

# **Netizens: An Anthology**

## **Part II - The Past: Where Has It All Come From?**

### **Chapter 5**

### **The Vision of Interactive Computing**

### **And the Future**

by Michael Hauben

What is the reality behind all the talk about the so called “Information Superhighway”? This is an important question which U.S. government policy makers seem to be ignoring. However, understanding the history of the current global computer networks is a crucial step towards building the network of the future. There is a vision that guided the origin and development of the Internet, Usenet and other associated physical and logical networks. What is that vision?

While the global computer networks are basically young – the ARPANET started in 1969 – their 25 plus years of growth has been substantial. The ARPANET was the experimental network connecting the mainframe computers of universities and other contractors funded and encouraged by the Advanced Research Projects Agency of the U.S. Department of Defense. The ARPANET started out as a research test bed for computer networking, communication protocols, computer and data resource sharing, etc. However, what it developed into was something surprising. The widest use of the ARPANET was for computer facilitated human-human communication using electronic mail (e-mail) and discussion lists. (Popular lists included Human-Nets, Wine-Tasters and Sci-Fi Lovers lists.) The human communications achievements of ARPANET research continue to be today’s most popular usage of the Net by a growing number and variety of people through e-mail, Usenet discussion groups, mailing lists, internet relay chat, and so on. The ARPANET was the product of previous U.S. government funded research in interactive computing and time-sharing of computers.

Until the 1960s, computers operated almost exclusively in batch mode. Programmers punched or had their programs punched onto cards. Then the stack of punched cards was provided to the local computer center. The computer operator assembled stacks of cards into batches to be feed to the computer for continuous processing. Often a programmer had to wait over a day in order to see the results from his or her input. In addition, if there were any mistakes in the creation of the punched cards, the stack or part of it had to be punched again and resubmitted, which would take another day. Bugs in the code could only be discovered after an attempt to compile the code and therefore “debugging” was a slow process. This batch processing mode was a very inefficient way of utilizing the power of the computer. People began thinking of ways to alter the interface between people and computers. The idea of time-sharing developed among some in computer research communities. Time-sharing makes it possible for people to utilize a computer (then predominately the IBM mainframe) simultaneously. Time-sharing operates by giving the impression that the each user is the only one using the computer. This is executed by having the computer divvy out slices of CPU time to all the users in a rapid, sequential manner.

Crucial to the development of today's global computer networks was the vision of researchers interested in time-sharing. These researchers began to think about social issues related to time-sharing. They observed the communities that formed from the people who used time-sharing systems and considered the social significance of these communities. Two of the pioneers involved in research in time-sharing at MIT, Fernando Corbato and Robert Fano, wrote, "The time-sharing computer system can unite a group of investigators in a cooperative search for the solution to a common problem, or it can serve as a community pool of knowledge and skill on which anyone can draw according to his needs. Projecting the concept on a large scale, one can conceive of such a facility as an extraordinarily powerful library serving an entire community in short, an intellectual public utility."<sup>1</sup>

Research in time-sharing started in the early 1960s around the country at different research centers. Some examples were CTSS (Compatible Time-sharing System) at MIT, DTSS (Dartmouth Time-sharing System) at Dartmouth, a system at BBN, Project GENIE at the University of California at Berkeley, and so on. J. C. R. Licklider, the founding director of ARPA's Information Processing Techniques Office (IPTO) thought of time-sharing as interactive computing. Interactive computing meant the user could communicate and respond to the computer's responses in a way that batch processing did not allow.

Licklider was one of the first users of the new time-sharing systems, and took the time to play around with them. Examining the uses of this new way of communicating with the computer enabled Licklider to think about the future possibilities. This was helpful because Licklider went on to establish the priorities and direction for ARPA's IPTO research monies. Many of the interviewees in a series of interviews conducted by the Charles Babbage Institute (CBI) said that ARPA's money was given in those days under Licklider's guidance to seed research which would be helpful to society in general and only secondarily helpful to the military.

Both Robert Taylor and Larry Roberts, future successors of Licklider as director of IPTO, pinpoint Licklider as the originator of the vision which set ARPA's priorities and goals and basically drove ARPA to help develop the concept and practice of networking computers.

In one of the CBI interviews, Roberts said: "what I concluded was that we had to do something about communications, and that really, the idea of the galactic network that Lick talked about, probably more than anybody, was something that we had to start seriously thinking about. So in a way networking grew out of Lick's talking about that, although Lick himself could not make anything happen because it was too early when he talked about it. But he did convince me it was important."<sup>2</sup>

Taylor, also in a CBI conducted interview, pointed out the importance of Licklider's vision to future network development, "I don't think anyone who's been in that DARPA position since [Licklider] has had the vision that Licklider had. His being at that place at that time is a testament to the tenuousness of it all. It was really a fortunate circumstance. I think most of the significant advances in computer technology, especially in the systems part of computer science were simply extrapolations of Licklider's vision. They were not really new visions of their own. So he's really the father of it all."<sup>3</sup>

Taylor also described how research in time-sharing led to surprising results. He succeeded Licklider as Director of the IPTO at ARPA. A phrase that J. C. R. Licklider frequently used to express his vision was "an Intergalactic Network." Taylor explains that Licklider used this phrase

to describe the potential community that he realized would emerge from the interconnection of the local communities of Net users that develop from time-sharing. At first, Taylor notes ARPA supported research had as its goal achieving compatibility and resource sharing across different computer systems. However, he explains, “They were just talking about a network where they could have a compatibility across these systems, and at least do some load sharing, and some program sharing, data sharing that sort of thing. Whereas, the thing that struck me about the time-sharing experience was that before there was a time-sharing system, let’s say at MIT, then there were a lot of individual people who didn’t know each other who were interested in computing in one way or another, and who were doing whatever they could, however they could. As soon as the time-sharing system became usable, these people began to know one another, share a lot of information, and ask of one another, ‘How do I use this? Where do I find that?’ It was really phenomenal to see this computer become a medium that stimulated the formation of a human community. And so, here ARPA had a number of sites by this time, each of which had its own sense of community and was digitally isolated from the other one. I saw a phrase in the Licklider memo. The phrase was in a totally different context something that he referred to as an ‘intergalactic network.’ I asked him about this...in fact I said, ‘Did you have a networking of the ARPANET sort in mind when you used that phrase?’ He said, ‘No, I was thinking about a single time-sharing system that was intergalactic’”<sup>4</sup>

As Taylor explains, the users of the time-sharing systems would form, usually unexpectedly, a new community. People were connected to others who were also interested in these new computing systems.

The vision driving ARPA inspired bright researchers working on computer related topics. Roberts explains that Licklider’s work (and that of the IPTO’s directors after him) educated people who were to become the future leaders in the computer industry. Roberts describes the impact that Licklider and his vision made on ARPA and future IPTO directors: “Well, I think that the one influence is the production of people in the computer field that are trained, and knowledgeable, and capable, and that form the basis for the progress the United States has made in the computer field. That production of people started with Lick, when he started the IPTO program and started the big university programs. It was really due to Lick, in large part, because I think it was that early set of activities that I continued with that produced the most people with the big university contracts. That produced a base for them to expand their whole department, and produced excitement in the university.”<sup>5</sup>

Roberts describes how ARPA supported university research had a significant impact on the computer industry as well. “So it was clear that that was a big impact on the universities and therefore, in the industry. You can almost track all those people and see what effect that has had. The people from those projects are in large part the leaders throughout the industry”<sup>6</sup>

Licklider’s vision was of an “Intergalactic Network,” a time-sharing utility that would serve the entire galaxy. This early vision of time-sharing spawned the idea of interconnecting different time-sharing systems by networking them together. This network would allow those on geographically separated time-sharing systems to share data, programs, research, and later other ideas and anything that could be typed out. In the article, “The Computer as a Communications Device”, Licklider and Taylor predicted the creation of a global computer network. They wrote: “We have seen the beginnings of communication through a computer – communication among people at consoles located in the same room or on the same university campus or even at distantly separated

laboratories of the same research and development organization. This kind of communication – through a single multiaccess computer with the aid of telephone lines – is beginning to foster cooperation and promote coherence more effectively than do present arrangements for sharing computer programs by exchanging magnetic tape by messenger or mail.”<sup>7</sup>

They point out how the interconnection of computers leads to a much broader class of connections than might have been expected. A new form of community is generated. “The collection of people, hardware, and software the multiaccess computer together with its local community of users will become a node in a geographically distributed computer network. Let us assume for a moment that such a network has been formed. Through the network of message processors, therefore, all the large computers can communicate with one another. And through them, all the members of the super community can communicate with other people, with programs, with data, or with a selected combinations of those resources.”<sup>8</sup>

Licklider and Taylor consider more than just hardware and software when they write about the new social dynamics that the connections of dispersed computers and people will create. They explain: “[These communities] will be communities not of common location, but of common interest. In each field, the overall community of interest will be large enough to support a comprehensive system of field-oriented programs and data.”<sup>9</sup>

In exploring this community of common affinity, they describe the main advantages that come from connecting to and being part of these new computer facilitated communities. Life will be enriched for those people who can communicate on-line with others who have similar goals and interests, as they won’t be limited by geography. Communication will be more productive and thus more enjoyable. And the kind of programs that those on-line will have access to will be customized to one’s interests and abilities, and thus more satisfying. And they describe the advantages to society that the increased opportunities and resources made possible by the Net can provide for everyone.<sup>10</sup>

Since the advantages that computer networks make possible for society will only happen if these advantages are available to all who want to make use of them, Licklider and Taylor realize there is a crucial challenge put on the agenda of our times by the development of the Net. They conclude their article with a prophetic question: “Will ‘to be on line’ be a privilege or a right?”<sup>11</sup> They argue that it must be a right. Otherwise, instead of providing all the many benefits it makes possible, it will only increase the inequities of intellectual opportunity that currently exist.

The challenge they raise is one of access. The authors point out that the positive effects of computer networking will only come about if the networks are made easy to use and available to all. They argue that access should be made available because of the global benefits that would ensue. They conclude by describing how humankind can benefit immeasurably from the educational opportunities the Net makes possible, “if the network idea should prove to do for education what a few have envisioned in hope...surely the boon to humankind would be beyond measure.”<sup>12</sup>

Licklider and Taylor raise the important point that access should be made available to all who want to use the computer networks. Therefore it is important to ask if the National Information Infrastructure is being designed with the principle of equality of access. The vision of the interconnection and interaction of diverse communities guided the creation of the original ARPANET. In the design of the expansion of the Net, it is important to keep the original vision in mind to consider if the vision was correct, or if it was just important in the initial development of networking technologies and techniques. However, very little emphasis has been placed on either

the study of Licklider's vision or the role and advantages of the Net up to this point. In addition, the public has not been allowed to play a role in the planning process for the new initiatives which the U.S. government is currently undertaking. This is a plea to you to demand more of a part in the development of the future of the Net.

---

Notes for Chapter 5

1. "Time-sharing on Computers", in *Information*, A Scientific American Book, San Francisco, 1966, p. 76.
  2. Lawrence G. Roberts, Interview by Arthur L. Norberg, 4 April 1989, San Mateo, California, Charles Babbage Institute, The Center for the History of Information Processing, University of Minnesota, Minneapolis, Minnesota, p. 29.
  3. Robert W. Taylor, Interview by William Aspray, 28 February 1989, Palo Alto, California, Charles Babbage Institute, The Center for the History of Information Processing, University of Minnesota. Minneapolis, Minnesota, p. 8.
  4. *Ibid.*, p. 24.
  5. Lawrence G. Roberts Interview, p. 29.
  6. *Ibid.*, p. 30.
  7. "The Computer as a Communication Device," in *In Memoriam: J. C. R. Licklider: 1915-1990*, p. 28.
  8. *Ibid.*, p. 32.
  9. *Ibid.*, p. 38.
  10. *Ibid.*, p. 40.
  11. *Ibid.*, p. 40.
  12. *Ibid.*, p. 40.
-

# Chapter 6

## Cybernetics, Time-sharing, Human-computer Symbiosis And On-line Communities

### Creating a Super-community of On-line Communities

by Ronda Hauben  
rh120@columbia.edu

#### Part I - Foundations of the Cybernetic Revolution

In 1961, MIT was to celebrate its centennial anniversary. Martin Greenberger, who had joined the MIT faculty in 1958, describes how a call went out for interesting ways to celebrate. “I proposed a series of lectures,” he recalled, “on the computer and the future. We threw open the hatches and got together the best people we could assemble – whatever their fields. We asked these thinkers to project ahead and help us understand what was in store.”<sup>1</sup>

Charles Percy Snow, a British writer was invited to be the keynote speaker. His talk “Scientists and Decision Making,” discussed the need for democratic and broad based participation in the decisions of society. “We happen to be living at a time of a major scientific revolution,” he observed, “probably more important in its consequences, than the first Industrial Revolution, a revolution which we shall see in full force in the very near future.”<sup>2</sup>

He and the other speakers expressed their concern that the challenges represented by the computer be understood and treated seriously. They felt that there would need to be government decisions regarding the development and application of the computer. They cautioned that these decisions be entrusted to people who understood the problems they were dealing with. Also, they were concerned that the smaller the number of people involved in important social decisions, the more likely it would be that serious errors of judgment would be made. They urged that it was necessary to open up the decision making process to as broad a set of people as possible.

Present at this gathering were several of the pioneers who had helped to set the foundation for the developing cybernetic revolution. What was the revolution they were describing?

John Pierce, a pioneer in electronics research at Bell Labs, was one of the speakers at the MIT Centennial Conference. In an article published several years earlier in *Scientific American*, Pierce described the foundation of the cybernetic revolution that was then unfolding.<sup>3</sup> Pierce noted the intellectual ferment that accompanied two publications in 1948. One was “The Mathematical Theory of Communication” by Claude Shannon, published in July and October 1948 in the Bell Labs Journal. The other was the publication of Norbert Wiener’s book *Cybernetics: Control and Communication in the Animal and the Machine*.

Summing up Shannon’s contribution, Pierce noted that Shannon had changed communication theory from guess work to science. Shannon, Pierce wrote, “has made it possible for communication engineers to distinguish between what is possible and what is not possible. Communication theory has disposed of unworkable inventions that are akin to perpetual motion machines. It has directed the attention of engineers to real and soluble problems. It has given them a quantitative measure of the effectiveness of their system.”<sup>4</sup>

In the 1930's, the mathematician and computer pioneer, Alan Turing had determined that it

was possible to design a universal or general purpose computer. Such a computer would be able to solve any calculation that could be solved by a machine, provided the computer had a program describing the calculation. Building on Turing's contribution, Shannon had demonstrated how Boolean algebra and logic could be used in the analysis and synthesis of switching and computer circuits.

Another founder of the Cybernetics Revolution was Norbert Wiener. Pierce recalled the important intellectual catalyst that Wiener's book provided when it appeared in 1948. Wiener was interested in the means by which feedback could be communicated to help correct the problems that develop in an organism. Describing the contribution Wiener's work made in defining the need for feedback, Pierce gives the example of a community "where the Lords of Things as They Are protect themselves from hunger by wealth, from public opinion by privacy and anonymity, from private criticism by the laws of libel and the possession of the means of communication." It is in such a society, he explains, that "ruthlessness can reach its most sublime levels." And he points out that the creation of such a society requires "the control of the means of communication" as "the most effective and important element."<sup>5</sup> Such a community, he observed, is very unstable.

Wiener, in an interview in 1959, explained why such a community is unstable. Describing the importance of accurate information and feedback, Wiener referred to someone driving a car, but "instead of seeing where you are going, somebody puts a picture in front of you. Clearly, it won't be very long before you hit the curb. This is true in other spheres. Facing the contingencies of life depends on adequate and true information. The more that information is conditioned by the people who are doing the controlling, the less they will be able to meet emergencies. In the long run, such a system of misinformation can only lead to catastrophe," warned Wiener.<sup>6</sup>

In Cybernetics, Wiener defined three central concepts which he maintains were crucial in any organism or system. They are communication, control and feedback. Wiener coined the term "cybernetics" to designate the important role that feedback plays in a communication system. He took the word from the Greek term "kybernetes" meaning "governor" or "steersman."<sup>7</sup> Wiener believed that the digital computer had raised the question of the relationship between the human and the machine, and that it was necessary to explore that relationship in a scientific manner. He wrote that what "functions should properly be assigned to these two agencies" is the crucial question for our times.<sup>8</sup>

Important to Wiener's vision was the understanding that the more complex the machine, like the developing digital computer, the more, not less, direction and intelligence were required on the part of its human partner. Wiener often pointed to the literal way in which the computer interpreted the data provided to it. He explained the necessity for increased human guidance and forethought when directing computers: "Here I must enter a protest against the popular understanding of computing machines and similar quasimechanical aids. Many people suppose that they are replacements for intelligence and have cut down the need for original thought. . . . This is not the case. If simple devices need simple thought to get the most out of them, complicated devices need a vastly reinforced level of thought. . . . Moreover this work cannot be put off until the machines have already processed their data. It is very rare, and to say the least, by no means normal, that data that has been thoughtlessly selected can be organized by an after thought so as to produce significant results."<sup>9</sup>

In the introduction to Cybernetics, Wiener described some of the important influences on his development as a scientist and on his thinking in the field of cybernetics. He told of how in the

1930's, he was invited to attend a series of private seminars on the scientific method held by Dr. Arturo Rosenblueth of the Harvard Medical School in Cambridge, Massachusetts. He and Dr. Rosenblueth had a common interest in understanding the scientific method and both believed that science had to be a collaborative endeavor.<sup>10</sup> Scientists involved in a variety of fields of study were invited to the seminars to encourage an interdisciplinary approach to the problems of communication in machine and animals that those in the seminars explored. Describing the methodology of the seminars, Wiener writes: "In those days, Dr. Rosenblueth...conducted a monthly series of discussion meetings on scientific method. The participants were mostly young scientists at the Harvard Medical School, and we would gather for dinner about a round table in Vanderbilt Hall.... After the meal, somebody – either one of our group or an invited guest – would read a paper on some scientific topic, generally one in which questions of methodology were the first consideration, or at least a leading consideration. The speaker had to run the gauntlet of an acute criticism, good-natured but unsparring.... Among the former habitues of these meetings there is more than one of us who feels that they were an important and permanent contribution to our scientific unfolding."<sup>11</sup>

Wiener was a member of this group until the onset of WWII ended the seminars. After the War was over, Wiener began a set of seminars near MIT modeled on his earlier experience in the seminars with Dr. Rosenblueth. The post war seminars that Wiener convened were to have an important influence on the work of several of the pioneers of the upcoming computer networking revolution.

