre of practical personal computing.

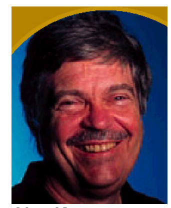


Bob Taylor



Butler Lampson

Chuck Thacker



Alan Kay

There were three people who were absolutely indispensible to Xerox PARC's success: Bob Taylor, Butler Lampson, and Chuck Thacker. Receiving this award with them is a truly incredible honor. Since this award is about a whole genre of computing, it is extremely important to acknowledge and thank the larger group of several dozen PARC researchers who helped conceive the dreams, build them and make them work. This was especially so in our Learning Research Group, where a wide range of special talents collaborated to design and build our computing and educational systems. I particularly want to thank Dan Ingalls and Adele Goldberg, my closest colleagues at PARC for helping realize our dreams.

Dan Ingalls

Adele Goldberg

About 10 years ago I wrote a history paper about our group's research (available online: see references below) and found, even in 60 pages, I could not come close to mentioning all the relevant influences. This is because I've long been an enthusiastic appreciator of great ideas in many genres—ranging from the graphic, musical and theatrical arts to math, science and engineering. I've been driven by beauty, romance and idealism, and owe more intellectual debts than most, starting with my artistic and musical mother and scientist father.

"When there was only one personal computer... "
The UR-Vision: Ivan Sutherland and Sketchpad on TX-2
3am-6am at Lincoln Labs in 1962

My best results have come from odd takes on ideas around me—more like rotations of point of view than incremental progress. For example, many of the strongest ingredients of my object-oriented ideas came from Ivan Sutherland's Sketchpad, Nygaard & Dahl's Simula, Bob Barton's B5000, the ARPAnet goal, Algebra and Biology. One of the deepest insights came from McCarthy's LISP. But the rotational result was a new and different species of programming and systems design that turned out to be critically useful at PARC and beyond.

Bob Barton

John McCarthy

James Watson

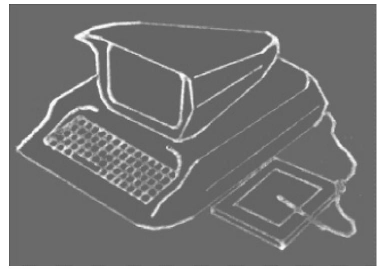
Similarly, my start in personal computing came first from my colleague and friend, the wonderful and generous Ed Cheadle, who got me deeply involved in building "a little desk-top machine"—the FLEX Machine—that we called a "personal computer". Many of the later ideas incorporated were “adaptations, rotations, and dual reflections" of the lively ARPA interactive computing community with its cosmic visions of Licklider, Taylor, Engelbart, Clark, Shaw, Ellis, and many others about “man-computer symbiosis and intergalactic networks”.

My interest in children's education came from a talk by Marvin Minsky, then a visit to Seymour Papert's early classroom experiments with LOGO. Adding in McLuhan led to an analogy to the history of printed books, and the idea of a Dynabook *metamedium*: a notebook-sized wireless-networked "personal computer for children of all ages". The real printing revolution was a qualitative change in thought and argument that lagged the hardware inventions by almost two centuries. The special quality of computers is their ability to rapidly simulate arbitrary descriptions, and the real computer revolution won't happen until children can learn to read, write, argue and think in this powerful new way. We should all try to make this happen much sooner than 200 or even 20 more years! This got me started designing computer languages and authoring environments for children, and I've been at it ever since.

Looking back on these experiences, I’m struck that my lifelong processes of loving ideas and reacting to them didn’t bear really interesting fruit until I encountered “The ARPA Dream” in grad school at the University of Utah. A fish on land still waves its fins, but the results are qualitatively different when the fish is put in its most suitable watery environment.

This is what I call "The power of the context" or "Point of view is worth 80 IQ points". Science and engineering themselves are famous examples, but there are even more striking processes within these large disciplines. One of the greatest works of art from that fruitful period of ARPA/PARC research in the 60s and 70s was the almost invisible context and community that catalysed so many researchers to be incredibly better dreamers and thinkers. That it was a great work of art is confirmed by the world-changing results that appeared so swiftly, and almost easily. That it was almost invisible, in spite of its tremendous success, is revealed by the disheartening fact today that, as far as I'm aware, no governments and no companies do edge-of-the-art research using these principles. Of course I would like be shown that I'm wrong on this last point.

Just as it is difficult to pin down all the processes that gave rise to the miracle of the United States Constitution, catching the key principles that made ARPA/PARC special has proven elusive.

