A

Dr. Dobb's Interview with Alan Kay

Dr. Dobb's Website, July 10, 2012

A.1 A Note About Dr. Dobb's Journal

The Dr. Dobb's journal was a very popular magazine among programmers. Launched in 1976, it was the first software-oriented monthly magazine targeting practitioners. Unfortunately, December 2014 saw its last paper edition, while the website and code repository survive on-line. Most likely the magazine's existence was jeopardized by the Web itself, and the way advertising campaigns changed over the years.

I held a subscription for about a decade, starting in the mid-1990s. I even published a paper in it about Web-Service Notification with two undergraduate students in 2005 [5]. I jokingly refer to that paper as my most important scientific contribution. Could be. For sure, it is the only journal or magazine paper I ever wrote for which the publisher paid me.

© Springer International Publishing AG, part of Springer Nature 2018
M. Aiello, *The Web Was Done by Amateurs*,
https://doi.org/10.1007/978-3-319-90008-7

A.2 The Interview

The pioneer of object-orientation, co-designer of Smalltalk, and UI luminary opines on programming, browsers, objects, the illusion of patterns, and how Socrates could still make it to heaven.

By Andrew Binstock

In June of this year, the Association of Computing Machinery (ACM) celebrated the centenary of Alan Turing's birth by holding a conference with presentations by more than 30 Turing Award winners. The conference was filled with unusual lectures and panels [...] both about Turing and present-day computing. During a break in the proceedings, I interviewed Alan Kay—a Turing Award recipient known for many innovations and his articulated belief that the best way to predict the future is to invent it.[1]

Childhood as a Prodigy

Binstock:	*Let me start by asking you about a famous story. It states that you'd read more than 100 books by the time you went to first grade. This reading enabled you to realize that your teachers were frequently lying to you.*
Kay:	Yes, that story came out in a commemorative essay I was asked to write.
Binstock:	*So you're sitting there in first grade, and you're realizing that teachers are lying to you. Was that transformative? Did you all of a sudden view the whole world as populated by people who were dishonest?*
Kay:	Unless you're completely, certifiably insane, or a special kind of narcissist, you regard yourself as normal. So I didn't really think that much of it. I was basically an introverted type, and I was already following my own nose, and it was too late. I was just stubborn when they made me go along.
Binstock:	*So you called them on the lying.*

(continued)

[1]A side note: Re-creating Kay's answers to interview questions was particularly difficult. Rather than the linear explanation in response to an interview question, his answers were more of a cavalcade of topics, tangents, and tales threaded together, sometimes quite loosely—always rich, and frequently punctuated by strong opinions. The text that follows attempts to create somewhat more linearity to the content.—ALB.

Kay: Yeah. But the thing that traumatized me occurred a couple years later, when I found an old copy of Life magazine that had the Margaret Bourke-White photos from Buchenwald. This was in the 1940s—no TV, living on a farm. That's when I realized that adults were dangerous. Like, really dangerous. I forgot about those pictures for a few years, but I had nightmares. But I had forgotten where the images came from. Seven or eight years later, I started getting memories back in snatches, and I went back and found the magazine. That probably was the turning point that changed my entire attitude toward life. It was responsible for getting me interested in education. My interest in education is unglamorous. I don't have an enormous desire to help children, but I have an enormous desire to create better adults.

The European Invasion in Computer Science

Kay: You should talk to William Newman, since he's here. He was part of the British brain-drain. There was also Christopher Strachey, whom I consider one of the top 10 computer scientists of all time. The British appreciate him. They also had Peter Landin. They had memory management and they had timesharing before we did. Then there was a crisis in the early 1960s. And suddenly the young Brits were coming to the United States.

William was one of the guys who literally wrote the book on computer graphics: Principles of Interactive Computer Graphics with Robert Sproull. William came to Harvard and was Ivan Sutherland's graduate student—got his Ph.D. in 1965 or 1966. William followed Ivan out to Utah; then when Xerox PARC was set up, William came to PARC.

A similar thing happened, but I think for different reasons, in France. So one of the things we benefited from is that we got these incredibly well-prepared Brits and French guys reacting to the kind of devil-may-care attitude, and funding like nobody had ever seen before. These guys were huge contributors. For example, the first outline fonts were done by Patrick Baudelaire at PARC, who got his Ph.D. at Utah. The shading on 3D is named Gouraud shading after Henri Gouraud, who was also at Utah—also under Ivan, when Ivan was there.