Jerome Wiesner, another MIT computing pioneer, who later became a Science Advisor to President Kennedy, described the role Wiener's seminars played in helping to develop the interdisciplinary tradition of research at MIT's Research Laboratory for Electronics (known as the RLE). Wiener's ideas about communication and feedback in man and machine, along with Shannon's work in information theory "spawned a new vision of research for everyone interested in communications, including neurophysiology, speech, and linguistics investigation," wrote Wiesner, "The work was both theoretical and experimental as well as basic and applied. It led to exciting new ideas and to their implementation in practice which "remains a hallmark of the present-day RLE."<sup>12</sup>

Wiesner provides the following account of the seminars that Wiener set up after WWII: "In the winter of 1947, Wiener began to speak about holding a seminar that would bring together the scientists and engineers who were doing work on what he called communications. He was launching his vision of cybernetics in which he regarded signals in any medium, living or artificial, as the same; dependent on their structure and obeying a set of universal laws set out by Shannon. In the spring of 1948, Wiener convened the first of the weekly meetings that was to continue for several years.... The first meeting reminded me of the tower of Babel, as engineers, psychologists, philosophers, acousticians, doctors, mathematicians, neurophysiologists, philosophers, and other interested people tried to have their say. After the first meeting, one of us would take the lead each time, giving a brief summary of his research, usually accompanied by a running discussion. As time went on, we came to understand each other's lingo and to understand, and even believe, in Wiener's view of the universal role of communications in the universe. For most of us, these dinners were a seminal experience which introduced us to both a world of new ideas and new friends, many of whom became collaborators in later years."<sup>13</sup>



## Part II - Interactive Computing, Time-sharing and Human-Computer Symbiosis

Wiener's stress on interdisciplinary and practical work in the field of communications helped to set the foundation for the upcoming developments in digital computers. By the mid 1950's, several members of the MIT community had been introduced to a new form of computing – interactive computing – in their work on the Whirlwind Computer. Whirlwind research began at MIT in 1947, providing those involved with important practical experience in digital computing. Whirlwind came on line around 1950 and was used until 1957 when the MIT Computation Center began using another vacuum tube computer, the IBM 704.<sup>14</sup> Only when the Computation Center computer was upgraded to the first transistorized computer in the IBM family, the IBM 7090, from vacuum tube computers, did time-sharing become possible.<sup>15</sup>

IBM, which was a main provider of computers during this period, promoted batch processing and saw it as the form of computing for the future. Researchers at MIT, however, had a different vision. Some had worked on the Whirlwind Computer and had experienced a form of interactive computing that made it possible to use the computer directly, rather than having to submit punch cards to a central computer center and await the results.<sup>16</sup> The experience of real time activity at the computer had been a significant advance over the frustration of awaiting the results of one's program under the batch processing system.

Computer resources during this period were very expensive. In general, the cost prohibited a single person from using a computer in real time. A few farsighted researchers, however, had the idea of a time-sharing system which would take advantage of the speed of the computer to allow several users to work with the computer at the same time. The computer scheduled their different work in a way that gave the illusion that the computer was being used by each independently. In June, 1959, Christopher Strachey, a British researcher, presented a talk at the International Conference on Info Processing, UNESCO, proposing time-sharing.<sup>17</sup> Also, in 1959, John McCarthy, an MIT faculty member, wrote a memo describing a new form of computing that time-sharing would make possible and proposing that MIT begin to plan to implement this form of computing once the IBM 7090, the new transistorized computer that they were expecting from IBM to replace the IBM 704, arrived. McCarthy advocated developing a "general-purpose system where you could program in any language you wanted."<sup>18</sup> In his memorandum to MIT Professor P.M. Morse in January 1959, McCarthy writes: "This memorandum is based on the assumption that MIT will be given a transistorized IBM 709 about July 1960. I want to propose an operating system for it that will substantially reduce the time required to get a problem solved on the machine.... The proposal requires a complete revision in the way the machine is used.... I think the proposal points to the way all computers will be operated in the future, and we have a chance to pioneer a big step forward in the way computers are used."<sup>19</sup>

At the same time as McCarthy was proposing a new form of computing, – i.e. time-sharing and interactive computing – another computer pioneer, J. C. R. Licklider, who would play an important role in the developing computer revolution, was working on a paper exploring the concept of human-computer interaction that Norbert Wiener had stressed was so crucial.

Licklider had done his graduate degree in psychology and after WWII, did research at Harvard and worked as a lecturer. He attended the postwar Wiener circles. "At that time," he explained in an interview, "Norbert Wiener ran a circle that was very attractive to people all over

Cambridge, and Tuesday nights I went to that. I got acquainted with a lot of people at MIT.”<sup>20</sup> He describes how another important influence on his work was the Summer Projects at MIT that he attended, starting in 1951. Beginning in the summer of 1952, an interdisciplinary series of summer projects were carried on at MIT for several years which Licklider found “exhilarating.” He remembered how “they brought together all these people – physicians, mathematicians. You would go one day and there would be John von Neumann, and the next day there would be Jay Forrester having the diagram of a core memory in his pocket and stuff – it was fantastically exciting.”<sup>21</sup>

Licklider became involved with MIT and Lincoln Laboratory and “computers and radar sets and communications.” As their token psychologist, he was the only psychologist in this interdisciplinary group of physicists, mathematicians and engineers. “So it was a fantastic opportunity,” he noted. The lab he worked at was run by the RLE and he described how it “gave me a kind of access to the most marvelous electronics there was.”<sup>22</sup>

By 1958-9, Licklider was working with the engineering company, Bolt, Beranek and Newman doing acoustical research. There he had access to digital computers, first a Royal McBee LGP-30, and then one of the earliest DEC PDP-1 computers. Licklider learned how to program on the LGP-30 and when the PDP-1 arrived, one of the earliest time-sharing systems was created for it. Licklider notes the grand time he had exploring what it made possible: “Well, it turned out that these guys at MIT and BBN. We’d all gotten really excited about interactive computing and we had a kind of little religion growing here about how an was going to be totally different from batch processing.”<sup>23</sup>

It was during this period that Licklider carried out an experiment to try to determine how the computer could aid him in his intellectual work. “More significantly,” he explained, “from my point of view, a lot hinged on a little study I had made on how I would spend my time. It showed that almost all my time was spent on algorithmic things that were no fun, but they were all necessary for the few heuristic things that seemed to be important. I had this little picture in my mind of how we were going to get people and computers to really think together.”<sup>24</sup>

Inspired by the Wiener seminars, Licklider tried to set up an interdisciplinary study circle to conduct a study for the Air Force. He explains: “Oh, yes. We had a project with the Air Force Office of Scientific Research to develop the systems concept. Now it’s corny, but then it was an interesting concept. We were trying to figure out what systems meant to the engineer and the scientific world. That involved some meetings in which we brought [together] good thinkers in several fields. We wanted a kind of Wiener circle...we put a lot of hours into trying to do that.”<sup>25</sup>

This study is described in the article “Man-Computer Symbiosis”. Norbert Wiener had proposed that man-computer symbiosis was a subset of the man-computer relationship. Licklider took that observation seriously and wrote an article which was published in March 1960 exploring the meaning and import of man-computer interaction and interdependence.

“Man-computer symbiosis,” he wrote, “is an expected development in cooperative interaction between man and electronic computers. It will involve very close coupling between the human and electronic members of the partnership. The main aims,” he outlined, “are 1) to let computers facilitate formulative thinking as they now facilitate the solution of formulated problems, and 2) to enable man and computers to cooperate in making decisions and controlling complex situations without inflexible dependence on predetermined programs.”<sup>26</sup>

The article became an important formulation of a vision of computing for the developing

computer revolution in time-sharing and networking. Licklider did not promote the computer as a replacement for humans nor see humans as servants to computers. Instead he proposed research exploring the role of humans and machines. His goal was to enhance the symbiotic relationship between the human and computer partners needed to aid intellectual activity.

### **Part III - CTSS and Project MAC**

One of those who was to play an important role in implementing the vision of human-computer symbiosis was Robert Fano. Fano worked at RLE after doing his Ph.D. at MIT in June 1947. In his introduction to his book "Transmission of Information", he described his early contact with Norbert Wiener and Claude Shannon.<sup>27</sup> He explained how he studied the theoretical questions raised by Wiener and Shannon and did research to explore the theories they had pioneered.

In 1960, Fano was a senior faculty member at MIT. Gordon Brown, then Dean of the Engineering School of MIT, arranged for several faculty members to take a course in computing taught by Fernando Corbato and John McCarthy. Fano, remembering his excitement in learning how to program during this course, recalled, "I wrote a program that worked."<sup>28</sup>

Gordon Brown, according to Fano, understood that the computer was going to be very important and encouraged his senior faculty to become familiar with it. In 1960, the MIT administration appointed a committee to make recommendations about the future needs of MIT regarding computers. Fano was one of the faculty members appointed to the committee. This committee created a technical committee made up of Fernando Corbato, John McCarthy, Marvin Minsky, Doug Ross, and Jack Dennis, with Herb Teager acting as Chair. This committee became known as The Long Term Computation Study Group.

It was during this period that the celebration of the MIT centennial was being planned. Eight talks were scheduled. After one of the speakers cancelled at the last minute, John McCarthy, who had been working on the long range computer study, was invited to speak. In his talk, McCarthy described the rationale behind time-sharing and the important vision for the future of computing that it represented. Other participants at the conference included Norbert Wiener, Claude Shannon, John Kemeny, Robert Fano, Alan Perlis, and J. C. R. Licklider.<sup>29</sup> In the course of the conference, Wiener explained that "a computing machine is a general-purpose device that can be programmed to do very specific jobs." But, Wiener warned, if you fail to give a necessary instruction to a computer, "you cannot expect the machine itself to think of this restriction."<sup>30</sup> Wiener explained that humans had to oversee the computer. "An unsafe act thus," Wiener cautioned, "may not show its danger until it is too late to do anything about it."<sup>31</sup>

In his comments, Licklider described how a human being "must not so clutter his mind with codes and formats that he cannot think about his substantive problem."<sup>32</sup> Licklider described his vision of the future of the computer: "In due course it will be part of the formulation of problems; part of real-time thinking, problem solving, doing of research, conducting of experiments, getting into the literature and finding references.... And it will mediate and facilitate communication among human beings."<sup>33</sup>

He expressed his hope that the computer "through its contribution to formulative thinking... will help us understand the structure of ideas, the nature of intellectual processes."<sup>34</sup> And he proposed that the "most important present function of the digital computer in the university should be to catalyze the development of computer science."<sup>35</sup>

Another participant at the conference, the linguist Y. Bar-Hillel, pointed out no one at the conference knew what was going to happen in the long term future, or even in the short term. Because of this uncertainty, it was important to decide what type of future it would be worthwhile to encourage. There were two paths to choose from, and he posed the question as to which path should be taken. “Do we want computers that will compete with human beings and achieve intelligent behavior autonomously, or do we want what has been called man-machine symbiosis?”<sup>36</sup>

“I think computer people have the obligation to decide which of the two aims they are going to adopt,” he proposed. Arguing that the human brain was more developed than it would be possible to make a machine brain at the current stage of technological development, he recommended that the best path was that of man-machine symbiosis. “I admit that these two aims do not definitely exclude each other,” he acknowledged, “but there has been an enormous waste during the last few years in trying to achieve what I regard as the wrong aim at this stage, namely, computers that will autonomously work as well as the human brain with its billion years of evolution.”

Fano, taking a sabbatical in the Summer of 1961, had gone to work at Lincoln Labs because he hoped to learn more about digital computers there. “You know,” he explained, “we used to talk about components such as modulators and detectors and all the gadgets that went into communication systems. That’s the past. We have to talk about functions now, because with a computer you can implement any function you want.”<sup>37</sup> He proposed one had to begin to think about communication in the general purpose way that the digital computer was making possible.

In the meantime, the Long Term Computation Study Group published its reports. There were two proposals for how to proceed. One, from Herbert Teager, who had been Chairman of the Committee, and a second Report from the rest of the committee. Fernando Corbato, a member of the committee and then Associate Director of the MIT Computing Center set out to implement an “interim” solution to the kind of computing the majority report proposed. Corbato describes the subsequent events, “I started up with just a couple of my staff people Marjorie Daggett...and Bob Daley. We hammered out a very primitive prototype. We started thinking about it in Spring of 1961. I remember that by the summer of 1961 we were in the heat of trying to work out the intricacies of the interrupts.”<sup>38</sup>

He explains how he and the other programmers were acting on the vision that had been developed by the majority of the Long Term Study Group Committee. “I sketched out what we would try to do,” he explains, “and Marjorie, Daley and I worked out the hairy details of trying to cope with this kind of poor hardware. By November, 1961,” he notes, “we were able to demonstrate a really crude prototype of the system.”<sup>39</sup>

They gave a seminar and demonstration with their prototype time-sharing system in November 1961. “That’s the date that’s branded in my mind,” Corbato notes. “It was only a four-Flexowriter system. People were pleased that there were finally examples surfacing from [the work]. They did not view [it] as an answer to anybody’s problem. We made the [first] demo in November 1961 on an [IBM] 709,” he recalls. “The switch to the [IBM] 7090 occurred in the spring of 1962 at the Computation Center.”<sup>40</sup>

Corbato describes how CTSS (Compatible Time-Sharing System), as the time-sharing system he was working on was called, couldn’t go into operation until the transistorized IBM 7090 hardware had arrived and could be used in early spring of 1962.<sup>41</sup> Only then could they begin to deal with the real problems to make a working system.

Corbato gave a talk at a Conference about CTSS in May, 1962, but they still didn't have a working system running. By October, 1962, however, J. C. R. Licklider had accepted a position with ARPA under the U.S. Department of Defense on his condition that he would be allowed to implement the vision of interactive computing and time-sharing.

In November, 1962, Licklider and Fano both attended an unclassified meeting held for the Air Force in Hot Springs, Virginia. Fano had been invited to chair a session on Communications, and he and Licklider both attended some of the sessions on command and control. On the way back from the conference, on the train returning to Washington D.C., several people from the meeting were in the same car. They all chatted about what had happened and moved from seat to seat to talk to different people. "And I did spend quite a bit of time with Lick," Fano recalled, "and I understood better what he had in mind."<sup>42</sup>

Fano spent Thanksgiving Day 1962 thinking over the discussion he had had with Licklider. The day after Thanksgiving he had a meeting set up with the Provost at MIT, Charlie Townes. When he told the Provost what he had been thinking, the Provost told Fano, "Go ahead." Fano wrote out his thoughts in a 2 page memorandum that he distributed broadly around MIT. In the proposal, he put forward three goals: 1) time-sharing 2) a community using it and 3) education, which meant supporting research projects.

The following Tuesday, Fano met with Jay [Julius A] Stratton, then President of MIT. Fano was surprised that Stratton asked him which building he would use for the project, encouraging him to begin to implement his proposal.

In reviewing the period, Corbato described how Licklider went to ARPA "as a 'Johnny Appleseed' with a mission" and that was more than his superiors had expected. They tolerated it, Corbato observed, but Licklider was "the one who was driving it rather than them."<sup>43</sup> Lick added that while his superiors called for Command and Control, he made clear he was going to be involved with "interactive computing."<sup>44</sup> "I just wanted to make it clear," Lick noted, "that I wasn't going to be running battle planning missions or something. I was going to be dealing with the engineering substratum that [would] make it possible to do that stuff [command and control]."

Fano developed a funding proposal for Project MAC. It was submitted to ARPA. The contract was signed on July 1, 1963, the day the 1963 summer study began at MIT to demonstrate and create enthusiasm for time-sharing and interactive computing. "Time-sharing," Martin Greenberger recalled, "on the Computation Center machine was available on the opening day of the summer study project."<sup>45</sup>

When asked how he felt when he learned that there would be funding to develop CTSS as part of Project MAC, Corbato recalled, "Well, it was a cooperative thing. Nobody had license to run wild – but you had license to try to make something happen."<sup>46</sup> Corbato clarified, "I wasn't trying to start a company or anything like that; my goal was to exhibit it."

By mid October a second time-sharing computer was available for Project MAC. And it was operating within a week.