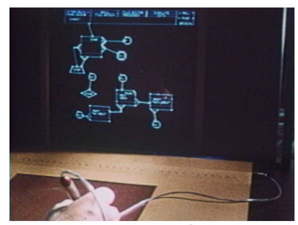


FLEX Machine self portrait, ca 1968

Ed Cheadle

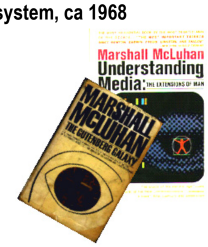
A picture that could have been taken yesterday: Doug Engelbart in 1967

“The LINC was early and small”, Wes Clark and LINC in 1963

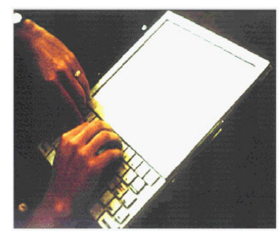


Tom Ellis’ penbased GRAIL system, ca 1968

Seymour Papert with early LOGO Turtle



McLuhan’s astounding insights about media



Dynabook Model, ca. 1968

The 4 ARPA-IPTO “Golden Age” Directors 1962-72

J.C.R. Licklider
1962-64

Ivan Sutherland
1964-66

Bob Taylor
1966-69

Larry Roberts
1969-72

We know that the designers of the Constitution were brilliant and well educated, but, as Ben Franklin pointed out at the culmination of the design, there was still much diversity of opinion and, in the end, it was the good will of the participants that allowed the whole to happen. Subsequent history has shown many times that it is the good will and belief of Americans in the Constitution that has allowed it to be such a power for good—no scrap of paper full of ideas, however great, is sufficient.

Similarly, when I think of ARPA/PARC, I think first of good will, even before brilliant people. Dave Evans, my advisor, mentor, and friend was simply amazing in his ability to act as though his graduate students were incredible thinkers. Only fools ever let him find out otherwise! I really do owe my career to Dave, and learned from him most of what I think is important. On a first visit to the Lincoln Labs ARPA project, we students were greeted by the PI Bert Sutherland, who couldn't have been happier to see us or more interested in showing us around. Not too many years later Bert was my lab manager at Xerox PARC. At UCLA, young professor Len Kleinrock became a lifelong friend from the first instant. A visit to CMU in those days would find Bill Wulf, a terrific systems designer and a guy who loved not just his students but students from elsewhere as well. If one made a pilgrimage to Doug Engelbart's diggings in Menlo Park, Bill English, the co-inventor of the mouse, would drop what he was doing to show everything to the visiting junior researchers. Later at PARC, Bill went completely out of his way to help me set up my own research group. Nicholas Negroponte visited Utah and we've been co-conspirators ever since. Bob Taylor, the director of ARPA-IPTO at that time, set up a yearly ARPA grad student conference to further embed us in the larger research processes and collegial relationships. As a postdoc, Larry Roberts got me to head a committee for an ARPAnet AI supercomputer where considerably senior people such as Marvin Minsky and Gordon Bell were theoretically supposed to be guided by me. They were amazingly graceful in how they dealt with this weird arrangement. Good will and great interest in graduate students as "world-class researchers who didn't have PhDs yet" was the general rule across the ARPA community.

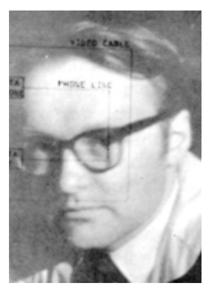
What made all this work were a few simple principles articulated and administered with considerable purity. For example, it is no exageration to say that ARPA/PARC had "visions rather than goals" and "funded people, not projects". The vision was "interactive computing as a complementary intellectual partner for people pervasively networked world-wide". By not trying to derive specific goals from this at the funding side, ARPA/PARC was able to fund rather different and sometimes opposing points of view. For example, Engelbart and McCarthy had extremely different ways of thinking of the ARPA dream, but ideas from