Computing as Pop Culture

Binstock: *You seem fastidious about always giving people credit for their work.*

Kay: Well, I'm an old-fashioned guy. And I also happen to believe in history. The lack of interest, the disdain for history is what makes computing not-quite-a-field.

Binstock: *You once referred to computing as pop culture.*

Kay: It is. Complete pop culture. I'm not against pop culture. Developed music, for instance, needs a pop culture. There's a tendency to over-develop. Brahms and Dvorak needed gypsy music badly by the end of the nineteenth century. The big problem with our culture is that it's being dominated, because the electronic media we have is so much better suited for transmitting pop-culture content than it is for high-culture content. I consider jazz to be a developed part of high culture. Anything that's been worked on and developed and you [can] go to the next couple levels.

Binstock: *One thing about jazz aficionados is that they take deep pleasure in knowing the history of jazz.*

Kay: Yes! Classical music is like that, too. But pop culture holds a disdain for history. Pop culture is all about identity and feeling like you're participating. It has nothing to do with cooperation, the past or the future—it's living in the present. I think the same is true of most people who write code for money. They have no idea where [their culture came from]—and the Internet was done so well that most people think of it as a natural resource like the Pacific Ocean, rather than something that was man-made. When was the last time a technology with a scale like that was so error-free? The Web, in comparison, is a joke. The Web was done by amateurs.

The Browser: A Lament

Binstock: *Still, you can't argue with the Web's success.*

Kay: I think you can.

Binstock: *Well, look at Wikipedia—it's a tremendous collaboration.*

Kay: It is, but go to the article on Logo, can you write and execute Logo programs? Are there examples? No. The Wikipedia people didn't even imagine that, in spite of the fact that they're on a computer. That's why I never use PowerPoint. PowerPoint is just simulated acetate overhead slides, and to me, that is a kind of a moral crime.

(continued)

That's why I always do, not just dynamic stuff when I give a talk, but I do stuff that I'm interacting with on-the-fly. Because that is what the computer is for. People who don't do that either don't understand that or don't respect it.

The marketing people are not there to teach people, so probably one of the most disastrous interactions with computing was the fact that you could make money selling simulations of old, familiar media, and these apps just swamped most of the ideas of Doug Engelbart, for example. The Web browser, for many, many years, and still, even though it's running on a computer that can do X, Y, and Z, it's now up to about X and 1/2 of Y.

Binstock: *How do you mean?*

Kay: Go to a blog, go to any Wiki, and find one that's WYSIWYG like Microsoft Word is. Word was done in 1984. HyperCard was 1989. Find me Web pages that are even as good as HyperCard. The Web was done after that, but it was done by people who had no imagination. They were just trying to satisfy an immediate need. There's nothing wrong with that, except that when you have something like the Industrial Revolution squared, you wind up setting de facto standards—in this case, really bad de facto standards. Because what you definitely don't want in a Web browser is any features.

Binstock: *"Any features?"*

Kay: Yeah. You want to get those from the objects. You want it to be a mini-operating system, and the people who did the browser mistook it as an application. They flunked Operating Systems 101.

Binstock: *How so?*

Kay: I mean, look at it: The job of an operating system is to run arbitrary code safely. It's not there to tell you what kind of code you can run. Most operating systems have way too many features. The nice thing about UNIX when it was first done is not just that there were only 20 system commands, but the kernel was only about 1,000 lines of code. This is true of Linux also.

Binstock: *Yes.*

Kay: One of the ways of looking at it is the reason that WYSIWYG is slowly showing up in the browser is that it's a better way of interacting with the computer than the way they first did it. So of course they're going to reinvent it. I like to say that in the old days, if you reinvented the wheel, you would get your wrist slapped for not reading. But nowadays people are reinventing the flat tire. I'd personally be happy if they reinvented the wheel, because at least we'd be moving forward. If they reinvented what Engelbart did, we'd be way ahead of where we are now.