Reviewing the reasons for the success of Project MAC, Greenberger explained, "CTSS was an open system. It challenged the user to design his own subsystem, no matter what discipline he came from, no matter what his research interests."<sup>47</sup>

Fano significantly pointed out that one of the goals of Project MAC had not been achieved. This goal identified an important technical and social need that would inspire future networking

developments. The ever developing and changing computer hardware and software posed the challenge of providing a support network for users, both locally and remotely. “One of our goals,” Fano explained, “was to make the computer truly accessible to people wherever they were. We did not succeed. For people who lived in the community that used the system, it was fine. In any system like that, you keep learning things, you keep using new things, and so you keep having troubles. If you can go next door and say, ‘Hey, I was doing this and something strange happened, do you know what I did wrong?’ usually somebody in your neighborhood will be able to help you. If instead you are far away, you are stuck.... We tried to develop some way of helping remote users.... Well, we never did. So in fact, we failed to make the computer truly accessible regardless of the location of the user.”<sup>48</sup>

Despite the problems, Greenberger observed, “I think one of the greatest successes was that CTSS gave so many people, with such widely different backgrounds, a system and experience that they would not have gotten any other way at that point.” Fano explained the importance of developing time-sharing was not just in developing something technical. Rather, he noted, “I am really talking about the interaction of users in the sharing. That’s important,” he emphasized, “I feel that systems that do this as easily as time-sharing systems do not exist.”<sup>49</sup> Remembering how Project MAC created an on-line community, Fano recalled, “friendships being born out of using somebody else’s program, people communicating through the system and then meeting by accident and say ‘Oh, that’s you.’ All sorts of things. It was a non-reproducible community phenomenon,” he concluded.<sup>50</sup>

Offering his summary of the achievements, Corbato explained: “Two aspects strike me as being important. One is the kind of open system quality, which allowed everyone to make the system kind of their thing rather than what somebody else imposed on them.... So people were tailoring it to mesh with their interests. And the other thing is, I think, we deliberately kept the system model relatively unsophisticated (maybe that’s the wrong word – uncomplicated), so we could explain it easily.”<sup>51</sup>

The achievements of Project MAC and the other time-sharing systems built as a result of Licklider’s tenure at ARPA provided the basis for the vision that would guide the development of the ARPANET.<sup>52</sup> In the paper, “The Computer as a Communication Device,” Licklider and Robert Taylor predicted, “In a few years, men will be able to communicate more effectively through a machine than face to face.... We believe that we are entering into a technological age, in which we will be able to interact with the richness of living information – not merely in the passive way that we have become accustomed to using books and libraries, but as active participants in an ongoing process, bringing something to it through our interaction with it, and not simply receiving something from it by our connection to it.”<sup>53</sup>

While they acknowledged that technical uses like the switching function were important in the transfer of information, such uses were not the aspect they were interested in. Instead they proposed that there was a power and responsiveness that online interaction with a computer made possible that would significantly affect the communication possible between humans using the computer.

Though they were familiar with commercial time-sharing facilities that called themselves “multiaccess,” they explained that these had not succeeded in creating the kind of multiaccess computer communities that the academic and research time-sharing systems spawned.

They described these time-sharing communities, of which Project MAC was one of the early examples, as “socio-technical pioneers...out ahead of the rest of the computer world.” They attributed this to the fact that some of the members of these online communities were computer scientists and engineers who understood both the concept of human computer interaction and the technology of interactive, multiaccess systems. Among the members of these online communities were creative people in different fields and disciplines who recognized the potential value of these multiaccess communities to their work. Thirdly, the online communities had access to large multiaccess computers and knew how to use them. Fourthly, they maintained that the efforts of those online had a regenerative effect.<sup>54</sup>

Elaborating on what they meant by regenerative, they wrote, that in the half dozen time-sharing on-line communities in existence during the 1960s, those doing research and development of computer systems and applications provided mutual support for each other. The product was a growing quantity of resources including programs, data, and technical know how. “But we have seen only the beginning,” they predicted, “There is much more programming and data collection – and much more learning how to cooperate – to be done before the full potential of the concept can be realized.” They go on to caution that these systems could only be developed interactively. And they explain that, “The systems being built must remain flexible and open-ended throughout the process of development, which is evolutionary.”<sup>55</sup>

They also describe how there were systems that were advertising themselves via the same labels as “interactive,” “time-sharing” and “multiaccess.” But these were commercial systems and they found that there were distinct differences between the commercial systems and the academic and research time-sharing ones. The commercial systems did not offer the same “power”, “flexibility” of software resources and the general purposeness that the research and academic time-sharing system at MIT, UCB, Stanford and SDC had made available to over 1000 people for a number of years.<sup>56</sup>

They discussed their vision of the future, predicting that linking up the existing online communities would create a still more powerful and important development – online super-communities made up of the existing communities created by the time-sharing systems. “The hope,” they explained, “is that interconnection will make available to all the communities the programs and data resources of the entire super-community.” They predicted that the future would bring “a mobile network of networks – ever changing in both content and configuration.” And just as Licklider and Taylor realized that a time-sharing system was more than a collection of computers and software, Fano recognized that a time-sharing system was more than just a set of people using common resources; it was also a means of communicating and sharing ideas.<sup>57</sup>

Another time-sharing pioneer, Doug Ross, observed that Project MAC made CTSS available rather than waiting for the ideal technical system to be developed, as others had favored. By producing a prototype and encouraging others to contribute to it, CTSS had a significant impact on others who, therefore, had the ability to build into the system what they needed and to contribute so it would serve their needs. “I always say,” Ross concluded, “you can’t design an interface from just one side.”<sup>58</sup> This quality of putting an open system out and encouraging people to contribute to it to make it what they needed, was building a human centered rather than technology centered system.<sup>59</sup>

Summing up the achievement of the Project MAC pioneers, John A. N. Lee, editor of two special issues of *The IEEE Annals of the History of Computing* which document the development

of time-sharing and Project MAC at MIT, writes: “With the development of computer networking, which almost naturally followed on the development of time-sharing and interactive computing, it is as if the whole world now time shares myriad computers, providing facilities which were beyond the dreams of even the MIT researchers of 1960.... But this is where it started – with the ideas of John McCarthy, the implementation skills of Fernando Corbato, the vision of J. C. R. Licklider, and the organizational skills of Robert Fano.”<sup>60</sup>

## **Part IV - The Implications**

The pioneers of cybernetics and multiaccess computing who gathered at the MIT centennial in the Spring of 1961 to discuss the future of computing proposed that the crucial issue one must determine in trying to solve a problem is how to formulate the question. They expressed concern that the computer would bring great changes into our world and that people who understood the issues involved be part of setting government policy regarding these developments.

The pioneers also observed that there were opposing visions of what the future should be. One road was that of human-computer symbiosis, of a close interaction between the human and the computer so each could function more effectively. “The hope is that, in not too many years,” J. C. R. Licklider wrote, “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.”<sup>61</sup> The other road was that of creating computers that would be able to do the thinking or problem solving without human assistance. Pioneers like Licklider explained that “man-computer symbiosis is probably not the ultimate paradigm for complex technological systems” and that in the future at some point “electronic or chemical ‘machines’ will outdo the human brain in most of the functions we now consider exclusively within its province.” He maintained, however, that, “there will nevertheless be a fairly long interim during which the main intellectual advances will be made by men and computers working together in intimate association.”<sup>62</sup> Though Licklider was willing to concede, “dominance in the distant future of celebration to machines alone,” he recognized the creative and important developments that the partnership between the human and computer would make possible. He predicted that the years of human-computer symbiosis, “should be intellectually the most creative and exciting in the history of mankind.”<sup>63</sup>

In the years following the development of CTSS and Project MAC and the linking of different time-sharing systems to create a super-community of on-line communities which became known as the ARPANET, the firm foundation set by CTSS and Project MAC and the helpful vision and direction set by Licklider and Fano and other pioneers of the period, gave birth to the sprawling and impressive networking communities that today we call the Internet and Usenet.

The pioneers of time-sharing and interactive computing provided a vision of human-computer symbiosis as an intellectual advance for humans. Online human-computer, and computer facilitated human to human communication was seen as the embodiment of this symbiosis. The vision of the computer pioneers of the 1960's, of human-computer symbiosis, and of creating a multiaccess, interactive, network of networks, or a super community network as they termed it, is the vision that can still fruitfully guide the work to build and extend the global computer network in the U.S. and around the world today.



---

Notes for Chapter 6

1. "The Project MAC Interviews," *IEEE Annals of the History of Computing*, Vol. 14 no. 2, 1992, p. 15. The interviews were conducted on October 18, 1988. They were in two group interviews/recollection exchanges. The interviewers were John A. N. Lee and Robert Rosin. The participants were Fernando J. Corbato, Robert M. Fano, Martin Greenberger, Joseph C. R. Licklider, Douglas T. Ross and Allan L. Scherr.
2. Martin Greenberger, ed, *Management and Computers of Future*, The MIT Press, Cambridge, Massachusetts, 1962, p. 8.
3. John R. Pierce, "Communication," *Scientific American*, Vol. 227 no. 3, September, 1972.
4. *Ibid.*, p. 33.
5. *Ibid.*, p. 41.
6. "Challenge Interview: Norbert Wiener: Man and the Machine", June 1959, in *Collected Works of Norbert Wiener with Commentaries*, Vol. 4, The MIT Press, Cambridge, Massachusetts, 1985, p. 717.
7. Norbert Wiener, *Cybernetics: or Control and Communication in the Animal and the Machine*, The MIT Press, Cambridge, Massachusetts, 1948, pp. 11-12. Wiener wrote, "In choosing this term, we wish to recognize that the first significant paper on feedback mechanisms is an article on governors, which was published by Clerk Maxwell in 1868.... We also wish to refer to the fact that the steering engines of a ship are indeed one of the earliest and best-developed forms of feedback mechanisms."
8. *God and Golem*, The MIT Press, Cambridge, Massachusetts, p. 71.
9. "A Scientist's Dilemma in a Materialist World," by Norbert Wiener, in *Collected Works*, Vol 4, p. 709.
10. Norbert Wiener, *I Am A Mathematician*, The MIT Press, Cambridge, Massachusetts, 1956, p. 171.
11. Norbert Wiener, *Cybernetics*, p. 1.
12. From "The Legacy of Norbert Wiener: A Centennial Symposium," Cambridge, Massachusetts, 1994, p. 19. Licklider, Fano, Minsky and other MIT pioneers refer to the important influence that being part of the RLE had on their subsequent work.
13. *Ibid.*
14. One of the reasons that a computer using vacuum tubes was not appropriate for a time-sharing system, according to Robert Fano, was that the "mean time between failures was seven or nine [hours]." See *IEEE Annals of the History of Computing*, Vol. 14 no. 2, 1992, p. 25.
15. Chronology from *IEEE Annals of the History of Computing*, Vol. 14 no. 1, 1994, p. 18. (Here after, *Annals*)
16. See *Annals*, Vol. 14 no. 1, 1992, p. 38 for a description of the frustrations of batch processing.
17. See Strachey, C. "Time sharing in large fast computers," Proc Int. Conf. on Info Processing, UNESCO, June, 1959, pp. 336-341. See also Frederick P. Brooks Jr., *The Mythical Man-Month, Essays on Software Engineering*, Reading, Massachusetts, 1972, p. 146.

18. See *Annals*, Vol. 14, no. 1, 1992, "John McCarthy's 1959 Memorandum," pp. 20-21. See also J. A. N. Lee "Claims to the Term Time-Sharing", *ibid*, p. 16-17.
19. "John McCarthy's 1959 Memorandum", p. 20.
20. *Annals*, vol. 14, no. 2, 1992, p. 16.
21. *Ibid*.
22. *Ibid*.
23. Interview by William Aspray and Arthur L. Norberg, Tape recording, Cambridge, Massachusetts, 28 October 1988, OH 150, Charles Babbage Institute, University of Minnesota, Minneapolis, Minnesota.
24. J. C. R. Licklider, "Man-Computer Symbiosis", *IRE Transactions on Human Factors in Electronics*, Vol. HFE-1, pp. 4-11, March, 1960. Reprinted in In Memoriam: J. C. R. Licklider 1915-1990, Digital Systems Research Center, Palo Alto, California, August 7, 1990, pp. 1-19.
25. *Ibid*.
26. *Ibid.*, p. 1.
27. Robert Fano, *Transmission of Information*, MIT Press, Cambridge, Massachusetts, p. .
28. "An Interview with ROBERT M. FANO," Conducted by Arthur L. Norberg on 20-21 April 1989, Cambridge, MA, Charles Babbage Institute, Center for the History of Information Processing, University of Minnesota, Minneapolis
29. The book was first published under the title *Management and the Future of the Computer* by The MIT Press, Cambridge, Massachusetts, in 1962 and later in hardback and paperback under the title *Computers and the World of the Future*. It was edited by Martin Greenberger.
30. *Ibid.*, p. 24.
31. *Ibid.*, p. 32.
32. *Ibid.*, pp. 204-5.
33. *Ibid.*, p. 205.
34. *Ibid.*, p. 206.
35. *Ibid.*, p. 207.
36. *Ibid.*, p. 324.
37. *Annals*, Vol. 14 no. 2, 1992, p. 20.
38. *Annals*, Vol. 14 no. 1, 1992, p. 44. Teager's recommendations are described on pages 24-27. Excerpts from the Long Range Computation Study Group's recommendation for a time-sharing systems which resulted in Corbato's work on CTSS are in the same issue on pages 28-30.
39. "What we had done was [that] we had wedged out 5K words of the user address space and inserted a little operating

system that was going to manage the four typewriters. We did not have any disk storage, so we took advantage of the fact that it was a large machine and we had a lot of tape drives. We assigned one tape drive per typewriter.” Ibid., p. 45.

40. Ibid., pp.45-46. Corbato describes how he thought CTSS would be running on the IBM 7090 by the time he was to give a talk on it at the AFIPS Spring Joint Computer Conference in May, 1962. But that they were not able to get it running by the time the paper was presented. Despite his disappointment, the paper is an important historical document. See "An Experimental Time-Sharing System," by Fernando J. Corbato, Marjorie Merwin-Daggett, Robert C. Daley, Proceedings of the American Federation of Information Processing Societies, Spring Joint Computer Conference, May 1-3, 1962, Vol. 21, pp. 335-344.

41. It was called the Compatible Time-Sharing System as it was developed in the Computation Center and so had to be compatible with the batch system running there.

42. *Annals*, Vol. 14 no. 2, pp. 21-22.

43. Ibid., p. 24

44. Ibid.

45. Ibid., p. 26. Fano explained that Licklider wanted interactive computing with time-sharing. He notes that “one was the ‘tool’ the other the ‘goal’. This is where the name MAC came from. There was a goal and there was a tool – the tool that was most appropriate at that time.” He goes on to explain that there had been the vision on the part of people like John McCarthy and later Licklider “of what could come out of it when you started building a computer utility. It didn’t exist then. It didn’t exist until the time of Project MAC because it was just that year that Corby finished the model that really could serve a community. It didn’t exist before.” Ibid., p. 23.

46. Ibid. p. 26.

47. Ibid. Greenberger describes how he designed a subsystem for CTSS where students created a set of commands to simulate stock market, accounting, production scheduling, online modeling, etc. They put these commands together into a system under CTSS that they called OPS, (On-line Programming and Simulation). Ibid., p. 27.

48. Ibid., p. 31.

49. Ibid., p. 35.

50. Ibid.

51. Ibid., p. 33.

52. By Fall, 1967, there were 35 time-sharing systems operational or planned, at research and academic sites, mainly in the U.S., according to the “Time-Sharing System Scorecard.” (“Prolog to the Future”, *Annals*, Vol. 14, no. 2, p. 42-47.) The scorecard also lists 15 commercial time-sharing installations being planned or in existence.

53. “The Computer as a Communication Device,” *IRE Transactions on Human Factors in Electronics*, volume HFE-1, pages 4-11, March 1960, as reprinted in “In Memoriam: J. C. R. Licklider: 1915-1990”, Palo Alto, August 7, 1990, p. 21.

54. Ibid., pp. 30-31.

55. Ibid., p. 31.

56. The time-sharing systems they are describing are listed in the “Time-Sharing System Scorecard” as having been begun in the following years:

MIT [Project MAC at MIT begun in May, 1963],

UCB [Project GENIE at the University of California at Berkeley begun in April, 1965],

Stanford University [Stanford, California begun in August, 1964],

and SDC [Systems Development Corporation in Santa Monica, California, begun in August, 1964].

57. Corbato explains that Robert Fano “correctly saw that a time-sharing system was more than just a set of people using common resources; it was also a means of communicating and sharing ideas.” *Annals*, Vol. 14 no. 1, p. 48.

58. *Ibid.*, p. 51.

59. *Annals*, Vol. 14, no. 2, one of the interviewers, Robert Rosin noted, “You see, if what you’re trying to do is optimize technical resources (physical resources), Herb’s point of view was exactly right. If you try to optimize the use of human resources, then the point of view you were taking was a lot closer to reality.”

60. *Annals*, Vol. 14, no. 1, p. 3-4.

61. “Man Computer Symbiosis,” p. 3. Licklider proposes the role that each partner will play in the symbiotic relationship. The human partner will “set the goals, formulate the hypotheses, determine the criteria, and perform the evaluations.” The computers “will do the routinizable work that must be done to prepare the way for insights and decisions in technical and scientific thinking.” p. 1.

62. *Ibid.*, pp. 2-3.

63. *Ibid.*

---

MAC is an acronym that has had several explanations, including Machine Aided Cognition, MAN and Computer and Minsky against Corbato, according to Peter Elias, in the 25th Anniversary Project MAC Time Line.

---

Thanks to Tom Van Vleck, Alex McKenzie and to Fernando Corbato for pointing out sources that were helpful in doing the research for this paper. Also, thanks to Scott Dorsey who suggested I try to find out about Project MAC. Another source which covers this material in a way that is helpful is “The Evolution of Interactive Computing Through Time-Sharing and Networking” by Judy Elizabeth O’Neill, June 1992. The interviews in the *IEEE Annals of the History of Computing* Special Issues (Vol. 14 no. 1 and no. 2) are an important source of information about the period. They are supplemented by interviews that are available from the Charles Babbage Institute.

---

# Chapter 7

## Behind the Net: The Untold History of the ARPANET and Computer Science

by Michael Hauben

The global Internet's progenitor was the Advanced Research Projects Agency Network (ARPANET) financed and encouraged by the U.S. Department of Defense. This is important to remember, because the support and style of management by ARPA of its contractors was crucial to the success of the ARPANET. As the Internet develops and the struggle over the role it plays unfolds, it will be important to remember how the network developed and the culture with which it was connected. The culture of the Net As a facilitator of communication, is important feature to understand.

The ARPANET Completion Report, published jointly in 1978 by BBN of Cambridge, Massachusetts, and ARPA, concludes by stating: "...it is somewhat fitting to end on the note that the ARPANET program has had a strong and direct feedback into the support and strength of computer science, from which the network itself sprung."<sup>1</sup> In order to understand the wonder that the Internet and various other components of the Net represent, we need to understand why the ARPANET Completion Report ends with the suggestion that the ARPANET is fundamentally connected to and born of computer science rather than of the military.