Dave Evans in the 60s

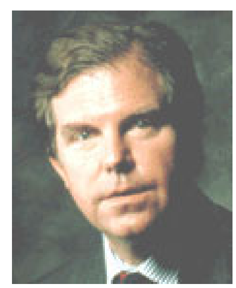
Bert Sutherland

Len Kleinrock late 60s

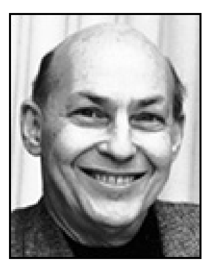
Bill Wulf



Bill English in the late 60s



Nicholas Negroponte



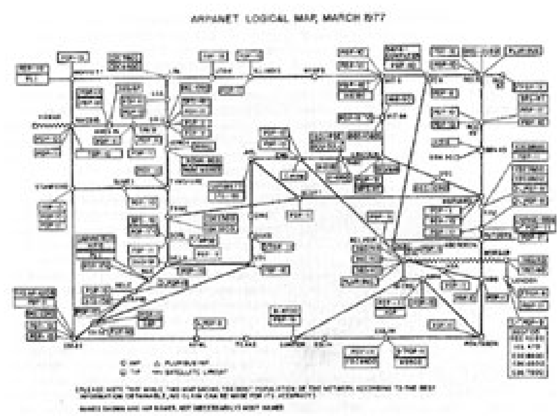
Marvin Minsky

Gordon Bell at his PDP-6, mid 60s

both of their research projects are important parts of today's interactive computing and networked world.

Giving a professional illustrator a goal for a poster usually results in what was desired. If one tries this with an artist, one will get what the artist needed to create that day. Sometimes we make, to have, sometimes to know and express. The pursuit of Art always sets off plans and goals, but plans and goals don't always give rise to Art. If "visions not goals" opens the heavens, it is important to find artistic people to conceive the projects.

Thus the "people not projects" principle was the other cornerstone of ARPA/PARC’s success. Because of the normal distribution of talents and drive in the world, a depressingly large percentage of organizational processes have been designed to deal with people of moderate ability, motivation, and trust. We can easily see this in most walks of life today, but also astoundingly in corporate, university, and government research. ARPA/PARC had two main thresholds: self-motivation and ability. They cultivated people who "had to do, paid or not" and "whose doings were likely to be highly interesting and important". Thus conventional oversight was not only not needed, but was not really possible. "Peer review" wasn't easily done even with actual peers. The situation was "out of control", yet extremely productive and not at all anarchic.



The ARPAnet itself was “out of control” in one sense — there was no centralized controller — but was perfectly convergent in what it was supposed to do

"Out of control" because artists have to do what they have to do. "Extremely productive" because a great vision acts like a magnetic field from the future that aligns all the little iron particle artists to point to “North” without having to see it. They then make their own paths to the future. Xerox often was shocked at the PARC process and declared it out of control, but they didn't understand that the context was so powerful and compelling and the good will so abundant, that the artists worked happily at their version of the vision. The results were an enormous collection of breakthroughs, some of which we are celebrating today.

Our game is more like art and sports than accounting, in that high percentages of failure are quite OK as long as enough larger processes succeed. Ty Cobb's lifetime batting average was "only" .368, which means that he failed almost 2/3s of the time. But the critical question is: what happened in the 1/3 in which he was succeeding? If the answer is "great things" then this is all the justification that should be needed. Unless I'm badly mistaken, in most processes today—and sadly in most important areas of technology research—the administrators seem to prefer to be completely in control of mediocre processes to being "out of control" with superproductive processes. They are trying to "avoid failure" rather than trying to "capture the heavens".

Ty Cobb – only 37% effective?

What if you have something cosmic you really want to accomplish and aren't smart and knowledgable enough, and don't have enough people to do it? Before PARC, some of us had gone through a few bitter experiences in which large straight-ahead efforts to create working artifacts turned out to be fragile and less than successful. It seems a bit of a stretch to characterize PARC's group of supremely confident technologists as "humble", but the attitude from the beginning combined both big ideas and projects, with a large amount of respect for how complexity can grow faster than IQs. I remember Butler, in his first few weeks at PARC, arguing as only he could that he was tired of bubble-gummed !@#$%^&* fragile research systems that could barely be demoed by their creators. He called for two general principles: that we should not make anything that was not engineered for 100 users, and we should all have to use our creations as our main computing systems (later called Living Lab). Naturally we fought him for a short while, thinking that the extra engineering would really slow things down, but we finally gave in to his brilliance and will. The scare of 100 users and having to use our own stuff got everyone to put a lot more thought early on before starting to crab together a demo. The result was almost miraculous. Many of the most important projects got to a stable, usable, and user-testable place a year or more earlier than our optimistic estimates.

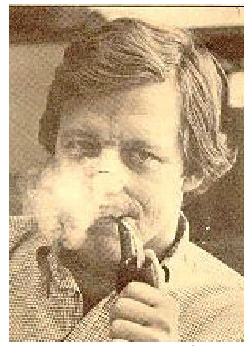
Respect for complexity, lack of knowledge, the small number of researchers and modest budgets at PARC led to a *finessing* style of design. Instead of trying to build the complex artifacts from scratch—like trying to build living things cell by cell—many of the most important projects built a kernel that could *grow* the artifact as new knowledge was gained—that is: get one cell's DNA in good shape and let it help grow the whole system.