Objects

Kay: The flaw there is probably the fact that C is early-bound. Because it's not late-bound, because it's not a dynamic system, pretty much the only way you can link in features is to link them in ahead of time. Remember when we had to boot the computer? There's no need for that. There's never been any need for it. Because they did it that way, you wind up with megabytes of features that are essentially bundled together whether you want them or not. And now a thousand system calls, where what you really want is objects that are migrating around the net, and when you need a resource, it comes to you—no operating system. We didn't use an operating system at PARC. We didn't have applications either.

Binstock: *So it was just an object loader?*

Kay: An object exchanger, really. The user interface's job was to ask objects to show themselves and to composite those views with other ones.

Binstock: *You really radicalized the idea of objects by making everything in the system an object.*

Kay: No, I didn't. I mean, I made up the term "objects." Since we did objects first, there weren't any objects to radicalize. We started off with that view of objects, which is exactly the same as the view we had of what the Internet had to be, except in software. What happened was retrograde. When C++ came out, they tried to cater to C programmers, and they made a system that was neither fish nor fowl. And that's true of most of the things that are called object-oriented systems today. None of them are object-oriented systems according to my definition. Objects were a radical idea, then they got retrograded.

Binstock: *How do you view the Actor model?*

Kay: The first Smalltalk was presented at MIT, and Carl Hewitt and his folks, a few months later, wrote the first Actor paper. The difference between the two systems is that the Actor model retained more of what I thought were the good features of the object idea, whereas at PARC, we used Smalltalk to invent personal computing. It was actually a practical programming language as well as being interesting theoretically. I don't think there were too many practical systems done in Actors back then.

Programming

Binstock: *Are you still programming?*

Kay: I was never a great programmer. That's what got me into making more powerful programming languages. I do two kinds of programming. I do what you could call metaprogramming, and programming as children from the age of 9 to 13 or 14 would do. I spend a lot of time thinking about what children at those developmental levels can actually be powerful at, and what's the tradeoff between…Education is a double-edged sword. You have to start where people are, but if you stay there, you're not educating.

The most disastrous thing about programming—to pick one of the 10 most disastrous things about programming—there's a very popular movement based on pattern languages. When Christopher Alexander first did that in architecture, he was looking at 2,000 years of ways that humans have made themselves comfortable. So there was actually something to it, because he was dealing with a genome that hasn't changed that much. I think he got a few hundred valuable patterns out of it. But the bug in trying to do that in computing is the assumption that we know anything at all about programming. So extracting patterns from today's programming practices ennobles them in a way they don't deserve. It actually gives them more cachet.

The best teacher I had in graduate school spent the whole semester destroying any beliefs we had about computing. He was a real iconoclast. He happened to be a genius, so we took it. At the end of the course, we were free because we didn't believe in anything. We had to learn everything, but then he destroyed it. He wanted us to understand what had been done, but he didn't want us to believe in it.

Binstock: *Who was that?*

Kay: That was Bob Barton, who was the designer of the Burroughs B5000. He's at the top of my list of people who should have received a Turing Award but didn't. The award is given by the Association for Computing Machinery (ACM), so that is ridiculous, but it represents the academic bias and software bias that the ACM has developed. It wasn't always that way. Barton was probably the number-one person who was alive who deserved it. He died last year, so it's not going to happen unless they go to posthumous awards.

Binstock: *I don't think they do that.*

Kay: They should. It's like the problem Christian religions have with how to get Socrates into heaven, right? You can't go to heaven unless you're baptized. If anyone deserves to go to heaven, it's Socrates, so this is a huge problem. But only the Mormons have solved this—and they did it. They proxy-baptized Socrates.

(continued)

Binstock: *I didn't realize that. One can only imagine how thankful Socrates must be.*

Kay: I thought it was pretty clever. It solves a thorny problem that the other churches haven't touched in 2,000 years.

Group Work

Kay: Have you interviewed Vint Cerf?

Binstock: *No.*

Kay: He's a very special guy. Not just for brains. He's one of the better organizers of people. If you had to point to one person, given that the Internet was a community effort, the one who made that community work was Vint. And he also was the co-guy on TCP/IP. I love him. I've known him for years. He runs a pretty tough, pretty organized meeting, but he does it so well that everyone likes it.
[Digression on who, in addition to Cerf, should have won various computing prizes...]
The prizes aren't a thing that Dr. Dobb's worries about, because prizes are mostly for individuals, not for teams that are trying to do serious engineering projects. The dynamics are very different. A lot of people go into computing just because they are uncomfortable with other people. So it is no mean task to put together five different kinds of Asperger's syndrome and get them to cooperate. American business is completely fucked up because it is all about competition. Our world was built for the good from cooperation. That is what they should be teaching.