### Part I: the History of Arpa Leading up to the ARPANET

A climate of scientific research surrounded the entire history of the ARPANET. ARPA was formed to fund basic research, and thus was not oriented toward military products. The formation of this agency was part of the U.S. Government response to the then Soviet Union's launch of Sputnik in 1957.<sup>2</sup> One area of ARPA supported research concerned the question of how to utilize the military's investment in computers to do Command and Control Research (CCR). Dr. J. C. R. Licklider was chosen to head this effort. Licklider came to ARPA from BBN in October 1962.<sup>3</sup> His educational background was a combination of engineering studies and physiological psychology. His multidisciplinary experiences provided Licklider with an unusual perspective uncommon among engineers.

As a result of Licklider's arrival, the Agency's contracts were shifted from independent corporations towards "the best academic computer centers."<sup>4</sup> The then current method of computing was via batch processing. Licklider saw improvements could be made in CCR only from work that would advance the current state of computing technology. He particularly wanted to move forward into the age of interactive computing, and the Defense Department contractors were not moving in that direction. In an interview, Licklider described how at one of the contractors, System Development Corporation (SDC) the computing research being done "was based on batch processing, and while I was interested in a new way of doing things, they [SDC] were studying how to make improvements in the ways things were done already."<sup>5</sup> To reflect the changed direction Licklider was bringing to ARPA supported research, his division of ARPA was renamed Information Processing Techniques Office (IPT or IPTO). The office "developed into a far-reaching basic research program in advanced technology."<sup>6</sup>

The Completion Report states that “Prophetically, Licklider nicknamed the group of computer specialists he gathered the ‘Intergalactic Network’.”<sup>7</sup> Before work on the ARPANET began, the foundation had been established by the creation of the Information Processing Techniques Office of ARPA. Robert Taylor, Licklider’s successor at the IPTO, reflects on how this foundation was based on Licklider’s interest in interconnecting communities. “Lick was among the first to perceive the spirit of community created among the users of the first time-sharing systems...” Taylor continues, “In pointing out the community phenomena created, in part, by the sharing of resources in one timesharing system, Lick made it easy to think about interconnecting the communities, the interconnection of interactive, on-line communities of people,....”<sup>8</sup>

The “spirit of community” was related to Licklider’s interest in having computers help people communicate with other people.<sup>9</sup> Licklider’s vision of an “intergalactic network” connecting people represented an important conceptual shift in computer science. This vision guided the researchers who created the ARPANET. After the ARPANET was functioning, the computer scientists using it realized that assisting human communication was a major fundamental advance that the ARPANET made possible.

As early as 1963, a commonly asked question of the IPTO directors by the ARPA directors about IPTO projects was “Why don’t we rely on the computer industry to do that?”, or occasionally more strongly, “We should not support that effort because ABC (read, “computer industry”) will do it – if it’s worth doing!”<sup>10</sup> This question leads to an important distinction – this ARPA research was different from what the computer industry had in mind to do, or was likely to undertake. Since Licklider’s creation of the IPTO, the work supported by ARPA/IPTO continued his explicit emphasis on communications. The Completion Report explains: “The ARPA theme is that the promise offered by the computer as a communication medium between people, dwarfs into relative insignificance the historical beginnings of the computer as an arithmetic engine.”<sup>11</sup> The Completion Report goes on to differentiate the research ARPA supported from the research done by the computer industry: “The computer industry, in the main, still thinks of the computer as an arithmetic engine. Their heritage is reflected even in current designs of their communication systems. They have an economic and psychological commitment to the arithmetic engine model, and it can die only slowly....”<sup>12</sup> The Completion Report further analyzes this problem by tracing it back to the nation’s universities: “...furthermore, it is a view that is still reinforced by most of the nation’s computer science programs. Even universities, or at least parts of them, are held in the grasp of the arithmetic engine concept....”<sup>13</sup>

ARPA’s IPTO was responsible for the research and development which led to the success of first the ARPANET, and later the Internet. Without the commitment that existed via this support, such a development might never have happened. One of ARPA’s criterion for supporting research was that the research had to be of such a level as to offer an order of magnitude of advance over the current state of development. Such research is never immediately profitable. In society therefore there is the need for organizations which do not pursue profit as their goal, but rather work on furthering the state of the art. Computer networking was developed and spread widely in an environment outside of commercial and profit considerations, an environment that supported such research.

Others have understood the communications promise of computers. For example, in RFC-1336, David Clark, senior research scientist at MIT’s Laboratory for Computer Science, is

quoted, "It is not proper to think of networks as connecting computers. Rather, they connect people using computers to mediate. The great success of the internet is not technical, but in human impact. Electronic mail may not be a wonderful advance in Computer Science, but it is a whole new way for people to communicate. The continued growth of the Internet is a technical challenge to all of us, but we must never lose sight of where we came from, the great change we have worked on the larger computer community, and the great potential we have for future change."<sup>14</sup>

Research predating the ARPANET had been done by Paul Baran, Thomas Marill and others.<sup>15</sup> This led Lawrence Roberts and other IPTO staff to formally introduce the topic of networking computers of differing types (i.e., incompatible hardware and software) together in order to make it possible for ARPA's Principal Investigators (PI) to share resources. The ARPA Principal Investigators meeting was held annually for university and other contractors to summarize results of the previous year and discuss future research. In the Spring of 1967, it was held at the University of Michigan, in Ann Arbor. Networking was one of the topics brought up at this meeting. As a result of discussion at this meeting, it was decided that there had to be agreement on conventions for character and block transmission, error checking and retransmission, and computer and user identification. These specifications became the contents of the inter-host communication's "protocol." Frank Westervelt was chosen to write about this protocol and a communication group was formed to study the questions.<sup>16</sup>

In order to develop a network of varied computers, two main problems had to be solved:

1. To construct a 'subnetwork' consisting of telephone circuits and switching nodes whose reliability, delay characteristics, capacity, and cost would facilitate resource sharing among computers on the network.
2. To understand, design, and implement the protocols and procedures within the operating systems of each connected computer, in order to allow the use of the new subnetwork by the computers in sharing resources.<sup>17</sup>

After one draft and additional work on this communications position paper were completed, a meeting was scheduled in early October 1967 by ARPA at which the protocol paper and specifications for the Interface Message Processor (IMP) were discussed. A subnetwork of IMPs, dedicated mini-computers connected to each of the participant computers, was the method chosen to connect the participants' computers (hosts) to each other via phone lines. This standardized the subnet to which the hosts connected. Now, only the connection of the hosts to the network would depend on vendor type, etc. ARPA had picked 19 possible participants in what was now known as the "ARPA Network."

From the time of the 1967 PI Meeting, various computer scientists who were ARPA contractors were busy thinking about various aspects which would be relevant to the planning and development of the ARPANET. Part of that work was a document outlining a beginning design for the IMP subnetwork. This specification led to a competitive procurement for the design of the IMP subnetwork.

By late 1967 ARPA had given a contract to the Stanford Research Institute (SRI) to write the specifications for the communications network they were developing. In December of 1968, SRI issued a report "A Study of Computer Network Design Parameters." Elmer Shapiro played an important role in the research for this report. Based on this work, Roberts and Barry Wessler of

ARPA wrote the final ARPA version of the IMP specification.<sup>18</sup> This specification was ready to be discussed at the June, 1968 PI meeting.

The Program Plan, “Resource Sharing Computer Networks,” was submitted June 3, 1968 by the IPTO to the ARPA Director, who with unusual speed, approved it on June 21, 1968. It outlined the objectives of the research, and the plan of how the objectives would be fulfilled. The proposed network was impressive as it would prove useful to both the computing research centers which connected to the network and the military. The proposed requirements for the research would provide immediate benefits to the computer centers the network would connect. ARPA’s stated objectives were to experiment with varied interconnections of computers and the sharing of resources in an attempt to improve productivity of computer research. Justification was drawn from technical needs in both the scientific and military environments. The Program Plan developed into a set of specifications. These specifications were connected to a competitive Request for Quotation (RFQ) to find an organization which would design and build the IMP subnetwork.<sup>19</sup>

Following the approval of the Program Plan, 140 potential bidders were mailed the Request for Quotation. After a bidders conference, twelve proposals were received and from them ARPA narrowed the bidders down to four. BBN was the eventual recipient of the contract.<sup>20</sup>

The second technical problem, as defined by the ad hoc Communications Group, still remained to be solved. The set of agreed upon communications settings (known as a protocol), which would allow the hosts to communicate with each other over the subnetwork, had to be developed. This work was left “for host sites to work out among themselves.”<sup>21</sup> This meant that the software necessary to connect the hosts to the IMP subnetwork had to be developed. ARPA assigned this duty to the initially designated ARPANET sites. Each of the first sites had a different type of computer to connect. ARPA trusted that the programmers at each site would be capable of modifying their operating systems in order to connect their systems to the subnetwork. In addition the sites needed to develop the software necessary to utilize the other hosts on the network. By assigning them responsibilities, ARPA made the academic computer science community an active part of the ARPANET development team.<sup>22</sup>

Steve Crocker, one of graduate students involved with the development of the earliest ARPANET protocols, associates the placement of the initial ARPANET sites at research institutions with the fact that the ARPANET was ground-breaking research. He wrote in a message responding to questions on the COM-PRIV mailing list: “During the initial development of the ARPANET, there was simply a limit as to how far ahead anyone could see and manage. The IMPs were placed in cooperative ARPA R&D sites with the hope that these research sites would figure out how to exploit this new communication medium.”<sup>23</sup>

The first sites of the ARPANET were picked to provide either network support services or unique resources. The key services the first four sites provided were:<sup>24</sup>

UCLA - Network Measurement Center

SRI - Network Information Center

UCSB - Culler-Fried interactive mathematics

UTAH - graphics (hidden line removal)

Crocker recounts that the reason for selecting these particular four sites was because they were “existing ARPA computer science research contractors.” This was important because “the research community could be counted on to take some initiative.”<sup>25</sup>



The very first site to receive an IMP was UCLA. Professor Leonard Kleinrock of UCLA was involved with much of the early development of the ARPANET. His work in queuing theory gave him a basis to develop measurement techniques used to monitor the ARPANET's performance. This made it natural to make sure that UCLA received one of the first nodes as it would be important to measure the network's activity from early on. In order for the statistics to have correct data and for analysis purposes – one of the first two or three sites had to be the measurement site. Sure enough, UCLA was assigned to be the Network Measurement Center (NMC).<sup>26</sup>

## **Part II: The Network Working Group**

Once the initial sites were picked, representatives from each site gathered together to start talking about solving the technical problem of getting the hosts to communicate with each other. The ARPANET Completion Report tells us about this beginning: “To provide the hosts with a little impetus to work on the host-to-host problems. ARPA assigned Elmer Shapiro of SRI ‘to make something happen’, a typically vague ARPA assignment. Shapiro called a meeting in the summer of 1968 which was attended by programmers from several of the first hosts to be connected to the network. Individuals who were present have said that it was clear from the meeting at that time, no one had even any clear notions of what the fundamental host-to-host issues might be.”<sup>27</sup>

This group, which came to be known as the Network Working Group (NWG), was exploring new territory. The first meeting took place several months before the first IMP was configured. The group had to think from a blank slate. In Crocker's recollections of the important developments produced by the NWG provided as the introduction to RFC-1000, the reader is reminded that the thinking involved was groundbreaking and thus exciting. Crocker remembers that the first meeting was chaired by Elmer Shapiro of SRI, who initiated the conversation with a list of questions.<sup>28</sup> Also present at this first meeting were Steve Carr from the University of Utah, Stephen Crocker from UCLA, Jeff Rulifson from SRI, and Ron Stoughton from UCSB. These attendees, most of them graduate students, were the programmers described in the ARPANET Completion Report.

According to Crocker, this was a seminal meeting. The attendees could only be theoretical, as none of the lowest levels of communication had been developed yet. They needed a transport layer or low-level communications platform to build upon. BBN would not deliver the first IMP until August 30, 1969. It was important to meet before this date, as the NWG “imagined all sorts of possibilities.”<sup>29</sup> Only once their thought processes started could this working group actually develop anything. These fresh thoughts from fresh minds helped to incubate new ideas. The ARPANET Completion Report properly acknowledges what this early group helped accomplish. “Their early thinking was at a very high level.”<sup>30</sup> A concrete decision of the first meeting was to continue holding meetings similar to the first one. This wound up setting the precedent of holding exchange meetings at each of their sites.

Crocker, describing the problems facing these networking pioneers, writes: “With no specific service definition in place for what the IMPs were providing to the hosts, there wasn't any clear idea of what work the hosts had to do. Only later did we articulate the notion of building a layered set of protocols with general transport services on the bottom and multiple application-specific protocols on the top. More precisely, we understood quite early that we wanted quite a bit of generality, but we didn't have a clear idea how to achieve it. We struggled between a grand design and getting something working quickly.”<sup>31</sup>

The initial protocol development led to DEL (Decode-Encode-Language) and NIL (Network Interchange Language). These languages were more advanced than what was needed or possible at the time. The basic purpose was to form an on-the-fly description that would tell the receiving end how to understand the information that would be sent. This first set of meetings were extremely abstract as neither ARPA nor the universities had conceived of any official charter. However, the lack of a specific charter allowed the group to think broadly and openly.

BBN did provide details about the host-IMP interface specifications from the IMP side. This information gave the group some definite starting points to build from. Soon after BBN provided more information, on Valentine's Day, 1969, members of the NWG, members of BBN and members of the Network Analysis Corporation (NAC) met for the first time. The NAC had been invited because it had been contracted by ARPA to specify the topological design of the ARPANET and to analyze its cost, performance, and reliability characteristics.<sup>32</sup> As all the parties had different priorities in mind, the meeting was a difficult one. BBN was interested in the lowest level of making a reliable connection. The programmers from the host sites were interested in getting the hosts to communicate with each other either via various higher level programs. Even when the crew from BBN did not turn out to be the “experts from the East,” members of the NWG still expected that “a professional crew would show up eventually to take over the problems we were dealing with.”

A step of great importance which began the open documentation process occurred as a result of a “particularly delightful” meeting that took place a month later in Utah. The participants decided it was time to start recording their meetings in a consistent fashion. What resulted was a set of informal notes titled “Request for Comments,” (RFC). Stephen Crocker writes about their formation: “I remember having great fear that we would offend whomever the official protocol designers were, and I spent a sleepless night composing humble words for our notes. The basic ground rules were that anyone could say anything and that nothing was official. And to emphasize the point, I labeled the notes ‘Request for Comments.’ I never dreamed these notes would be distributed through the very medium we were discussing in these notes. Talk about Sorcerer’s Apprentice!”<sup>33</sup>

Crocker replaced Shapiro as the Chairman of the NWG soon after the initial meeting. He describes how they wrestled with the creation of the host-host protocols: “Over the spring and summer of 1969 we grappled with the detailed problems of protocol design. Although we had a vision of the vast potential for inter-computer communication, designing usable protocols was another matter. A custom hardware interface and custom intrusion into the operating system was going to be required for anything we designed, and we anticipated serious difficulty at each of the sites. We looked for existing abstractions to use. It would have been convenient if we could have made the network simply look like a tape drive to each host, but we knew that wouldn’t do.”<sup>34</sup>

The first IMP was delivered to UCLA in late August, 1969. The next was delivered to SRI a month later in October.<sup>35</sup> Once more than one IMP existed, the NWG had to implement a working communications protocol. This first set of pairwise host protocols included remote login for interactive use (telnet), and a way to copy files between remote hosts (FTP). Crocker writes: “In particular, only asymmetric, user-server relationships were supported. In December 1969, we met with Larry Roberts in Utah, [and he] made it abundantly clear that our first step was not big enough, and we went back to the drawing board. Over the next few months we designed a symmetric host-host protocol, and we defined an abstract implementation of the protocol known as the Network Control Program. (‘NCP’ later came to be used as the name for the protocol, but it originally meant

the program within the operating system that managed connections. The protocol itself was known blandly only as the host-host protocol.) Along with the basic host-host protocol, we also envisioned a hierarchy of protocols, with Telnet, FTP and some splinter protocols as the first examples. If we had only consulted the ancient mystics, we would have seen immediately that seven layers were required.”<sup>36</sup>

The Network Working Group went on to develop the protocols necessary to make the network viable. The group swelled in attendance as more and more sites connected to the ARPANET. The group became large enough (around 100 people) that one meeting was held in conjunction with the 1971 Spring Joint Computer Conference in Atlantic City. A major test of the NWG’s work came in October 1971, when a meeting was held at MIT. Crocker continues the story: “[A] major protocol ‘fly-off’ – Representatives from each site were on hand, and everyone tried to log in to everyone else’s site. With the exception of one site that was completely down, the matrix was almost completely filled in, and we had reached a major milestone in connectivity.”<sup>37</sup>

The NCP was creating what was called the “host to host protocol.” Explaining why this was important, the authors of the ARPA draft write: “The problem is to design a host protocol which is sufficiently powerful for the kinds of communication that will occur and yet can be implemented in all of the various different host computer systems. The initial approach taken involved an entity called a “Network Control Program” which would typically reside in the executive of a host, such that processes within a host would communicate with the network through this Network Control Program. The primary function of the NCP is to establish connections, break connections, switch connections, and control flow. A layered approach was taken such that more complex procedures (such as File Transfer Procedures) were built on top of similar procedures in the host Network Control Program.”<sup>38</sup>

As the ARPANET grew, the number of users bypassed the number of developers. This signaled the success of these networking pioneers. Steve Crocker appointed Alex McKenzie and Jon Postel to replace him as Chairmen of the Network Working Group. The Completion Report details how this role changed: “McKenzie and Postel interpreted their task to be one of codification and coordination primarily, and after a few more spurts of activity the protocol definition process settled for the most part into a status of a maintenance effort.”<sup>39</sup>

ARPA was a management body which funded academic computer scientists. ARPA’s funding paved the way for these scientists to create the ARPANET. BBN helped by developing the packet switching techniques which served as the bottom level of transmitting information between sites. The NWG provided an important development in its “Request for Comments” documentation which made possible the developing new protocols.