For example: Chuck's beautiful and parsimonious architecture for the Alto allowed most functions that were normally frozen in hardware to be re-microcoded at will as new ideas came forth, without requiring the low-level HW to be redesigned and built.

The Smalltalk system that I designed, and Dan Ingalls implemented, used an important meta-idea from LISP that allowed its DNA to be completely described on one sheet of paper, implemented in a month, and then *grown* in the presence of experience and new ideas into the powerful system it became.

The bitmap display acted as "silicon paper" that could show any image and this allowed us not to have to be perfect about the kinds of graphics that could be displayed. This led directly to bitmap painting, animation and typography.

Beanbag room at PARC where all matters high and low were debated and decided

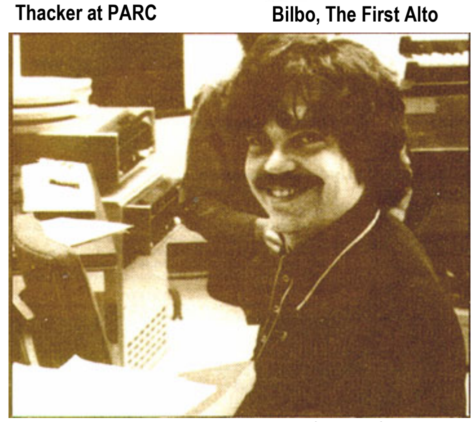


Bob Taylor at PARC: the master of social dynamics and the critical "impressario" (as Chuck likes to call him)

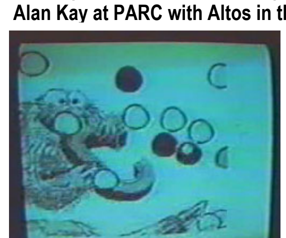
One of the most amazing people I've ever met: Butler Lampson, early days at PARC

"Mr. Make It Work": Chuck Thacker at PARC

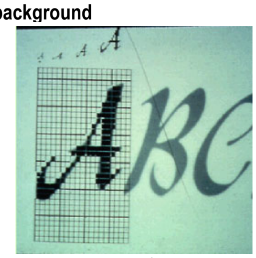
Bilbo, The First Alto



Would you trust this child with your funding? Alan Kay at PARC with Altos in the background



Smalltalk realtime 2.5D paint-ing and animation on Alto



Printing quality fonts could also be "painted"

The overlapping window interface was a finesse that tried to give children of all ages a simple universal way to communicate with anything on the computer in a form that revealed how windows were made (the original version was just 2 pages of Smalltalk).

The desktop publishing finesse was the realization that it was really just "object-graphics done right", that is, arbitrary and open-ended graphic objects that could be laid out in 2-1/2 D.

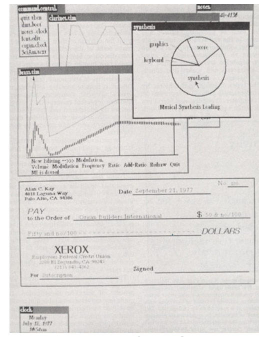
Smalltalk was a language powerful enough to write its own operating system but in the friendly form of what today would be called a scripting language. So children were also authors (our main user community) and created many interesting interactive systems. This greatly extended the wide range of user studies that were done on the Alto.

A beautiful finesse was Butler's and Charles Simonyi's approach to the text editor BRAVO (the direct precursor to MS Word). It was partly an experiment in programming and partly in trying to design a new kind of word processor. They hit on the idea of providing something everybody wanted (printing on the new laser printer), dealt with the many early bugs by guaranteeing that the system could replay right up to a crash, and provided an online complaint and suggestion service. Most versions of BRAVO—as with Smalltalk and many of the other systems at PARC—were thus heavily used during their actual incremental creation: they were *grown* into being.

Another example of finessing avoided trying to make a perfect artifact—e.g. a network that has no noise and transmits perfectly. Instead Metcalfe's and Boggs' Ethernet (codesigned by Lampson & Thacker) was set up for errors-as-normal but could always eventually send the messages perfectly, even under extreme conditions. The difference between having to make a perfect artifact and one that can eventually do something perfectly is enormous.

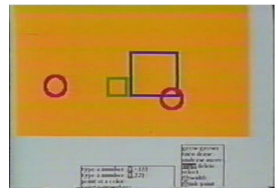
One of the keys to how all this worked was the PARC version of Catch-22, known as "Error-33". One committed Error-33 by putting any externally controlled system, in-house or out, on one's critical path. This included vendors. Error-33 was avoided by doing all that was necessary within a research group and then sharing. Thus, virtually all the PARC hardware — including two big time-sharing main frames, the Altos, Ethernet, Laserprinter, file storage, and the systems that followed — and software — including operating systems, programming languages and development systems, productivity tools, etc. — were completely built inhouse by these few dozen researchers.