Binstock: *That's one of the few redeeming things about athletics.*

Kay: Absolutely! No question. Team sports. It's the closest analogy. Everyone has to play the game, but some people are better at certain aspects.

References

1. Achrekar, H., Gandhe, A., Lazarus, R., Yu, S.-H., & Liu, B. (2011). Predicting flu trends using twitter data. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 702–707). New York: IEEE.
2. Aiello, M. (2017). *two bestsellers*. The Netherlands: Saccargia Holding BV Publisher.
3. Aiello, M., Monz, C., Todoran, L., & Worring, M. (2002). Document understanding for a broad class of documents. *International Journal of Document Analysis and Recognition, 5*(1), 1–16.
4. Aiello, M., Papazoglou, M., Yang, J., Carman, M., Pistore, M., Serafini, L., et al. (2002). A request language for web-services based on planning and constraint satisfaction. In *VLDB Workshop on Technologies for E-Services (TES)*. Lecture notes in computer science (pp. 76–85). Berlin: Springer.
5. Aiello, M., Zanoni, M., & Zolet, A. (2005). Exploring WS-notification: Building a scalable domotic infrastructure. *Dr. Dobb's Journal, 371*, 48–51.
6. Antoniou, G., & van Harmelen, F. (2004). *A semantic web primer*. Cambridge: MIT Press.
7. Asimov, I. (1950). *I, Robot*. New York: Spectra.
8. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge: Cambridge University Press.
9. Bak, P. (1996). *How nature works: The science of self-organized criticality*. New York: Copurnicus.
10. Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286*(5439), 509–512.
11. Barnes, S. B. (2003). ACM Turing Award biography. http://amturing.acm.org/award_winners/kay_3972189.cfm

12. Barnett, R. (2013). *Web application defender's cookbook: Battling hackers and protecting users*. London: Wiley.
13. Barone, G., & Mocetti, S. (2016). What's your (sur)name? Intergenerational mobility over six centuries. http://voxeu.org/article/what-s-your-surname-intergenerational-mobility-over-six-centuries
14. Battelle, J. (2005). The birth of Google. *Wired, 13*(8), 108.
15. BBC News. Web's inventor gets a knighthood. http://news.bbc.co.uk/2/hi/technology/3357073.stm
16. BBC News. Creator of the web turns knight. http://news.bbc.co.uk/2/hi/technology/3899723.stm
17. Becchetti, L., & Castillo, C. (2006). The distribution of PageRank follows a power-law only for particular values of the damping factor. In *Proceedings of the 15th International Conference on World Wide Web* (pp. 941–942). New York: ACM.
18. Beck, M. T., Werner, M., Feld, S., & Schimper, T. (2014). Mobile edge computing: A taxonomy. In *Proceedings of the Sixth International Conference on Advances in Future Internet*. Citeseer.
19. Berners-Lee, T., & Fischetti, M. (1999). *Weaving the web: The original design and ultimate destiny of the world wide web by its inventor*. San Francisco: Harper.
20. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American, 284*(5), 28–37.
21. Bianchini, M., Gori, M., & Scarselli, F. (2005). Inside pagerank. *ACM Transactions on Internet Technology, 5*(1), 92–128.
22. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* (pp. 13–16). New York: ACM.
23. Borges, J. L. (1962). The garden of forking paths. In *Collected fictions* (pp. 119–128). New York: Grove Press.
24. Bosch, A., Bogers, T., & Kunder, M. (2016). Estimating search engine index size variability: A 9-year longitudinal study. *Scientometrics, 107*(2), 839–856.
25. Bouguettaya, A., Singh, M., Huhns, M., Sheng, Q. Z., Dong, H., Yu, Q., et al. (2017). A service computing manifesto: The next 10 years. *Communications of the ACM, 60*(4), 64–72.
26. Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1), 107–117.
27. Broder, A. Z., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. (2000). Graph structure in the web. *Computer Networks, 33*(1–6), 309–320.
28. Brown, M. H., & Najork, M. A. (1996). Distributed active objects. *Computer Networks and ISDN Systems, 28*(7), 1037–1052. Proceedings of the Fifth International World Wide Web Conference 6–10 May 1996.
29. Bush, V. (1945). As we may think. *Atlantic Monthly, 176*, 101–108.
30. Calvino, I. (2010). *The castle of crossed destinies*. New York: Random House.
31. Camden, R. (2015). *Client-side data storage*. Sebastopol: O'Reilly.