### **Part III: About RFC’s as “Open” Documentation**

The open exchange of ideas initiated from the very first meeting of the Network Working Group was continued in the Request For Comments. As meeting notes, the RFC’s were meant to keep members updated on the status of various developments and ideas. They were also meant to gather responses from people. RFC-3, “Documentation Conventions,” documents the “rules” governing the production of these notes. Heading RFC-3 are the open distribution rules: “Documentation of the NWG’s effort is through notes such as this. Notes may be produced at any site by anybody and included in this series.”<sup>40</sup> These opening sentences invite anyone willing to be

helpful into the protocol definition process. This is important because all restrictions are lifted by these words, allowing for the open process aimed for. The guide goes on to describe the rules concerning the contents of the RFC's: "The content of a NWG note may be any thought, suggestion, etc. related to the HOST software or other aspect of the network. Notes are encouraged to be timely rather than polished. Philosophical positions without examples or other specifics, specific suggestions or implementation techniques without introductory or background explication, and explicit questions without any attempted answers are all acceptable. The minimum length for a NWG note is one sentence."<sup>41</sup> In RFC-3, Crocker continues to explain the philosophy behind the perhaps unprecedented openness represented: "These standards (or lack of them) are stated explicitly for two reasons. First, there is a tendency to view a written statement as ipso facto authoritative, and we hope to promote the exchange and discussion of considerably less than authoritative ideas. Second, there is a natural hesitancy to publish something unpolished, and we hope to ease this inhibition."<sup>42</sup>

[The entire RFC-3 is reproduced in the **Appendix**.]

This open process encouraged and led to the exchange of information. Technical development is only successful when information is allowed to flow freely and easily between the parties involved. Encouraging participation is the main principle that made the development of the Net possible.

Statements like the ones contained in RFC-3 are democratic in their support of a process of openness. They were written during the late 1960s, a time of popular protest for freedom of speech. People were demanding more of a say in how their countries were run. The open environment needed to develop new technologies is consistent with the cry for more democracy which students and others were raising during the 1960s throughout the world. What is amazing is the collaboration of the NWG (mostly graduate students) and ARPA (a component of the military) during the 1960s and 1970s. This seems unusual given the context of the times, e.g., the student anti-war movement. Robert Braden of the Internet Activities Board reflects on this collaboration: "For me, participation in the development of the ARPAnet and the Internet protocols has been very exciting. One important reason it worked, I believe, is that there were a lot of very bright people all working more or less in the same direction, led by some very wise people in the funding agency. The result was to create a community of network researchers who believed strongly that collaboration is more powerful than competition among researchers. I don't think any other model would have gotten us where we are today."<sup>43</sup> Such collaboration is why the work of these computer scientists led to such an amazing and democratic achievements, the Net and the cooperative culture of the Net."<sup>44</sup>

Calling these notes a "Request for Comment" established a fascinating tradition. It predates the Usenet post, which in a fashion could also be called a "request for comment." Both are the presentation of a particular person's ideas, questions or comments to the general public for comments, criticism or suggestions. Early RFC's established this tradition. Many RFC's are in fact comments on previous RFC's.<sup>45</sup>

## **Part IV: Conclusion**

How were the developments of the ARPANET made possible? None of the participants had previous solutions to any of the problems and tasks that they faced to establish a working packet switched test bed with host to host connectivity. They had to put much thought and work into their research. As the resulting ARPANET was tremendously successful and fulfilled the objectives of

the project ARPA presented, it is important to see what can be learned from the research and research methods from which it emerged. Bernie Cosell, who worked at BBN during this early period, describes the importance of an open process in a developmental situation: “\*no\*one\* had the necessary expertise [and vision] to figure any of this out on their own. The cultures among the early groups were VERY different [– ] multics, sigma-7, IBM ...at Rand, ...PDP-10s at BBN and SRI... [and possibly] UCSB and Utah had pdp-10's, too. The pie-in-the-sky applications ranged over a WIDE landscape, with no one knowing quite where it would lead. Some kind of free, cross-cultural info/idea exchange \*had\* to happen.”<sup>46</sup>

The computer scientists and others involved were encouraged in their work by ARPA’s philosophy of gathering the best computer scientists working in the field and supporting them: “IPT usually does little day-to-day management of its contractors. Especially with its research contracts, IPT would not be producing faster results with such management as research must progress at its own pace. IPT has generally adopted a mode of management which entails finding highly motivated, highly skilled contractors, giving them a task, and allowing them to proceed by themselves.”<sup>47</sup>

The work of the Network Working Group was vital to the development of the ARPANET. Vinton Cerf, another of the graduate students involved with the early protocol development and still closely connected to the Internet, echoed this sentiment when he opened his paper “An Assessment of ARPANET Protocols,” by writing: “The history of the Advanced Research Project Agency resource sharing computer network (ARPANET) is in many ways a history of the study, development, and implementation of protocols.”<sup>48</sup> Cerf supports Cosell’s opinion about the uncertainty and newness of the entire project: “The tasks facing the ARPANET design teams were often unclear, and frequently required agreements which had never been contemplated before (e.g., common protocols to permit different operating systems and hardware to communicate). The success of the effort, seen in retrospect, is astonishing, and much credit is due to those who were willing to commit themselves to the job of putting the ARPANET together.”<sup>49</sup>

The NWG’s work blazed the trail which the developers of the TCP/IP suite of protocols (Transport Control Protocol/ Internet Protocol) successfully followed when the need to expand and include other networks based on other technologies than NCP arose. The principles embodied by RFC-3 and the open RFC documentation process provided a strong foundation which began with NCP and was continued by the work on TCP/IP. NCP was developed in the field and versions of it were released early in its development so various programmers could work on implementing and improving the protocol. In addition all specifications were free and easily available for people to examine and comment on. Through this principle of early release, the problems and kinks were found and worked out in a timely manner. The future developers of TCP/IP learned from the developers of NCP a practice of developing from the bottom up. The bottom-up model allows for a wide-range of people and experiences to join in and perfect the protocol and make it the best possible.

The public funding of the ARPANET project meant that the documentation could be made public and freely available. The documentation was neither restricted nor classified. This open process encouraging communication was necessary for these pioneers to succeed. Research in new fields of study requires that researchers cooperate and communicate in order to share their expertise. Such openness is especially critical when no one person has the answers in advance. In his article, “The Evolution of Packet Switching,” Larry Roberts described the public nature of the process:

“Since the ARPANET was a public project connecting many major universities and research institutions, the implementation and performance details were widely published.”<sup>50</sup>

The people at the forefront of development of these protocols were the members of the Network Working Group, many of whom came from academic institutions, and who therefore had the support and time needed for the research. In summing up the achievements of the process that developed the ARPANET, the ARPANET Completion Report Draft explains: “The ARPANET development was an extremely intense activity in which contributions were made by many of the best computer scientists in the United States. Thus, almost all of the “major technical problems” already mentioned received continuing attention and the detailed approach to those problems changed several times during the early years of the ARPANET effort.”<sup>51</sup>

Fundamental to the ARPANET, as explained by the Completion Report, was the discovery of a new way of looking at computers. The developers of the ARPANET viewed the computer as a communications device rather than only as an arithmetic device.<sup>52</sup> This new view made the building of the ARPANET possible. This view came from the research conducted by those in academic computer science. Such a shift in understanding the role of the computer is fundamental to advancing computer science. The ARPANET research has provided a rich legacy for the further advancement of computer science and it is important that the significant lessons learned be studied and used to further advance the study of computer science.

---

#### Notes for Chapter 7

1. Chapter III, p. 132, Section 2.3.4, ARPANET Completion Report, F. Heart, A. McKenzie, J. McQuillan, D. Walden, Washington, D.C., 1978. (Hereafter, ARPANET Completion Report)
2. ARPANET Completion Report Draft, September 9, 1977, unpublished, p. III-6. (Hereafter, Completion Report Draft)
3. Ibid.
4. Ibid., p. III-7.
5. Interview by William Aspray and Arthur L. Norberg, Tape recording, Cambridge, Massachusetts, 28 October 1988, OH 150, Charles Babbage Institute, University of Minnesota, Minneapolis, Minnesota.
6. Completion Report Draft, p. III-7.
7. Ibid.
8. Ibid., p. III-21.
9. See for example J. C. R. Licklider and Robert Taylor, “The Computer as a Communication Device,” In Memoriam: J. C. R. Licklider 1915-1990, Digital Research Center, Palo Alto, California, August 7, 1990; originally published in Science and Technology, April, 1968.
10. Completion Report Draft, p. III-23
11. Ibid., p. III-24.

12. Ibid.
13. Ibid.
14. David Clark, RFC-1336.
15. See Chapter 8 “The Birth and Development of the ARPANET.” Also, see F. Heart, A. McKenzie, J. McQuillan, D. Walden, ARPANET Completion Report, Washington, D.C., 1978, Chapter III, section 1.1.2, starting on page III-9.
16. Completion Report Draft, pp. III-25 and III-26.
17. ARPANET Completion Report, p. II-8.
18. Completion Report Draft, p. III-32.
19. Ibid., p. III-35 and ARPANET Completion Report, p. II-2.
20. Completion Report Draft, p. III-35.
21. Ibid., p. III-67.
22. Ibid., p. III-39 and personal discussion with Alex McKenzie, November 1, 1993.
23. E-mail message to Com-Priv mailing list (com-priv@psi.com) Subject “Re: RFC1000 (Partial response to part 1)” Date: Nov 27, 1993.
24. Vinton G. Cerf, private e-mail correspondence, dated Nov 27, 1993. Subject: “Re: Early Days of the ARPANET and the NWG”.
25. “The Origins of RFC’s” by Stephen D. Crocker, is contained in RFC-1000 by J. Reynolds and J. Postel, p. 1.
26. The following quotes show some of the reasoning that went into the choice of the initial ARPANET sites.

“CCN’s [The Campus Computing Network of UCLA’s] chance to obtain a connection to the ARPANET was a result of the presence at UCLA of Professor L. Kleinrock and his students, including S. Crocker, J. Postel, and V. Cerf. This group was not only involved in the original design of the network and the Host protocols, but also was to operate the Network Measurement Center (NMC). For these reasons the first delivered IMP was installed at UCLA, and ARPA was thus able to easily offer CCN the opportunity for connection.” (Completion Report Draft, p. III - 689)

“UCLA was specifically asked to take on the task of a ‘Network Measurement Center’ with the objective of studying the performance of the network as it was built, grown, and modified; SRI was specifically asked to take on the task of a ‘Network Information Center’ with the objective of collecting information about the network, about host resources, and at the same time generating computer based tools for storing and accessing that collected information. (Completion Report Draft, p. II-16)

“The accessibility of distributed resources carries with it the need for an information service (either centralized or distributed) that enables users to learn about those resources. This was recognized at the PI [ed. Primary Instigators] meeting in Michigan in the spring of 1967. At the time, Doug Engelbart and his group at the Stanford Research Institute were already involved in research and development to provide a computer-based facility to augment human interaction. Thus, it was decided that Stanford Research Institute would be a suitable place for a ‘Network Information Center’ (NIC) to be established for the ARPANET. With the beginning of implementation of the network in 1969, construction also began on the NIC at SRI.” (Completion Report Draft, p. III-60)

27. Completion Report Draft, p. III-67.
28. E-mail message to Com-Priv mailing list  
Subject: "Re: RFC1000 (End of response to part 1)"  
Date: Nov 27, 1993.
29. RFC-1000.
30. Completion Report Draft, p. III-67.
31. E-mail message to Com-Priv mailing list  
Subject "Subject: Re: RFC1000 (Response to part 2)"  
Date: Nov 27, 1993.
32. Completion Report Draft, p. III-30.
33. RFC-1000, p. 3.
34. Ibid.
35. In RFC-1000, Stephen Crocker reports on the process of the installation of the first IMP.

"[T]ime was pressing: The first IMP was due to be delivered to UCLA September 1, 1969, and the rest were scheduled at monthly intervals.

At UCLA we scrambled to build a host-IMP interface. SDS, the builder of the Sigma 7, wanted many months and many dollars to do the job.

Mike Wingfield, another grad student at UCLA, stepped in and offered to get the interface built in six weeks for a few thousand dollars. He had a gorgeous, fully instrumented interface working in five and one half weeks. I was in charge of the software, and we were naturally running a bit late. September 1 was Labor Day, so I knew I had a couple of extra days to debug the software. Moreover, I had heard BBN was having some timing troubles with the software, so I had some hope they'd miss the ship date. And I figured that first some Honeywell people would install the hardware – IMPs were built out of Honeywell 516s in those days – and then BBN people would come in a few days later to shake down the software. An easy couple of weeks of grace.

BBN fixed their timing trouble, air shipped the IMP, and it arrived on our loading dock on Saturday, August 30. They arrived with the IMP, wheeled it into our computer room, plugged it in and the software restarted from where it had been when the plug was pulled in Cambridge. Still Saturday, August 30. Panic time at UCLA.

The second IMP was delivered to SRI at the beginning of October, and ARPA's interest was intense. Larry Roberts and Barry Wessler came by for a visit on November 21, and we actually managed to demonstrate a Telnet-like connection to SRI."

36. RFC-1000, p. 4.
37. Ibid.
38. Completion Report Draft, p. II-24.
39. Completion Report Draft, p. III-69.



40. RFC-3, p.1.

41. Ibid.

42. Ibid.

43. RFC-1336

44. This democratic community is in danger of being fundamentally altered. This study of the history of the development of the ARPANET in conjunction with Chapter 2, "The Social Forces Behind the Development of Usenet News" are meant to help people understand where the Net has come from, in order to defend it, and try to fight to keep it open and democratic – "the seventh wonder of the world" as a recent ad called the Internet.

45. Some examples of comments upon comments include:

RFC-1 Crocker, S. Host software 1969 April 7

RFC-65 Walden, D. Comments on Host/Host Protocol document #1

RFC-36 Crocker, S. Protocol notes 1970 March 16

RFC-38 Wolfe, S. Comments on network protocol from NWG/RFC #36

RFC-39 Harslem, E.; Heafner, J. Comments on protocol re: NWG/RFC#36

RFC-33 Crocker, S. New Host-Host Protocol 1970 February 12

RFC-47 Crowther, W. BBN's comments on NWG/RFC #33 1970 April 20

46, Bernie Cosell, "Re: RFC1000 – Questions about the origins of ARPANET Protocols 2/2," alt.folklore.computers, Nov. 23, 1993.

47. Completion Report Draft, p. III-47.

48. Vinton Cerf, "An Assessment of ARPANET Protocols," Infotech Education Ltd., Stanford University, California

49. Ibid.

50. Lawrence Roberts, "The Evolution of Packet Switching," *Proceedings of the IEEE*, Vol. 66, no. 11, November 1978, p. 267.

51. ARPANET Completion Report, p. II-24.

52. Completion Report Draft, p. III-24.

---

Special thanks to Alexander McKenzie of BBN, Stephen Crocker of TIS, and Vinton Cerf of CNRI for making research materials available.

---

# Chapter 8

## The Birth and Development of the ARPANET

by Ronda Hauben  
rh120@columbia.edu

“The Nutrition of a Commonwealth consisteth, in the Plenty, and the Distribution of Materials, Conduive to Life.”

Thomas Hobbes The Leviathan

“The method I take...is not yet very usual; for instead of using only comparative and superlative words, and intellectual arguments, I have taken the course (as a Specimen of the Political Arithmetic I have long aimed at) to express myself in terms of Number, Weight, or Measure; to use only arguments of Sense, and to consider only such Causes, as have visible Foundations in Nature; leaving those that depend upon the mutable Minds, Opinions, Appetites, and Passions of particular Men, to the Conservation of others.”

Sir William Petty, Political Arithmetic

### Preface

The creation of a Global Computer Network is one of the surprising developments of our times. This achievement raises the question: What are the factors that nourished the growth and development of this network and what are those creating impediments to its continued development and expansion?<sup>1</sup>

### Introduction

J. C. R. Licklider was one of the early computer pioneers who helped to make the global computer network a reality. His vision of an Intergalactic Computer Network helped to inspire these developments. He and Albert Vezza, describing an earlier networking advance, wrote, “Shakespeare could have been foreseeing the present situation in information networking when he said, ‘...What’s past is prologue; what’s to come, in yours and my discharge’.”<sup>2</sup> The story of the network’s growth and development contains important lessons for its continued expansion. The development of this international network, linking millions of people around the world, now stands at a turning point. Will it continue to go forward or will it be detoured? An understanding of the environment and policies that nourished the development of the global computer network provides a foundation on which to base its further expansion. Such an understanding will also make it possible to contribute to the cooperative networking culture that has evolved and flourished through these policies.

### The Development of the ARPANET

In 1962, the report “On Distributed Communications Networks” by Paul Baran, was published by the Rand Corporation.<sup>3</sup> Baran’s research, funded under a standing contract from the U.S. Air Force, discussed how the U.S. military could protect its communications systems from serious attack. Baran outlined the principle of “redundancy of connectivity” and explored various

models of forming communications systems and evaluating their vulnerability. The report proposed a communications system where there would be no obvious central command and control point, but all surviving points would be able to reestablish contact in the event of an attack on any one point. Thus damage to a part would not destroy the whole and its effect on the whole would be minimized.

One of his recommendations was for a national public utility to transport computer data, much in the way the telephone system transports voice data. “Is it time now to start thinking about a new and possibly non-existent public utility,” Baran asked, “a common user digital data communication plant designed specifically for the transmission of digital data among a large set of subscribers?”<sup>4</sup> He cautioned against limiting the choice of technology to create such a network to that which was currently in use. He proposed that a packet switching, store and forward technology be developed for a data network. The 11 reports he wrote were the first published description of what we now call packet switching.<sup>5</sup> Another networking pioneer, Donald W. Davies, of the United Kingdom, also did important work in this field and has been credited with introducing the term “packet” switching.