This sounds disastrous, but there is an important collection of theories in which the 1st order version and the 2nd order version

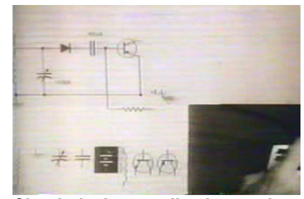


Early version of the Small-talk overlapping window GUI

Early version of desktop publishing with iconic GUI in Smalltalk



Draw application made by a 12-year-old girl in Smalltalk on the color Alto

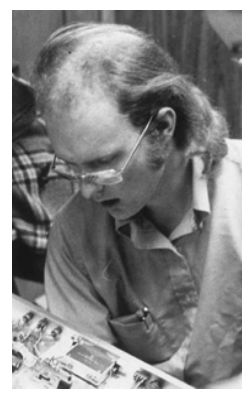


Circuit design application made by a 15-year-old boy in Smalltalk

Charles Simonyi at PARC

BRAVO WYSIWYG Display

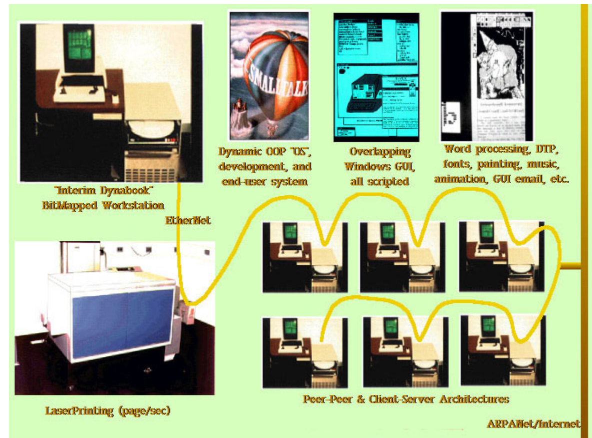
Bob Metcalfe at PARC



Dave Boggs at PARC

Gary Starkweather ca 1971 at PARC, and his hand-built first laser printer (500 pixels/inch and 1 page/second)

are completely different yet both are true. For example, in programming there is a wide-spread 1st order theory that one shouldn't build one's own tools, languages, and especially operating systems. This is true—an incredible amount of time and energy has gone down these ratholes. On the 2nd hand, if you *can* build your own tools, languages and operating systems, then you *absolutely should* because the leverage that can be obtained (and often the time not wasted in trying to fix other people's not quite right tools) can be incredible.



The "PARC genre" of Personal Computing: Alto personal computer, bit-map screen, overlapping window and icon interface, WYSIWYG word processing, email, and DTP, multimedia, end-user authoring and scripting, Ethernet, Laserprinter, Peer-Peer & Client-Server Distributed Architecture, and connections to ARPAnet/Internet.

All of these principles came together a little over 30 years ago to eventually give rise to 1500 Altos, Ethernetworked to: each other, Laserprinters, file servers and the ARPAnet, distributed to many kinds of end-users to be heavily used in real situations. This anticipated the commercial availability of this genre by 10-15 years. The best way to predict the future is to invent it.


Doug Fairbairn's/LRG's Smalltalk Notetaker ca. 1978

A few years later we had another thrill when we lugged Doug Fairbairn's Smalltalk Notetaker computer onto an airplane and did a full range of personal computing while in the air (and no flight attendents asked us to turn it off while taxiing and takeoff!). And, it's still fun today to write and publish these remarks using only descendents of the ARPA/PARC inventions. But, while we are celebrating what did make it out to the larger world, we should realize that many of the most important ARPA/PARC ideas haven't yet been adopted by the mainstream.


First Altos in a school (1975) Adele Goldberg holds forth to a classroom of enthusiastic students

For example, it is amazing to me that most of Doug Engelbart's big ideas about "augmenting the collective intelligence of groups working together" have still not taken hold in commercial systems. What looked like a real revolution twice for end-users, first with spreadsheets and then with Hypercard, didn't evolve into what will be commonplace 25 years from now, even though it could have. Most things done by most people today are still "automating paper, records and film" rather than "simulating the future". More discouraging is that most computing is still aimed at adults in business, and that aimed at nonbusiness and children is mainly for entertainment and apes the worst of television. We see almost no use in education of what is great and unique about computer modeling and computer thinking. These are not technological problems but a lack of perspective. Must we hope that the open-source software movements will put things right?