32. Campbell, M., Hoane, A. J., & Hsu, F.-H. (2002). Deep blue. *Artificial Intelligence, 134*(1–2), 57–83 (2002).
33. Cardelli, L. (1995). A language with distributed scope. In *Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (pp. 286–297). New York: ACM.
34. Cerf, V. G., & Kahn, R. E. (1974). A protocol for packet network intercommunication. *IEEE Transactions on Communication, 22*(5), 637–648.
35. Cerf, V. G., & Kahn, R. E. (2005). A protocol for packet network intercommunication. *ACM SIGCOMM Computer Communication Review, 35*(2), 71–82.
36. Chaffey, D. (2017, March). Mobile marketing statistics compilation. Technical report, Smart Insights. http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/
37. Chappell, D. (2004). *Enterprise service bus*. Sebastopol: O'Reilly Media, Inc.
38. Chen, F., Chen, Z., Wang, X., & Yuan, Z. (2008). The average path length of scale free networks. *Communications in Nonlinear Science and Numerical Simulation, 13*(7), 1405–1410.
39. Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM, 13*(6), 377–387.
40. Computer Museum. (2017). DEC PDP-1. http://www.computermuseum.li/Testpage/DEC-PDP1-1960.htm
41. Conklin, J. (1987). Hypertext: A survey and introduction. *Computer, 20*(9), 17–41.
42. Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed systems: Concepts and design*. Reading: Addison-Wesley.
43. Dever, J. (1984). *Lone Wolf. Flight from the dark*. Boston: Sparrow Books.
44. Donini, F. M., Lenzerini, M., Nardi, D., & Nutt, W. (1991). Tractable concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (Vol. 91, pp. 458–463).
45. Edwards, O. (1997, August 25). Ted Nelson. *Forbes*.
46. Engelbart, D. C. (1962). Augmenting human intellect: A conceptual framework. Technical Report Summary Report AFOSR-3223, Stanford Research Institute.
47. Fall, K. R., & Stevens, W. R. (2011). *TCP/IP illustrated, volume 1: The protocols*. Reading: Addison-Wesley.
48. Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine.
49. Fitzgerald, M. (2008, May 25). Cloud computing: So you don't have to stand still. http://www.nytimes.com/2008/05/25/technology/25proto.html
50. Franceschet, M. (2011). PageRank: Standing on the shoulders of giants. *Communications of the ACM, 54*(6), 92–101.
51. Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., et al. (2015). Edge-centric computing: Vision and challenges. *ACM SIGCOMM Computer Communication Review, 45*(5), 37–42.