Other researchers were interested in computers and communications, particularly in the computer as a communication device. J. C. R. Licklider was one of the most influential. He was particularly interested in the man-computer communication relationship. “Lick”, as he asked people to call him, wondered how the computer could help humans to think and to solve problems. In the article, “Man Computer Symbiosis”, he explored how the computer could help humans to do intellectual work. Licklider was also interested in the question of how the computer could help humans to communicate better.<sup>6</sup> In another article, “The Computer as Communications Device,” Licklider and his co-author Robert Taylor wrote, “In a few years men will be able to communicate more effectively through a machine than face to face.... When minds interact, new ideas emerge.”<sup>7</sup>

What pioneers like Paul Baran and J. C. R. Licklider were proposing was the development of computer technology in a direction that had not been developed before. Larry Roberts, also a pioneer during these important early days of networking, points out that Baran’s work was either classified or otherwise “unfortunately...very sparsely published in the scientific press.”<sup>8</sup> Thus according to Roberts, the impact of Baran’s work on the actual development of packet switching networks “was mainly supportative, not sparking its development.” Licklider, however, had access to military research and writing, but he was also involved in the academic computer science research and education community.<sup>9</sup> Roberts describes how he was influenced by Licklider’s vision of an Intergalactic Computer Network to change his life and career. Licklider’s concept of an Intergalactic Computer Network, Roberts explains, represented the effort to “define the problems and benefits resulting from computer networking.”<sup>10</sup>

After informal conversations with three important computer science pioneers, J. C. R. Licklider, Fernando Corbato and Alan Perlis, at the Second Congress on Information System Sciences in Hot Springs, Virginia, in November 1964, Larry Roberts “concluded that the most important problem in the computer field before us at the time was computer networking; the ability to access one computer from another easily and economically to permit resource sharing.” Roberts recalled, “That was a topic in which Licklider was very interested and his enthusiasm infected me.”<sup>11</sup>

During the early 1960s, the DoD under ARPA established two new funding offices, the IPTO and another for behavioral science. From 1962-64, Licklider took a leave from BBN in Cambridge, Massachusetts to give guidance to these two newly created offices. In reviewing this

seminal period, Alan Perlis recalled how Licklider's philosophy guided ARPA's funding of computer science research. Perlis explained, "I think that we all should be grateful to ARPA for not focusing on very specific projects such as workstations. There was no order issued that said, 'We want a proposal on a workstation.' Goodness knows, they would have gotten many of them. Instead, I think that ARPA, through Lick, realized that if you get 'n' good people together to do research on computing, you're going to illuminate some reasonable fraction of the ways of proceeding because the computer is such a general instrument." In retrospect Perlis concluded, "We owe a great deal to ARPA for not circumscribing directions that people took in those days. I like to believe that the purpose of the military is to support ARPA, and the purpose of ARPA is to support research."<sup>12</sup>

Licklider was guided in his philosophy by the rationale that a broad investigation of a problem was necessary in order to solve the problem. "There's a lot of reason for adopting a broad delimitation rather than a narrow one," he explained, "because if you're trying to find out where ideas come from, you don't want to isolate yourself from the areas that they come from."<sup>13</sup> And he succeeded in attracting others involved in computer research to his vision that computer networking was the most important challenge of the times.

In 1966-67, MIT's Lincoln Laboratory in Lexington, Massachusetts and System Development Corp. (SDC), in Santa Monica, California, got a grant from the U.S. Department of Defense to begin research on linking computers across the continent. Describing this work, Roberts explained, "Convinced that it was a worthwhile goal, we set up a test network to see where the problems would be. Since computer time sharing experiments at MIT [CTSS] and at Dartmouth [DTSS] had demonstrated that it was possible to link different computer users to a single computer, the cross country experiment built on this advance." (i.e. once timesharing was possible, the linking of remote computers became feasible.)<sup>14</sup>

Roberts reported that there was no trouble linking dissimilar computers (the TX-2 computer at Lincoln Laboratory in Massachusetts and the Q-32 computer at SDC in California). The problem, he claimed, was with using telephone circuit switching technology because the throughput was inadequate to accomplish their goals. Thus their experiment set the basis for justifying research in setting up a nationwide store and forward, packet switching data network.

During this period, ARPA was funding computer research at several U.S. universities and research laboratories. The decision was made to include these research contractors in an experimental network, eventually called the ARPANET. A plan was created for a working network to link 16 research groups together. This plan for the ARPANET was made available at the October 1967 ACM Symposium on Operating Principles in Gatlinburg, Tennessee.<sup>15</sup> The planned research was to be encouraged and funded by the IPTO. Roberts was recruited to head the IPTO and guide the research. A Request for Proposal (RFP) set out specifications for the project and asked for bids. Proposals were invited to create an operational network at four sites and to provide a design for a network that could include 16 sites.

The award for the contract went to BBN in January, 1969. The planned network would make use of mini-computers to serve as switching nodes for the host computers at four sites that were to be connected to the network. The Honeywell DDP-516 minicomputers were chosen for the network of interface message processors (IMPs) that would be linked to each other. At first, each of the IMPs (i.e. nodes) would be linked to one host computer. These IMPs were configured with 12,000 16-bit words of memory though they were among the most powerful minicomputers available at the time.

The opening stanzas of a poem by Vint Cerf, an ARPANET pioneer, describe computer science research before and after the ARPANET.

Like distant islands sundered by the sea,  
We had no sense of one community.  
We lived and worked apart and rarely knew  
That others searched with us for knowledge, too.

Distant ARPA spurred us in our quest  
And for our part we worked and put to test  
New thoughts and theories of computing art;  
We deemed it science not, but made a start.

Each time a new machine was built and sold,  
We'd add it to our list of needs and told  
Our source of funds "Alas! Our knowledge loom  
Will halt 'til it's in our computer room.

But, could these new resources not be shared?  
Let links be built; machines and men be paired!  
Let distance be no barrier! They set  
That goal: design and build the ARPANET!<sup>16</sup>

On August 30, 1969, the first IMP arrived at the University of California, Los Angeles (UCLA) which was to be the first site of the new network. It was connected to the SDS Sigma 7 computer at UCLA using the GENIE operating system. Shortly thereafter IMPs were delivered to the other three sites in this initial testbed network. At Stanford Research Institute (SRI), the IMP was connected to an SDS-940 computer using the SEX operating system. At the University of California, Santa Barbara (UCSB), the IMP was connected to an IBM 360/75 using OS/MVT. And at the University of Utah (Utah), the fourth site, the IMP was connected to a DEC PDP-10 using the TENEX operating system.

By the end of 1969, the first four IMPs had been connected to the host computers at their individual sites and the network connections between the IMPs were operational. The researchers and scientists involved could begin to identify the problems they had to solve to develop a working network where there would be communication from host to host.

There were programming and other technical problems to be solved so the different computers would be able to communicate with each other. Also, there was a need for an agreed upon set of signals that would open up communication channels, allow data to pass through, and then close the channels. These agreed upon standards were called protocols. The initial proposal for the ARPANET required that the sites work together to establish the necessary protocols. Beginning in 1968, meetings of a group to discuss establishing these protocols took place.<sup>17</sup> In 1969, the group which called itself the Network Working Group (NWG) began to put together a set of documents that would be available to everyone involved for consideration and discussion. They called these documents Requests For Comment (RFC) and RFC 1, dated April, 1969, was mailed to the

participants.<sup>18</sup>

As the problems of setting up the four computer network were identified and solved, the network was expanded to several more sites.<sup>19</sup> By April 1971, there were 15 nodes and 23 hosts in the network. The earliest sites attached to the network were connected to Honeywell DDP-516 IMPs.<sup>20</sup>

These sites were:

1. UCLA
2. SRI
3. UCSB
4. U. of UTAH
5. BBN
6. MIT
7. RAND Corp
8. SDC
9. Harvard
10. Lincoln Lab
11. Stanford
12. U. of Illinois, Urbana
13. Case Western Reserve U.
14. Carnegie Mellon U. (CMU)
15. NASA-AMES

Then a smaller minicomputer, the Honeywell 316, was utilized. It was compatible with the DDP-516 IMP but was available at half the cost. Some nodes were configured using these smaller minicomputers as TIPs (i.e., Terminal IMPs) beginning with NASA-AMES TIP and MITRE TIP.

By January 1973, there were 35 nodes of which 14 were TIPs including a satellite link which connected California with a TIP in Hawaii.

With the rapid increase of network traffic, problems were discovered with the reliability of the subnet and corrections had to be worked on. In mid 1973, Norway and England were added to the Net by a low speed line, adding to the problems to be solved. By September 1973, there were 40 nodes and 45 hosts on the network. And the traffic had expanded from 1 million packets per day in 1972 to 2.9 million packets per day by September, 1973.

By 1977, there were 111 host computers connected via the ARPANET. By 1983 there were 4,000.<sup>21</sup>

As the network was put into operation, the researchers learned which of their original assumptions and models were sound and which were inaccurate. For example, BBN describes how they had initially failed to understand that the IMPs would need to do error checking of the IMP/host interface. They explain: "The first four IMPs were developed and installed on schedule by the end of 1969. No sooner were these IMPs in the field than it became clear that some provision was needed to connect hosts relatively distant from an IMP (i.e., up to 2,000 feet instead of the expected 50 feet). Thus, in early 1970 a 'distant' IMP/host interface was developed. Augmented simply by heavier line drivers, these distant interfaces made clear, for the first time, the fallacy in the assumption that had

been made that no error control was needed on the host/IMP interface because there would be no errors on such a local connection.”<sup>22</sup>

The expanding operational network made it possible to uncover the actual bugs and make the needed corrections. In describing the importance of an operational network to the research efforts, as opposed to being limited to a laboratory model, Alex McKenzie and David Walden, in their article “ARPANET, the Defense Data Network, and Internet” write: “Errors in coding control were another problem. However carefully one designs, codes, and performs quality control, errors can still slip through. Fortunately, with a large number of IMPs in the network, most of these errors are found quickly because they occur so frequently. For instance, a bug in an IMP code that occurs once a day in one IMP, occurs every 15 min in a 100-IMP network. Unfortunately, some bugs still will remain. If a symptom of a bug is detected somewhere in a 100-IMP network once a week (often enough to be a problem), then it will happen only once every 2 years in a single IMP in a development lab for a programmer trying to find the source of the symptom. Thus, achieving a totally bug-free network is very difficult.”<sup>23</sup>

In October 1972, the First International Conference on Computer Communications was held in Washington, D.C. A public demonstration of the ARPANET was given, setting up an actual node with 40 terminals. Representatives from projects around the world including Canada, France, Japan, Norway, Sweden, Great Britain and the U.S. discussed the need to begin work establishing agreed upon protocols. The InterNetwork Working Group (INWG) was created to begin discussions for a common protocol and Vinton Cerf, who was involved with UCLA ARPANET, was chosen as the first chairman. The vision proposed for the international interconnection of networks was “a mesh of independent, autonomous networks interconnected by gateways, just as independent circuits of ARPANET are interconnected by IMPs.”<sup>24</sup>

The network continued to grow and expand.

In 1975, the ARPANET was transferred to the control of the Defense Communications Agency (DCA).

Evaluating the success of ARPANET research, Licklider recalled that he felt ARPA had been run by an enlightened set of military men while he was involved with it. “I don’t want to brag about ARPA,” he explained, “It is in my view, however, a very enlightened place. It was fun to work there. I think I never encountered brighter, more creative people, than the inhabitants of the third floor E-ring of the Pentagon. But that, I’ll say, was a long time ago, and I simply don’t know how bright and likeable they are now. But ARPA didn’t constrain me much.”<sup>25</sup>

The following description of his experience on the early ARPANET, was posted on Usenet by Eugene Miya, who had been a student at one of the early ARPA sites. His experience documents the encouraging research environment of the early ARPANET. He wrote: “It was an effort to connect different kinds of computers back when a school or company had only one (that’s 1) computer. The first configuration of the ARPANET had only 4 computers, I had luckily selected a school at one of those 4 sites: UCLA/Rand Corp, UCSB (us), SRI, and the U of Utah.

“Who? The U.S. DoD: Defense Department’s Advanced Research Projects Agency. ARPA was the sugar daddy of computer science. Some very bright people were given some money, freedom, and had a lot of vision. It not only started computer networks, but also computer graphics, computer flight simulation, head mounted displays, parallel processing, queuing models, VLSI, and a host of other ideas. Far from being evil warmongers, some neat work was done.

“Why? Lots of reasons: intellectual curiosity, the need to have different machines communicate, study fault tolerance of communications systems in the event of nuclear war, share and connect expensive resources, very soft ideas to very hard ideas....”

“I first saw the term ‘internetwork’ in a paper from Xerox PARC (another ARPANET host). The issue was one of interconnecting Ethernets (which had the 256 [slightly less] host limitation)....”

“I learned much of this with the help of the NIC (Network Information Center). This does not mean the Internet is like this today. I think the early ARPANET was kind of a wondrous neat place, sort of a golden era. You could get into other people’s machines with a minimum of hassle (someone else paid the bills). No more....”

He continued: “Where did I fit in? I was a frosh nuclear engineering major, spending odd hours (2 a.m. - 4 a.m., sometimes on Fridays and weekends) doing hackerish things rather than doing student things: studying or dating, etc. I put together an interactive SPSS and learned a lot playing chess on an MIT[-MC] DEC-10 from an IBM-360. Think of the problems: 32-bit versus 36-bit, different character set [remember I started with EBCDIC], FTP then is largely FTP now, has changed very little. We didn’t have text editors available to students on the IBM (yes you could use the ARPANET via punched card decks). Learned a lot. I wish I had hacked more.”<sup>26</sup>

Miya’s account, describing the stimulating research environment created via the ARPANET, documents how this environment not only helped to create a global computer network, but also led to the creation of new networking applications. For example, one of the surprising developments to the researchers of the ARPANET was the great popularity of electronic mail. Analyzing the reasons for this unanticipated benefit from their ARPANET research, Licklider and Vezza wrote: “By the fall of 1973, the great effectiveness and convenience of such fast, informed messages services... had been discovered by almost everyone who had worked on the development of the ARPANET – and especially by the then Director of ARPA, S. J. Lukasik, who soon had most of his office directors and program managers communicating with him and with their colleagues and their contractors via the network. Thereafter, both the number of (intercommunicating) electronic mail systems and the number of users of them on the ARPANET increased rapidly.”<sup>27</sup>

They describe the advantages of e-mail explaining “that, in an ARPANET message, one could write tersely and type imperfectly, even to an older person in a superior position and even to a person one did not know very well, and the recipient took no offense.” there was none of the formality and perfection that were required in a typed letter. They reasoned that perhaps this was “because the network was so much faster, so much more like the telephone.... Among the advantages of the network message services over the telephone were the fact that one could proceed immediately to the point without having to engage in small talk first, that the message services produced a preservable record, and that the sender and receiver did not have to be available at the same time.”<sup>28</sup>

Concurring about the importance of the development of e-mail, the authors of The Completion Report (1978) wrote: “The largest single surprise of the ARPANET program has been the incredible popularity and success of network mail. There is little doubt that the techniques of network mail developed in connection with the ARPANET program are going to sweep the country and drastically change the techniques used for intercommunication in the public and private sectors.”<sup>29</sup>

Not only did the ARPANET make it possible to learn what the actual problems of networking



would be, the communication it made possible gave the researchers the ability to collaborate to solve the problems.

Summarizing the important breakthrough this technology represented, the authors of “The Completion Report” wrote:<sup>30</sup>

“This ARPA program has created no less than a revolution in computer technology and has been one of the most successful projects ever undertaken by ARPA. The program has initiated extensive changes in the Defense Department’s use of computers as well as in the use of computers by the entire public and private sectors, both in the United States and around the world.”

“Just as the telephone, the telegraph, and the printing press had far-reaching effects on human intercommunication,” they noted, “the widespread utilization of computer networks which has been catalyzed by the ARPANET project represents a similarly far-reaching change in the use of computers by mankind.”

“The full impact of the technical changes set in motion by this project may not be understood for many years,” they appropriately concluded.

---

#### Notes for Chapter 8

1. See for example the methodology used by Sir William Petty in *Political Arithmetic* in *The Writings of Sir William Petty*, ed. Charles Hull, London, 1899, reprint, Augustus Kelley Publishers, New York, 1986, pp. 244-313.

2. “Applications of Information Network”, *Proceedings of the IEEE*, Vol. 66 no. 11, November, 1978, p. 57.

3. Paul Baran, “On Distributed Communications Networks,” Rand Corporation, September, 1962, p. 2. (Correspondence from Willis Ware of Rand Corporation indicated that this report was created under a standing contract with the U.S. Air Force.)

4. *Ibid.*, p. 40. See also, P. Baran et al, “On Distributed Communications,” Vol. I through XI, Memorandum, RAND Corporation, August, 1964. See also description by Larry Roberts, “The ARPANET and Computer Networks” reprinted in *A History of Personal Workstations*, ed. by Adele Goldberg, New York, 1988, 147.

5. See “The Evolution of Packet Switching,” *Proceedings of the IEEE*, Vol. 66 no. 11, November, 1978, pp. 266-7. See also “The ARPANET and Computer Networks,” in *A History of Personal Workstations*, p. 143.

6. See “Man Computer Symbiosis” in *In Memoriam: J. C. R. Licklider 1915-1990*, Digital Research Center, August 7, 1990, pp. 1-19.

7. “The Computer as a Communication Device” in *In Memoriam: J. C. R. Licklider 1915-1990*, p. 21.

8. “The ARPANET and Computer Networks,” p. 144.

9. *Ibid.*

10. See “The ARPANET and Computer Networks”, p. 143.

11. *Ibid.*, pp. 143-144.