The ARPA/PARC history shows that a combination of vision, a modest amount of funding, with a felicitous context and process can almost magically give rise to new technologies that not only amplify civilization, but also produce tremendous wealth for the society. Isn't it time to do this again by Reason, even with no Cold War to use as an excuse? How about helping children of the world grow up to think much better than most adults do today? This would truly create "The Power of the Context".


Today children in many parts of the world are starting to learn the most powerful ideas of humanity by creating models of them on distributed personal computers and networks using Squeak (a direct descendent of Xerox PARC software). This work was origiinally made possible by ARPA/PARC sponsorship and is now being supported by Hewlett-Packard. Visit http://www.squeakland.org to learn more.

References

Thanks to the fulfillment of "The ARPA Dream", personal computing and networking are now ubiquitous and inexpensive, allowing many of these references to be quickly and directly accessed online by readers of these remarks.

Histories of the Alto HW & SW by its Inventors

Learning Research Group History

The following history contains a pretty full account of this work from the point of view of our research group. There is an extensive citation of acknowledgements and influences.

Kay, Alan C., "The Early History of Smalltalk", in *History of Programming Languages II*, Bergin, T, Gibson, R (editors), ACM Press, New York, 1996, pp 511-598. Includes the paper & transcripts of the presentation, discussion by Adele Goldberg, Q&A with the audience. The preprint version of the history is available online: http://www.squeakland.org/Smallhistory.pdf

___________, "The Dynabook—Past, Present, and Future" Video of Banquet Talk, *A History of Personal Workstations*, 1988, available from: ***

History of the Alto by Butler Lampson and Chuck Thacker

It's worthwhile to compare the above with the two excellent history papers by Butler Lampson and Chuck Thacker (cited below), originally published in *A History of Personal Workstations*, ed. A. Goldberg, Addison-Wesley, New York, 1988. These histories together provide three different, but pretty coherent perspectives on this work.

Lampson, Butler W., "Personal Distributed Computing: The Alto and Ethernet Software", *A History of Personal Workstations*, ed. A. Goldberg, Addison-Wesley, 1988, pages 291-344 http://research.microsoft.com/lampson/38-AltoSoftware/Abstract.html There is also a video of this talk available from: ***

Thacker, Charles P., "Personal Distributed Computing: The Alto and Ethernet Hardware", *A History of Personal Workstation*s, ed. A. Goldberg, Addison-Wesley, 1988, pages *** http://***.html. There is also a video of this talk available from: ***

Histories of Workstations and Personal Computing

For a wider view of what some of the key researchers of the larger community thought about interactive and personal computing in the 50s, 60s and 70s, it is well worth perusing the entire book *A History of Personal Workstation*s, ed. A. Goldberg, ACM Press Addison-Wesley, 1988. There are rememberances by Licklider, Wes Clark, Gordon Bell, Doug Engelbart, and many others including those who worked on the huge early SAGE systems on the one hand, and those who tried to fit calculators into a shirt pocket on the other. A complete series of video tapes of all the talks is available from ***

Early Inspirations For Dynamic Objects

Watson, J., *Molecular Biology of the Gene*, W. A. Benjamin, New York 1965 How highly complex organizations might still be able to work.

Halmos, Paul R., *Finite-dimensional Vector Spaces*, Van Nostrand, New Jersey, 1958 The power of "algebra in the large"

Carnap, Rudolf, *Meaning and Necessity, A Study in Semantics and Modal Logic*, Chicago: University of Chicago Press, 1947

Madison, Hamilton, Jay, *The Federalist Papers* http://lcweb2.loc.gov/const/fed/fedpapers.html How highly complex organizations (of people) might be made to work.

Barton, R. S., "A new approach to the functional design of a digital computer", Proc WJCC, 1961, reprinted in IEEE Annals Perhaps the greatest single advance in computer design. http://csdl.computer.org/comp/mags/an/1987/01/a1011abs.htm Full text can be viewed at: ***.***

Sutherland, Ivan, "Sketchpad, A Man-Machine Communications System", MIT PhD Thesis (1963) "When there was only one personal computer." The UR-vision: Very very early: interactive computer graphics, object-oriented design, real-time problem solving. http://www.accad.ohio-state.edu/~waynec/history/PDFs/UCAM-CL-TR-574.pdf

Dahl, O-J, and Nygaard, K., "SIMULA: an ALGOL-based simulation language", Communications of the ACM, Volume 9 , Issue 9 (September 1966, Pages: 671– 678 The catalyst than changed my POV. http://***

Wirth, N., Weber, H., “EULER: A generalization of Algol, and its definition”, CACM 9 Part I, Jan 1966, Part II, Feb 1966. How to do an algebraic programming language beautifully and simply. It almost reinvented LISP from a different POV.