52. Garrett, J. J. (2005). Ajax: A new approach to web applications. http://adaptivepath.org/ideas/ajax-new-approach-web-applications/
53. Geddes, M. How far can the Internet scale? http://www.martingeddes.com/how-far-can-the-internet-scale/
54. Georgievski, I., & Aiello, M. (2015). HTN planning: Overview, comparison, and beyond. *Artificial Intelligence, 222*, 124–156.
55. Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. Burlington: Morgan Kaufmann.
56. Gray, M. (1995). Measuring the growth of the web: June 1993 to June 1995. http://www.mit.edu/people/mkgray/growth
57. Green, J. L., & Peters, D. J. (1985). Introduction to the space physics analysis network (span). Technical Report NASA-TM-86499, NAS 1.15:86499, NASA.
58. Hafner, K., & Lyon, M. (1996). *Where wizards stay up late: The origins of the Internet* (1st ed.). New York: Simon & Schuster.
59. Hayes, B. (2008). Cloud computing. *Communications of the ACM, 51*(7), 9–11.
60. Henning, M. (2008). The rise and fall of CORBA. *Communication of the ACM, 51*(8), 53–57.
61. History of Computer: TCP/IP. http://www.history-computer.com/Internet/Maturing/TCPIP.html
62. IETF and ISOC. RFC official repository. https://www.rfc-editor.org/
63. Internet Society. Brief history of the Internet. http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet
64. Isaacson, W. (2014). *The innovators. How a group of hackers, geniuses and geeks created the digital revolution*. New York: Simon & Schuster.
65. Kaldeli, E., Lazovik, A., & Aiello, M. (2016). Domain-independent planning for services in uncertain and dynamic environments. *Artificial Intelligence, 236*(7), 30–64 (2016).
66. Kaldeli, E., Warriach, E., Lazovik, A., & Aiello, M. (2013). Coordinating the web of services for a smart home. *ACM Transactions on the Web, 7(2)*, 10.
67. Kapur, A. (2017, February 22). Mobile apps vs. mobile web: Do you have to choose?. https://www.business.com/articles/mobile-apps-vs-mobile-web-do-you-have-to-choose/
68. Kay, A. C. (1972) A personal computer for children of all ages. In *Proceedings of the ACM Annual Conference-Volume 1* (p. 1). New York: ACM.
69. Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A. S. (1999). *The web as a graph: Measurements, models, and methods* (pp. 1–17). Berlin: Springer.
70. Knoblock, C. A., Minton, S., Ambite, J.-L., Ashish, N., Muslea, I., Philpot, A. G., et al. (2001). The Ariadne approach to web-based information integration. *International the Journal on Cooperative Information Systems, 10*(1–2), 145–169 (2001).
71. Kristol, D. M. (2001). Http cookies: Standards, privacy, and politics. *ACM Transactions on Internet Technology, 1*(2), 151–198.

72. Kuhn, T. (1962). *The structure of scientific revolutions*. Chicago: University of Chicago Press.
73. LaMonica, M. (2004, September). Sarvega brings routing to XML. https://www.cnet.com/au/news/sarvega-brings-routing-to-xml/
74. Langville, A. N., & Meyer, C. D. (2006) *Google's PageRank and beyond: The science of search engine rankings*. Princeton: Princeton University Press.
75. Laporte, L. (2016, April 25). Triangulation 247 interview with Bill Atkinson. https://twit.tv/shows/triangulation/episodes/247
76. Lazer, D., Kennedy, R., King, G., & Vespignani, A. (2014). The parable of Google Flu: Traps in big data analysis. *Science, 343*(6176), 1203–1205.
77. Lazovik, A., Aiello, M., & Papazoglou, M. (2003). Planning and monitoring the execution of Web service requests. In *Proceedings of the 1st International Conference on Service-Oriented Computing (ICSOC)* (pp. 335–350). Berlin: Springer.
78. Lazovik, A., Aiello, M., & Papazoglou, M. (2006). Planning and monitoring the execution of Web service requests. *Journal on Digital Libraries, 6*(3), 235–246.
79. Leontief, W. W. (1941). *Structure of American economy, 1919–1929*. Cambridge: Harvard University Press.
80. Licklider, J. (1960). Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics, 1*, 4–11.
81. Licklider, J. (1963, April 23). Memorandum for: Members and affiliates of the intergalactic computer network. Technical Report, ARPA.
82. Limer, E. (2014). My brief and curious life as a Mechanical Turk. http://gizmodo.com/
83. Lubbers, P., & Greco, F. (2010). HTML5 Web sockets: A quantum leap in scalability for the Web. *SOA World Magazine, (1).*
84. Lyons, D. A. (2012). Net neutrality and nondiscrimination norms in telecommunications. *Arizona Law Review, 54*, 1029.
85. Manning, R. (2004, May 20). Dynamic and distributed managed edge computing (Mec) framework. US Patent App. 10/850,291.
86. McDermott, D. (2002). Estimated-regression planning for interactions with Web Services. In *6th International Conference on AI Planning and Scheduling*. Palo Alto: AAAI Press.
87. McIlarth, S., & Son, T. C. (2002). Adapting Golog for composition of semantic web-services. In *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning (KR)* (pp. 482–496). Burlington: Morgan Kaufmann.
88. Milgram, S. (1967). The small world problem. *Psychology Today, 1*, 61–67.
89. Mims, C. (2014, November 17). The web is dying; apps are killing it. *Wall Street Journal.*
90. Minkowski, M. S., & Powell, J. C. (2014). *Single page web applications*. Shelter Island: Manning.