12. *Workstations*, *Ibid.*, p. 129.

13. "Some Reflections on Early History," Workstations, p. 118. Licklider also commented on how people who were opposed to Defense research during the 1960s wrote proposals for research to ARPA to spend money on something other than airplane carriers. See p. 130.
14. "Toward a Cooperative Network of Time-Shared Computers," by Thomas Marill and Lawrence G. Roberts, Proceedings – FJCC, 1966, p. 426.
15. Roberts, Workstations, p. 146. Describing ARPA's decision to build a network to connect the computer science and research contractors as the plan for the ARPANET, Roberts writes: "These projects and their computers provided an ideal environment for an experimental network project; consequently the ARPANET was planned during 1967 with the aid of these researchers to link these project's computers together. One task was to develop a computer interface protocol acceptable to all 16 research groups. A second task was to design a new communications technology to support 35 computers at 16 sites with 500,000 packets/day traffic. The initial plan for the ARPANET was published in October 1967 at the ACM Symposium on Operating System Principles in Gatlinburg Tennessee." (pp. 145-146) Also, Roberts describes the network design for the ARPANET. He writes, "The communications network design was that of the now conventional packet network; interface message processors (IMPs) at each node interconnected by leased telecommunication lines providing a store and forward service on very short messages." (p. 146)
16. From "Requiem for the ARPANET" by Vinton G. Cerf, Users' Dictionary of Computer Networks, Bedford, MA, 1989. Used with permission.
17. A description of the beginning of the Network Working Group, "The Origins of RFCs" by Stephen D. Crocker, is contained in RFC 1000 by J. Reynolds and J. Postel.
18. See The Completion Report, by F. Heart, A. McKenzie, J. McQuillian, and D. Walden, BBN Report 4799, January 4, 1978, pp. III 46-48.
19. Ibid.
20. List of sites based on a posting on Usenet by Joel Levin on Oct. 17, 1990. The Completion Report confirms these sites, but names Burroughs as one of the first 15 sites. 1. UCLA (University of California at Los Angeles), 2. SRI (Stanford Research Institute), 3. UCSB (University of California at Santa Barbara)
21. See The Completion Report and "ARPANET, the Defense Data Network, and Internet" in the Froehlich/Kent Encyclopedia of Telecommunications, vol. 1.
22. The Completion Report, p. III-55.
23. See The Completion Report and "ARPANET, the Defense Data Network, and Internet" in the Froehlich/Kent Encyclopedia of Telecommunications, vol. 1, p. 361.
24. Ibid. pp. 361-2.
25. Workstations, p. 126.
26. From Eugene Miya in alt.folklore.computers, comp.misc, "Re: Internet: The Origins," October 16, 1990.
27. "Applications", p. 44.
28. Ibid.
29. The Completion Report, pp. III 113-116.

30. Ibid., p. I-2.

---

Thanks to Harvey Lynn of Rand Corporation and Alex McKenzie of BBN for making important materials available and to Bernard Galler, John A. N. Lee and Judy O'Neill for commenting on versions of this article.

---

## Chapter 9

# On the Early History and Impact of Unix Tools to Build the Tools for a New Millennium

“When the barbarian, advancing step by step, had discovered the native metals, and learned to melt them in the crucible and to cast them in moulds; when he had alloyed native copper with tin and produced bronze; and, finally, when by a still greater effort of thought he had invented the furnace, and produced iron from the ore, nine tenths of the battle for civilization was gained. Furnished, with iron tools capable of holding both an edge and a point, mankind were certain of attaining to civilization.”

Lewis Henry Morgan

“When Unix evolved within Bell Laboratories, it was not a result of some deliberate management initiative. It spread through channels of technical need and technical contact.... This was typical of the way Unix spread around Bell Laboratories.... I brought out that little hunk of history to point out that the spread and success of Unix, first in the Bell organizations and then in the rest of the world, was due to the fact that it was used, modified, and tinkered up in a whole variety of organizations.”

Victor Vyssotsky

“UNIX is a lever for the intellect.”

John R. Mashey

### Iron Tools and Software Tools

Our era is witnessing the birth of an important new technology, different from any in the past. This new technology is the technology of software production. The new tools of our era are tools that make it possible to produce software. Unlike the tools forged in the past, software tools are not something you can grab or hold. Similarly the processes required to develop this new technology are new. Knowing the details of how such new tools have been created will make it possible to better understand the promise they present for our times and for the future.

Describing a previous technological leap, the American anthropologist Lewis Henry Morgan called the production of iron “the event of events in human experience.... Out of it,” he wrote, “came the metallic hammer and anvil, the axe and the chisel, the plow with an iron point, the iron sword; in fine, the basis of civilization which may be said to rest upon this metal.” Morgan maintained that until the human species had mastered the skills and methods necessary for forging iron, social development had been paused awaiting the achievement of this next necessary technological advance. “The want of tools,” he wrote, “arrested the progress of mankind in barbarism. There they would have remained to the present hour, had they failed to bridge the chasm.” Frustrated by the lack of knowledge about how iron was invented, Morgan lamented, “It would be a singular satisfaction could it be known to what tribe and family we are indebted for this knowledge and with it for civilization.”<sup>1</sup>

We are more fortunate than our ancestors. We can know some of the details of whom to

credit for the technological achievements of the computer revolution. Documenting their contributions while many of the pioneers are still alive will provide future generations with knowledge of whom they are indebted to for this new and stunning advance. Like the invention of iron, the technology of software production promises to make possible an important new level of civilization. Such knowledge will also help those who want to build on what has already been achieved.

## **Creating a Technology of Software Development**

In *Introduction to Computer Science*, Alan Perlis, one of the early pioneers of computer science, observed that the nature of computer programming is too often misunderstood. “Although computer programming is often inaccurately considered an art,” he wrote, “it is actually an infant technology.”<sup>2</sup>

Like Morgan, Perlis believed that the progressive and accelerating ability to use and produce tools represents a defining element in the advance of human society. He observed that the rapid increase in the ability of the human species to use and mold new tools was guided by two laws. The first law describes the rapid increase in the number of people who can wield the tools and in the number of tools available. The second describes how the rapid development of tools to create tools guarantees the continuing expansion of the ability to use the tools. Perlis wrote: “First, the increase in the population trained to use tools and in the number of tools available for their use implies that the potential for tool utility grows as a product rather than as a sum. Second, the amount of composition – that is the number and uses of tools to produce new tools – grows as a part of the product and assures an even greater overall growth in tool use.”<sup>3</sup>

Among the tools Perlis saw coming into common use “are those which perform automatically at great speed, and whose management must perform in the same way. Here,” he explained, “the computer is essential because of the need for quick response.”<sup>4</sup>

## **The Creation of Time-Sharing**

The development of the modern computer raised the question of how to manage its operation more efficiently. An important solution to the problem of computer management proposed in 1959 was the sequential and automatic interspersing of many tasks known as time-sharing.<sup>5</sup> Among the earliest time-sharing systems were those developed at MIT. By 1963, there were two versions of the Compatible Time Sharing System (CTSS) operating at MIT on two IBM 7094 computers, one at the Computation Center, and another at MIT’s Project MAC.

Those using the early time-sharing systems at MIT and elsewhere soon discovered the delights of interactive computing made possible by time-sharing.<sup>6</sup> Describing the advantages of interactive computing, time-sharing pioneers Robert Fano and Fernando Corbato, wrote: “For professional programmers the time-sharing system has come to mean a great deal more than mere ease of access to the computer. Provided with the opportunity to run a program in continuous dialogue with the machine, editing, ‘debugging’ and modifying the program as they proceed, they have gained immeasurably in the ability to experiment. They can readily investigate new programming techniques and new approaches to problems.”<sup>7</sup>

The results of this programming flexibility led both to a bolder and more flexible approach to problem solving and to undertaking new areas of research. Fano and Corbato reported that users

not only would build on each other's work, but also they would come to depend more and more on the computer to facilitate their work. The most surprising development that they encountered, however, was the fact that the users themselves created many of the programming commands used in the system, instead of needing professional programmers. While at the conventional computer installation, they noted, "one hardly ever makes use of a program developed by another user, because of the difficulty of exchanging programs and data," in the Project MAC time-sharing environment, "the ease of exchange has encouraged investigators to design their programs with an eye to possible use by other people. They have acted essentially as if they were writing papers to be published in technical journals."<sup>8</sup>

Fano and Corbato envisioned that time-sharing systems would have a profound impact on the future. "Communities will design systems," they predicted, "to perform various functions – intellectual, economic and social – and the systems in turn undoubtedly will have profound effects in shaping the patterns of human life."<sup>9</sup> "The coupling between such a utility and the community it serves," they discovered, "is so strong that the community is actually a part of the system itself." They foresaw the development of a symbiotic relationship between the computer systems and its human users which "will create new services, new institutions, a new environment and new problems." Among these, they proposed, would be the question of access. "How will access to the utility be controlled?" they asked, "To what ends will the system be devoted, and what safeguards can be devised for its misuses? It is easy to see," they concluded, "that the progress of this new technique will raise many social questions as well as technical ones."<sup>10</sup>

Others during this period were concerned with the impact the computer would have on current society. For example, John McCarthy predicted that, "The computer gives signs of becoming the contemporary counterpart of the steam engine that brought on the industrial revolution."<sup>11</sup> Unlike the steam engine, however, the utility of the computer was dependent on the successful development of software programs written to direct it. Therefore, along with the increasing speed and capacity of computer hardware, came the increase in the demand for and in the cost of software. By the mid 1960's, the U.S. government was spending increasing amounts of money to create programs to utilize computer resources. The U.S. government, wrote McCarthy, "with a dozen or so big systems serving its military and space establishments, is spending more than half of its 1966 outlay of \$844 million on software."<sup>12</sup>

Pointing out the need for studying the processes of programming, McCarthy observed, "What computers can do, depends on the state of the art and the science of programming as well as on speed and memory capacity."<sup>13</sup> Computer pioneers like McCarthy recognized that the computer was more than an efficient bookkeeping machine. There was a need to discover what new applications were possible, and to create these new applications. Therefore, there would be a need for breakthroughs in the process of programming software. McCarthy believed that it was important for the user to be able to program in order to realize the potential of the computer. He pointed out that programming was a skill that was not difficult to learn, and that it was more important to understand the task being automated than to master programming languages. "To program the trajectory of a rocket," McCarthy offers as an example, "requires a few weeks' study of programming and a few years' study of physics."<sup>14</sup>

These early explorations in time-sharing prepared the foundation for an important development in the process of creating software. Once the discovery was made that simple

programming tools could be created to aid in the process of software development, and that such tools could help those who understood the tasks to be automated, a needed leap could be made in how to develop software. Such a program was to be carried out by research programmers and developers at Bell Labs in the 1970's and early 1980's, building on the principles developed by the pioneers of time-sharing and Project MAC.

## **The Multics Collaboration**

In 1964, MIT joined with GE and AT&T in a project designed to implement time-sharing by developing a new computer and a new operating system. The joint research project among GE, MIT, and AT&T was created to extend time-sharing techniques from a pilot program into a useful prototype for the future information utility.<sup>15</sup> The researchers realized that there was no existing computer that would meet the demands of time-sharing. Therefore part of the goal of their collaboration was to make it possible to develop a new computer as well as a new operating system.

The collaborative project was called Multics [Multiplexed Information and Computing Service] and was to be implemented on the GE 645 computer.<sup>16</sup> Technical leadership of the project included F. J. Corbato from MIT and V.A. Vyssotsky from Bell Labs. "One of the overall design goals is to create a computing system," they wrote, "which is capable of meeting almost all of the present and near-future requirements of a large computer utility. Such systems, must run continuously and reliably 7 days a week, 24 hours a day in a way similar to telephone or power systems, and must be capable of meeting wide service demands: from multiple man-machine interaction to the sequential processing of absentee-user jobs...."<sup>17</sup>

The goal of the research was to produce a prototype time-sharing system. Berkley Tague, one of the Bell Labs researchers involved in the Multics project writes, "The Multics Project was a joint project of Bell Labs, the GE Computer Systems Division, and MIT's Project MAC to develop a new computer and operating system that would replace MIT's CTSS system, Bell Labs BESYS, and support the new GE machine."<sup>18</sup> Though AT&T withdrew from the project in 1969, the joint work achieved significant results. Summarizing these achievements, Tague writes, "Multics was one of the seminal efforts in computing science and operating system design. It established principles and features of operating system design that are taken for granted today in any modern operating system."<sup>19</sup>

## **The Need for Operating System Research at Bell Labs**

Even though AT&T withdrew from the research collaboration on Multics, computer scientists at Bell Labs wanted some way of continuing the advanced form of programming research that their work with CTSS and Multics had made possible. As early as 1957, Bell Labs had found they needed an operating system for their in-house computer center which was then running lots of short batch jobs. Describing the situation then facing the Labs, Vyssotsky explains, "We just couldn't take the time to get them on and off the machine manually. We needed an operating system to sequence jobs through and control machine resources."<sup>20</sup> The BESYS operating system was created at Bell Labs to deal with their in-house needs. When asked by others outside the Labs to make a copy available, they did so but with no obligation to provide support. "There was no support when we shipped a BESYS tape to somebody," Vyssotsky recalls, "we would answer reasonable questions over the telephone. If they found troubles or we found troubles, we would provide fixes."<sup>21</sup>

By 1964, however, Bell Labs was adopting third generation computer equipment and had to decide whether they would build their own operating system or go with one that was built outside the Labs. Vyssotsky recounts the process of deliberation at the time, “Through a rather murky process of internal deliberation we decided to join forces with General Electric and MIT to create Multics.” He explains that the Labs planned to use the Multics operating system, “as a mainstay for Bell Laboratories internal service computing in precisely the same way that we had used the BESYS operating system.”<sup>22</sup>

## **Unix Is Born and the Introduction of Pipes**

When AT&T made the decision to pull out of the Multics collaboration, they took the research operating system off their GE 645 computer and put up the GECOS operating system. Though GECOS was adequate for applications, it was “nowhere near as satisfactory if you were trying to do things that were technically difficult and imperfectly defined,” explained Vyssotsky, “which is the main task of research.”<sup>23</sup>

For the pioneering work of Bell Labs research programmers like Ken Thompson and the research purposes of the Labs, an operating system more like what Multics had promised was needed. Along with the advantages of immediate feedback which time-sharing provided, the Bell Labs researchers wanted to continue the ability to work collaboratively which time-sharing had made possible.

“What we wanted to preserve,” one of the creators of Unix, Dennis Ritchie writes, “was not just a good programming environment in which to do programming, but a system around which a fellowship could form. We knew from experience that the essence of communal computing, as supplied by remote-access, time-shared machines, is not just to type programs into a terminal instead of a keypunch, but to encourage close communication.”<sup>24</sup>

Ritchie describes how an informal group led by Thompson had begun investigating alternatives to Multics before the GE-645 Multics machine had been removed from the Labs.<sup>25</sup> Thompson and Ritchie presented Bell Labs with proposals to buy them a computer so they could build their own interactive, time-sharing operating system. Their proposals weren’t acted on. Eventually, Thompson found a little used PDP-7 computer. According to Vyssotsky, the orphaned PDP-7 computer was a machine, “more in the class of a Commodore 64 than the class of a PC-AT.”<sup>26</sup>

Ritchie, Thompson, and Rudd Canaday, who had been part of the Multics project, applied the lessons they had learned to the design of a file system for an experimental operating system. Writing on a chalk board, they created a file system design based on “the particularly simple way of viewing files that was suggested by the Multics I/O system.”<sup>27</sup>

“Soon,” Ritchie recounts, “Thompson began implementing the paper file system (perhaps ‘chalk file system’ would be more accurate) that had been designed earlier.” Thompson was eager to get a working model so he could test it out. He proceeded to create the other aspects of an operating system. “A file system without a way to exercise it was a sterile proposition,” notes Ritchie, “so he [Thompson] proceeded to flesh it out with the other requirements for a working operating system, in particular the notion of processes.”<sup>28</sup>

Describing the primitive conditions that Thompson faced, Ritchie writes, “At the start, Thompson did not even program on the PDP itself, but instead used a set of macros for the GEMAP



assembler on a GE-635 machine.”<sup>29</sup> A paper tape was generated on the GE 635 and then tested on the PDP-7 until, according to Ritchie, “a primitive Unix kernel, an editor, an assembler, a simple shell (command interpreter), and a few utilities (like the Unix `rm`, `cat`, `cp` commands) were completed. At this point, the operating system was self-supporting, programs could be written and tested without resort to paper tape, and development continued on the PDP-7 itself.”<sup>30</sup>

Ritchie describes how Thompson’s PDP-7 assembler was a model of simplicity. “There were no libraries, no loader or link editor,” he writes, “the entire source of a program was presented to the assembler, and the output file – with a fixed name – that emerged was directly executable.”<sup>31</sup> Once the assembler was completed, “the system was able to support itself. And thus the operating system we now call UNIX was born,” notes Ritchie.<sup>32</sup>

Among the other active contributors during this period were Bell Labs researchers Rudd Canaday and Joe Ossanna. The researchers were anxious to continue their work on a more advanced computer than the PDP-7. However, their efforts to get AT&T to buy them a more advanced computer for their time-sharing research hadn’t succeeded. With the help of Ossanna and another Bell Labs researcher Lee McMahon, they were finally able to convince management to buy them a new PDP-11 computer. To obtain this agreement, though, the researchers promised to create a text processing system. Doug McIlroy explains that “typists everywhere were seen as potential customers of the promised document-preparation program. Only later did Ossanna spot the patent department as a ripe candidate. By then there had already been clients in the telephone-operations business.”<sup>33</sup> By Spring 1971, the Bell Labs Unix pioneers had a text formatter, fulfilling their promise. They had translated one they had created for the PDP-7 from a program McIlroy had written in the BCPL language while working on Multics. The text formatter was an assembler program for the PDP-11.<sup>34</sup>

One of the important developments in Unix was the introduction of pipes. Pipes had been suggested by McIlroy during the early days of creating Unix. Ritchie explains how “the idea, explained one afternoon on a blackboard, intrigued us but failed to ignite any immediate action. There were several objections to the idea as put.... What a failure of imagination,” he admits.<sup>35</sup> McIlroy concurs, describing how the initial effort to add pipes to Unix occurred about the same time in 1969 that Ritchie, Thompson and Canaday were outlining ideas for a file system. “That was when,” he writes, “the simple pipeline as a way to combine programs, with data notationally propagating along a chain of (not necessarily concurrent) filters was articulated.”<sup>36</sup> However, pipes weren’t implemented in Unix until 1972. “Thanks to McIlroy’s persistence,” Ritchie writes, “pipes were finally installed in the operating system (a relatively simple job), and a new notation was introduced.”<sup>37</sup> Several of the old commands had to be changed to make them usable with pipes, notes Ritchie. Summarizing how pipes found their way into Unix, Vyssotsky notes that Thompson put them in, but “it was McIlroy who said, ‘look you ought to do it.’ Pipes, like most things in Unix were not a radically new idea.”<sup>38</sup> He describes how similar ideas had appeared in other languages like SIMULA as early as 1967.