Minsky, Marvin, *Computation: Finite and Infinite Machines*, Prentice-Hall, 1967 (Just a great book!)

McCarthy, John, "Recursive Functions of Symbolic Expressions and their Computation by Machine (Part I)", CACM 1960. http://www-formal.stanford.edu/jmc/recursive.html The “Maxwell’s Equations” of programming languages – the greatest single advance in programming thought.

________, et al., *The LISP 1.5 Programming Manual*, MIT Press, Cambridge 1962 (Another “just a great book!”)

Baran, Paul, RAND Reports on packet-switching and flexible routing in mesh-networks starting in early 60s: http://www.rand.org/publications/RM/baran.list.html, http://www.rand.org/publications/RM/RM3420/

Kleinrock, Leonard, *Communication Nets: stochastic message flow and delay*, McGraw-Hill, New York, 1964

Thomas Marill & Lawrence Roberts, "Toward a Cooperative Network of Time-Shared Computers", Fall AFIPS Conf (Oct 1966) http://www.packet.cc/files/toward-coop-net.html One of the early papers on the route to the ARPAnet.

Early Inspirations For Personal Computing & Networks

Bush, Vannevar, “As We May Think”, *Atlantic Monthly* (1945), also in: *From Memex to Hypertext: Vannevar Bush and the Mind’s Machine*, edited by James M, Nyce and Paul Kahn. San Diego: Academic Press. http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm

Licklider, J.C.R., “Man-Computer Symbiosis”, *IRE Transactions on Human Factors in Electronics (1960)* Reprinted in "In Memoriam: J.C.R. Licklider, 1915-1990". Digital Systems Research Center Reports, vol. 61. Palo Alto, Ca, 1990. The UR-paper on the “ARPA Dream” http://www.memex.org/licklider.pdf

J.C.R. Licklider and Robert W. Taylor, "The Computer as a Communication Device", *Science and Technology*, April 1968. Reprinted in "In Memoriam: J.C.R. Licklider, 1915-1990". Digital Systems Research Center Reports, vol. 61. Palo Alto, Ca, 1990. http://www.memex.org/licklider.pdf

Sutherland, Ivan, "Sketchpad, A Man-Machine Communications System", MIT PhD Thesis (1963). “When there was only one personal computer.” The UR-vision: Very very early: interactive computer graphics, object-oriented design, real-time problem solving. http://www.accad.ohio-state.edu/~waynec/history/PDFs/UCAM-CL-TR-574.pdf

Clark, Wesley, "The LINC was early and small", *A History of Personal Workstations*, ed. A. Goldberg, Addison-Wesley, 1988 http://***.*** — a terrific retrospective by a true pioneer: the main designer of both the huge TX-2 computer on which computer graphics was born and the smallish LINC (my vote for the first real personal computer).

Shaw, Cliff, "JOSS: a designer's view of an experimental online computer system", RAND, 1964 One of the first truly beautiful interactive systems for end-users that really cared about them in every possible way. A classic.

Engelbart, Douglas C., and English, W. K., “A research center for augmenting human intellect”, *Proceedings of the FJCC,* Vol 33, Part one, (pp 395-410). December, 1968 — This is the companion paper to perhaps the greatest public demo of an interactive computing system: to 3000 attendees of the 1968 FJCC in San Francisco.

Engelbart, Douglas C., "The Augmented Knowledge Workshop," in *A History of Personal Workstations*, ed. A. Goldberg, ACM Press, New York, 1988, pp. 185-236 — an excellent retrospective.

Engelbart, Douglas C., "The Augmented Knowledge Workshop," (82-min. VHS video cassette recording) Doug Engelbart’s presentation at the ACM Conference on the History of Personal Workstations, Palo Alto, CA, January 9-10, 1986; Includes 20 minutes from the historic 1968 FJCC demonstration

Tom O. Ellis, J.F. Heafner, W.L. Sibley, The GRAIL Project: An Experiment in Man-Machine Communications. RAND Corporation, Santa Monica CA, 1969 The first really great “intimate” GUI using gesture recognition. A classic. http://www.rand.org/cgi-bin/Abstracts/ordi/getabbydoc.pl?doc=RM-5999&hilite=1&qs=GRAIL

Milestones on the road to the ARPAnet.