91. Nelson, T. H. (1965). Complex information processing: A file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th National Conference* (pp. 84–100). New York: ACM.
92. Nelson, T. H. (1980). *Literary machines 93.1.: The report on, and of, project Xanadu concerning word processing, electronic publishing, hypertext, thinkertoys, tomorrow's intellectual revolution, and certain other topics including knowledge, education and freedom (1993 Edition)*. Sausalito: Mindful Press.
93. Nelson, T. H. (2007, May 17). Intertwingularity: When ideas collide uploaded. Ted Nelson's 70th Birthday Lecture.
94. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review, 45*(2), 167–256.
95. Pagani, G. A., & Aiello, M. (2013). The power grid as a complex network: A survey. *Physica A: Statistical Mechanics and Its Applications, 392*(1), 2688–2700.
96. Papazoglou, M. (2008). *Web services: Principles and technology*. London: Pearson Education.
97. Papazoglou, M., & Georgakopoulos, D. (2003). Service-oriented computing. *Communications of the ACM, 46*(10), 24–28.
98. Philp, R. K. (1865). *Enquire within upon everything*. London: Houlston and Wright.
99. Pinski, G., & Narin, F. (1976). Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Information Processing & Management, 12*(5), 297–312.
100. Raymond, E. S. (1996). *The new hacker's dictionary*. Cambridge: MIT Press.
101. Saltzer, J. H., Reed, D. P., & Clark, D. D. (1984). End-to-end arguments in system design. *ACM Transactions on Computer Systems, 2*(4), 277–288.
102. Seeley, J. R. (1949). The net of reciprocal influence. A problem in treating sociometric data. *Canadian Journal of Experimental Psychology, 3*, 234.
103. Seetharaman, K. (1998). The CORBA connection - Introduction to the CORBA special issue. *Communications of the ACM, 41*(10), 34–36.
104. Segal, B. (1985). A short history of Internet protocols at CERN. http://ben.web.cern.ch/ben/TCPHIST.html
105. Sink, E. Memories from the browser wars. http://ericsink.com/Browser_Wars.html
106. Spohrer, J., Maglio, P. P., Bailey, J., & Gruhl, D. (2007). Steps toward a science of service systems. *Computer, 40*(1), 71–77.
107. Stuttard, D., & Pinto, M. (2011). *The web application hacker's handbook* (2nd ed.). London: Wiley.
108. Tilley, S. (1999). The need for speed. *Communication of the ACM, 42*(7), 23–26.
109. Travers, J., & Milgram, S. (1969). An experimental study of the small world problem. *Sociometry, 32*(4), 425–443.
110. Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433–460.
111. van Benthem, J. (1983). *The logic of time*. London: Reidel.

112. Various Authors. (2005). InnoQ's web services standards poster, version 2.0.
113. Various Authors. HTML 5 test. http://www.html5test.com
114. Varshney, U., Snow, A., McGivern, M., & Howard, C. (2002). Voice over IP. *Communication of the ACM, 45*(1), 89–96.
115. Vise, D. (2007). The Google story. *Strategic Direction, 23*(10), 192–199.
116. Vogels, W. (2001, January). How and why did amazon get into the cloud computing business? https://www.quora.com/How-and-why-did-Amazon-get-into-the-cloud-computing-business
117. Waldrop, M. (2001) *The dream machine: JCR Licklider and the revolution that made computing personal.* New York: Viking Penguin.
118. Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature, 393*(6684), 440–442.
119. Weiser, M. (1991). The computer for the 21st century. *Scientific American, 265*(3), 94–104.
120. Witten, I. H., Gori, M., & Numerico, T. (2010). *Web dragons: Inside the myths of search engine technology*. Amsterdam: Elsevier.
121. Wolf, G. (1995). The curse of Xanadu. *Wired Magazine, 3*, 137–202.
122. Woods, W. A. (1975). What's in a link: Foundations for semantic networks. In *Representation and understanding: Studies in cognitive science* (pp. 35–82). New York: Academic.
123. Woollaston, V. (March 2014). How the web was described when Berners-Lee proposed the concept to his boss 25 years ago today.
124. Wu, T. (2003). Network neutrality, broadband discrimination. *Journal on Telecommunications and High Technology Law, 2*, 141–176.
125. Zimmermann, H. (1980). OSI reference model–The ISO model of architecture for open systems interconnection. *IEEE Transactions on communications, 28*(4), 425–432.

Index