Baran, Paul, RAND Reports on packet-switching and flexible routing in mesh-networks starting in early 60s: http://www.rand.org/publications/RM/baran.list.html, http://www.rand.org/publications/RM/RM3420/

Kleinrock, Leonard, *Communication Nets: stochastic message flow and delay*, McGraw-Hill, New York, 1964

Thomas Marill & Lawrence Roberts, "Toward a Cooperative Network of Time-Shared Computers", Fall AFIPS Conf (Oct 1966) Early test of packet-switching. http://www.packet.cc/files/toward-coop-net.html

Early Education and Dynabook Inspirations and Influences

An early oral version (in 1968) of Minsky's Turing Lecture made a great impression on me in many areas, especially the parts about learning and education.

Minsky, Marvin, "Form and Content in Computer Science", 1970 ACM Turing Award Lecture, Journal of the Association for Computing Machinery, Vol. 17, No. 2, April 1970. http://web.media.mit.edu/~minsky/papers/TuringLecture/TuringLecture.html

Papert, early papers *** These are really great ideas, and were the catalyst to my now 35 year interest with "helping children learn to grow up to think better than most adults do today".

Bruner, Jerome, *Toward A Theory Of Instruction*, Harvard-Belknap Press, 1965. Still the best single book on how to think about and design learning environments (precomputer, but still the best).

McLuhan, Marshall, *The Gutenberg Galaxy: The Making of Typographic Man*, University of Toronto Press, 1965, I had to learn to understand this one first before being able to grok Understanding Media

________, *Understanding Media: Extensions of Man*, Signet Press, 1964 You have to work to extract the gold from the dross, but the gold is 100% pure and critically important for humans to understand.

Selected Historical Papers About The Dynabook

Kay, Alan C., "A personal computer for children of all ages", Proc. ACM National Conf, Boston, Aug 1972

__________, "A dynamic medium for creative thought", National Teachers of English Conf, Nov 1972

__________, and Goldberg, Adele, "Personal Dynamic Media", IEEE Computer, March 1977

__________, "Microelectronics and the personal computer", *Scientific American*, Sept 1977

__________, "Programming your own computer", Science Year 1979, *WorldBook Encyclopedia, 1979*

US History

Franklin, B., " Speaking before the Convention in Philadelphia, 1787", http://www.lexrex.com/enlightened/writings/franklin_on_const.htm

Madison, Hamilton, Jay, *The Federalist Papers* http://lcweb2.loc.gov/const/fed/fedpapers.html

Selected Other Important PARC Research Documentation

Starkweather, Gary, ***

Metcalfe, R.M., Boggs, D.R., "Ethernet: Distributed Packet Switching For Local Computer Networks", *Communications of the ACM*, Vol 19, Num 7, July 1976, online at ACM Digital Library

Ingalls, Dan. H.H., “The Smalltalk-76 Programming System, Design and Implementation”, in 5th ACM *Symposium on Principles of Programming Languages*, Tucson, Jan 1978, online at ACM Digital Library, and at: http://users.ipa.net/~dwighth/smalltalk/St76/Smalltalk76ProgrammingSystem.html

Goldberg, Adele and Robson, David, *Smalltalk-80: The Language and its Implementation*, Addison-Wesley, 1983 How to do what we did.

The ARPA-IPTO/PARC History and Community as seen from the outside

None of these articles and books quite captures the zeitgeist. The "infamous" Rolling Stone article by Stewart Brand perhaps comes the closest. The Perry article about PARC is pretty good, and the Waldrop book gives a large and detailed picture of Licklider and what he was able to start and influence. Chigusa Kita’s history paper about Licklider is the most meticulously researched, by an extremely careful and diligent historian. The Rheingold book has a pretty good perspective from much earlier interviews.

Brand, Stewart, "Fanatic Life & Symbolic Death Among the Computer Bums", *Rolling Stone Magazine,* Dec 1972. http://www.wheels.org/spacewar/stone/rolling_stone.html

Perry, Tekla S., Wallich, P., "Inside the PARC: The 'Information Architects' " IEEE Spectrum (October 1985)

Reingold, Howard, *Tools For Thought*, New York: Simon & Schuster, 1985 -- An interesting interview with Bob Taylor is at: http://www.rheingold.com/texts/tft/10.html

Hiltzik, Michael, *Dealers of Lightning: Xerox PARC and the dawn of the computer age,* New York: Harper-Business, 1999

Waltrop, M. Mitchell, *The Dream Machine: J.C.R. Licklider and the revolution that made computing personal*, New York: Viking, 2001

Chigusa Ishikawa Kita, “J. C. R. Licklider 's Vision for the IPTO,” *IEEE Annals of the History of Computing*, vol.25, no.3, pp.61-77.

================