Dick Haight, a Unix pioneer who helped develop the Programmer’s Workbench of Unix tools, was present the day that pipes were introduced into Unix. He describes what happened: “I happened to have been visiting the research crew the day they implemented pipes. It was clear to everyone practically minutes after the system came up with pipes working that it was a wonderful thing. Nobody would ever go back and give that up if they could help it.”<sup>39</sup> Also describing the day pipes were introduced, McIlroy writes: “Open Systems! Our Systems! How well those who were

there remember the pipe-festooned garret where Unix took form. The excitement of creation drew people to work there amidst the whine of the computer's cooling fans, even though almost the same computer access, could be had from one's office or from home. Those raw quarters saw a procession of memorable events. The advent of software pipes precipitated a day-long orgy of one-liners.... As people reveled in the power of functional composition in the large, which is even today unavailable to users of other systems."<sup>40</sup>

## The Software Tool

Pipes had been created by the time the Version 3 Unix Manual appeared in February 1973. The date listed for the creation of pipes is January 15, 1973.<sup>41</sup> Not only were pipes a significant addition to Unix, but according to McIlroy, pipes made possible a subsequent important discovery. "In another memorable event," he writes, "the unarticulated notion of software tools, which had been bolstered by pipes, was finally brought home by the liberation of the pattern matching program `grep` from within the editor."

McIlroy describes how he asked Thompson to create a program to help him with some work that he was trying to do. This program resulted in the invention of the software tool 'grep.' Following is McIlroy's account of how 'grep' was taken out from the editor, leading to a clearer understanding of the notion of a software tool. He writes: "Grep was invented for me. I was making a program to read text aloud through a voice synthesizer. As I invented phonetic rules I would check Webster's dictionary for words on which they might fail. For example, how do you cope with the digraph 'ui', which is pronounced many different ways: 'fruit', 'guile', 'guilty', 'anguish', 'intuit', 'beguine'? I would break the dictionary up into pieces that fit in `ed`'s limited buffer and use a global command to select a list. I would whittle this list down by repeated scannings with 'ed' to see how each proposed rule worked."

"The process was tedious, and terribly wasteful, since the dictionary had to be split (one couldn't afford to leave a split copy on line). Then 'ed' copied each part into /tmp, scanned it twice to accomplish the 'g' command, and finally threw it away, which takes time too."

"One afternoon I asked Ken Thompson if he could lift the regular expression recognizer out of the editor and make a one-pass program to do it. He said yes. The next morning I found a note in my mail announcing a program named `grep`. It worked like a charm. When asked what that funny name meant, Ken said it was obvious. It stood for the editor command that it simulated, `g/re/p` (global regular expression print)."

"Progress on my talking program accelerated dramatically. From that special-purpose beginning, `grep` soon became a household word. (Something I had to stop myself from writing in the first paragraph above shows how firmly naturalized the idea now is: 'I used `ed` to `grep` out words from the dictionary.') More than any other single program, `grep` focused the viewpoint that Kernighan and Plauger christened and formalized in 'Software Tools': make programs that do one thing and do it well, with as few preconceptions about input syntax as possible."<sup>42</sup>

`grep` is listed in the Manual for Version 4 Unix which is dated November, 1973. The date given for the creation of `grep` is March 3, 1973, following the creation of pipes.<sup>43</sup> The creation of `grep`, McIlroy explains, was followed by the invention of other special purpose software programs that could be used as tools. He writes: "A while later a demand arose for another special-purpose program, `gres`, for substitution: `g/re/s`. Lee McMahon undertook to write it, and soon foresaw that

there would be no end to the family: g/re/d, g/re/a, etc. As his concept developed it became sed, a tool of remarkable utility that is largely unappreciated today, because it capitalizes on the perfect familiarity with ed that was universal ten years ago, but no more. Sed covers a lot of needs. For example, we have never seen fit to adopt the widespread ‘head’ utility because ‘sed 10q’ does just as well.”<sup>44</sup>

What McIlroy refers to as “the unarticulated notion of software tools...brought home by the liberation of...grep from within the editor” has become one of the significant achievements of the Bell Labs research work on Unix. By making it possible to use ‘grep’ to search for a data pattern in a file, without having to use an editor to go inside the file, the Bell Labs researchers discovered that they could create a plethora of software tools to be used in varying combinations, thus facilitating the customized application by a user, a goal sought by those doing research in programming.<sup>45</sup>

McIlroy explains that the notion of “software tool” only became articulated among the Bell Labs researchers with the publication of the book “Software Tools” by Kernighan and Plauger. “Still unnamed in our circles,” McIlroy notes, “until Kernighan and Plauger wrote their book, the idea nevertheless became a guiding principle.”<sup>46</sup> McIlroy adds that “We were definitely building tools before K&P, though we lacked the suggestive word.”<sup>47</sup>

Describing how the notion of software tools helps to create an effective programming environment, Brian W. Kernighan and Rob Pike, authors of *The Unix Programming Environment*, explain that each tool is designed to be used with other tools and achieving this end is more important than how each is designed internally.<sup>48</sup> The most important aspect considered with each tool is the function that the tool is to have. New tools are designed once a new function is intended.

“The guiding principle,” they write, “for making the choice should be that each program does one thing. Options are appropriately added to a program that already has the right functionality. If there is no such program then a new program is called for. In that case the usual criteria for program design should be used: the program should be as general as possible, its default behavior should match the most common usage and it should cooperate with other programs.”<sup>49</sup>

Thus Unix, according to Kernighan and Pike, created “a new style of computing, a new way of thinking of how to attack a problem with a computer. This style,” they explain, “was based on the use of tools: using programs separately or in combination to get a job done, rather than doing it by hand, by monolithic self-sufficient subsystems, or by special-purpose, one-time programs.”<sup>50</sup>

The philosophy of using software tools that developed from research in Unix is outlined in the “Foreword” to the special issue of *The Bell System Technical Journal* published in 1978, on “The Unix Time-sharing System.” Describing the principles which they have found to be an important component of the Unix philosophy of software design, the researchers write: “UNIX utilities are usually thought of as tools – sharply honed programs that help with generic data processing tasks. Tools were often invented to help with the development of UNIX programs and were continually improved by much trial, error, discussion, and redesign, as was the operating system itself. Tools may be used in combination to perform or construct specific applications.”<sup>51</sup>

They explain that a distinctive style evolved as part of Unix research. “Unix software works smoothly together; elaborate computing tasks are typically composed from loosely coupled small parts, often software tools taken off the shelf.”<sup>52</sup>

“Sophisticated tools to make tools have evolved,” they observe.<sup>53</sup> Software development tools such as “nroff” and “troff” were created. Not only was it important to create tools, but soon tools to

create tools, like “yacc” and “lex”, were developed. “Yacc” and “lex” were used to create numerous little languages and applications like eqn and awk that greatly enhanced the popularity of Unix.

The evolution of the tool “diff” created by McIlroy, a commonly used software tool, is an example of how tools were continually improved, based on the experience gained using them. McIlroy reports how he based his work on algorithms created by others and then he tried three different algorithms before settling on the one finally used.<sup>54</sup> Other Unix tools were created through a similar process. Each program was created to fulfill some simple capability and was called a tool. The programs were designed to be fun to use and to be helpful to programmers. Among the principles guiding the tool-builders were:

“(i) Make each program do one thing well. To do a new job, build afresh rather than complicate old programs by adding new features.

(ii) Expect the output of every program to become the input to another, as yet unknown, program. Don’t clutter output with extraneous information. Avoid stringently columnar or binary input formats. Don’t insist on interactive input.

(iii) Design and build software, even operating systems, to be tried early, ideally within weeks. Don’t hesitate to throw away the clumsy parts and rebuild them.

(iv) Use tools in preference to unskilled help to lighten a programming task, even if you have to detour to build the tools and expect to throw some of them out after you’ve finished using them.”<sup>55</sup>

“Our goals throughout the effort,” write Ritchie and Thompson, in describing the research objectives of their work on Unix, “when articulated at all, have always been to build a comfortable relationship with the machine and to explain ideas and inventions in operating systems and other software.”<sup>56</sup>

Frederick P. Brooks Jr. is another computer pioneer who recognized the importance of pipes and combining single function programs into what he calls “a unified programming environment” to make the work of programmers more productive. Brooks describes Unix as one of “the first integrated environments to come into widespread use,” which “improved productivity by integral factors.”<sup>57</sup>

In explaining how such an environment functions, he writes: “They attack the accidental difficulties that result from using individual programs together, by providing integrated libraries, unified file formats, and pipes and filters. As a result, conceptual structures that in principle could always call, feed, and use one another can easily do so in practice.”<sup>58</sup>

Following the development of single function software tools which could be used together via pipes in a programming environment, came the development of whole tool-chests and workbenches of tools to rationalize the work of programmers.

## **Open Source Code and the Early Development of Unix**

Since the Unix programming environment was used by the researchers themselves, they came to learn its weaknesses and problems and were encouraged to make needed improvements. Contributing to the value of Unix during its early development, was the fact that the source code was

open and available. It could be examined, improved, and customized. In an article “Can UNIX Survive Secret Source Code?”, Mike Lesk, then a Bell Labs researcher, observes how only when computer source code is open and can be modified, will it be developed and vibrant. He gives the example of COMIT, the string processing language. At one point, its owners decided there would no longer be any modifications in the code, and so only distributed it in binary form. “You haven’t heard of COMIT since,” he notes. He describes how the same fate befell TRAC, a language close to FORTH. “Software is more attractive to hackers,” Lesk maintains, “if it can be changed. As more and more UNIX suppliers restrict access to source code, fewer and fewer advanced research shops will be attracted to UNIX.”<sup>59</sup>

Commenting on the importance of open source code in the early years of Unix development at Bell Labs, Thompson and Ritchie write, “Because all source programs were always available and easily modified on-line we were willing to revise and rewrite the system and its software when new ideas were invented, discovered, or suggested by others.”<sup>60</sup>

Not only was the source code open and available to the Bell Labs researchers developing the system, but the Labs also made the sources available on tape at the request of academic colleagues. Robert Fabry of the University of California at Berkeley was able to get a tape of Version 4 Unix and that began the long and important role played by faculty and students at the University of California at Berkeley in the development of Unix.<sup>61</sup>

Source code tapes were made available to interested researchers in the academic and computer science community for a nominal fee. For example, when John Lions, a Professor of Computer Science at the University of New South Wales in Australia read a paper published by Thompson and Ritchie in mid 1974, he wrote them for a copy of the Unix tape. After signing a license agreement with the Labs, and a token payment of \$110 Australian (\$150 U.S.), the Unix Edition 5 tape and manuals arrived, “as a late Christmas present,” in December, 1974, remembers John Lions.<sup>62</sup>

While Bell Labs made the tape and manuals available, they did so with no support. Berkley Tague explains the release was “‘caveat emptor’ – i.e. dollars on the table up front and no support promised.”

Henry Spencer, a Unix pioneer from the University of Toronto in Canada, and one of the programmers of C News, describes how early users of Unix in the academic community had to provide their own support. He explains: “It was very common at the time. This was in the days when UNIX was still treated by the Bell System as, ‘Oh, just something we happen to do for our own internal use. You can have a copy if you want, but if you got problems, don’t bother us.’ And the result was if you wanted UNIX support you did it yourself or it didn’t happen.”<sup>63</sup>

Lions agrees, “We needed help,” he notes, “but we couldn’t get any from outside sources so we ended up generating our own expertise.”<sup>64</sup> Not only did those working on Unix implementation at the University of New South Wales have access to the code, but Lions explains how Ian Johnstone, the tutor working with him in his Operating Systems class, suggested making some of the code for the Unix kernel available to the students in his class. “I think it was in 1975,” remembers Lions, that Ian Johnstone asked, “‘Why don’t we run off a few of the source files for the kernel and ask the students to take a look at them? Then we can ask them some questions; maybe it will be interesting’.”<sup>65</sup> Lions took Johnstone’s suggestion and made some of the Unix source code available to his class, but his students complained. They felt they needed to see the source code for

the whole kernel in order to make sense out of any part.

Taking their suggestion, Lions decided to make a large part of the source code available to his class. “The following year,” he recounts, “I prepared a booklet containing the source files for a version of Edition 6 UNIX that could then run on a PDP-11/40 system.” Lions followed the book of source code with a book of “explanatory notes that was intended to introduce students to the code.”<sup>66</sup> Lions explains that working on his book, “A Commentary on the UNIX Operating System” was a real learning experience. By slowly and methodically surveying the whole kernel he notes, “I came to understand things that others had overlooked.”<sup>67</sup>

When he read the manual and wasn’t quite certain about his interpretation, Lions would read the code. Through this process, he was able to determine that the manual was “really quite accurate in its description of what a program actually does. “In the Thompson/Ritchie era,” he observes, “words were carefully chosen.”<sup>68</sup> Lions writes, “In our opinion, it is highly beneficial for students to have the opportunity to study a working operating system in all its aspects.”<sup>69</sup>

“Moreover,” he adds, “it is undoubtedly good for students majoring in computer science to be confronted at least once in their careers with the task of reading and understanding a program of major dimensions.”<sup>70</sup> Lions found that, “On the whole the authors of UNIX, Ken Thompson and Dennis Ritchie, have created a program of great strength, integrity and effectiveness,” which he urged his students to “admire and seek to emulate.”<sup>71</sup> Not only did students in Lions’ class read and study the Unix source code and Lions’ Commentary on the source code, but Lions describes how he sent more than 200 copies of his book to Bell Laboratories. Eventually, the Labs took over distribution of the book.

Tague relates how Lions’ book of commentary and the Unix source code were used at AT&T “as part of the documentation package for those who wanted to understand or modify the UNIX® source code that the USG [Unix Support Group] shipped.”<sup>72</sup> Even after Unix V6 had been replaced by Unix V7, Tague explains that Lions’ Commentary continued to be useful as an introduction to Unix. Tague writes: “It outlined the conceptual architecture, very clearly in the short form of the system before it had accreted all the minor changes and feature additions that disguised the original clarity of its structure. All new people were given a copy when they joined the USG. And I suspect most development groups did the same.”<sup>73</sup>

Pioneers like Henry Spencer describe how important it was to those in the Unix community to have the source code. He notes how having the sources made it possible to identify and fix the bugs that they discovered, “There is something the UNIX community has always been fairly strong on,” he explained during an interview, “admitting things you know don’t work about the software.” Even in the late 1970’s and early 1980’s, remembers Spencer, “practically every UNIX site had complete sources.”<sup>74</sup>

One of the early functions of Usenet, the early online community of Unix systems begun in 1979, according to Spencer, was to provide cooperative software support for the Unix community. He elaborates: “Well, for one thing, Usenet predated a lot of company bbs’s and the like. It was basically a cheap way to hear about things fast and this was at a time when practically every UNIX site had complete sources and so a bug report often came with a fix. It was a way of finding out what people had discovered and what fixes they’d worked out for it. Quickly and easily. And for that matter, if you ran into something that you couldn’t solve yourself, putting out an inquiry to a bunch of fairly bright people who were fairly familiar with the code, often got a response, ‘O Yeah. We

solved that one' or 'You're right. There's a bug. Here's how to fix it' or sympathy even if no one had a fix for it."<sup>75</sup>

Another Unix pioneer, Dick Haight, corroborates the important role open source code played for those in the Unix community: "That by the way, was one of the great things about UNIX in the early days: people actually shared each other's stuff. It's too bad that so many sites now have purposefully turned off the read privileges in order to keep their ideas from being stolen. Not only did we learn a lot in the old days from sharing material, but we also never had to worry about how things really worked because we always could go read the source. That's still the only thing that matters when the going gets tough."<sup>76</sup>

Unix was continually used and improved by its creators. A growing community of programmers, system administrators, and users, both at Bell Labs and at academic and research sites outside the Labs, also used, developed and debugged the code. The fact that the source code was open and available made this possible. The result was the creation of Unix as a powerful and flexible programming environment.

Though Unix was primarily designed to "help build research software," Bell Labs software developers soon found that, "what worked well in the programming laboratory also worked well on modest projects to develop minicomputer-based systems in support of telephone company operations."<sup>77</sup>

## **AT&T Automates Internal Operations**

In the late 1960s, there had been an upsurge in demand for telephone company service which taxed all AT&T's resources.

The 1970 annual report of AT&T described the crisis: "The company concluded that the convergence of a number of adverse factors contributed to the crisis, including an unforecasted rapid upsurge of demand for business traffic on a poor switching system, faulty trunk maintenance and administration, inadequate training of maintenance personnel and poor main distributing frames administration."<sup>78</sup>

Section 1 of the U.S. Communications Act of 1934 charges the Federal Communications Commission (FCC) with promoting a "rapid, efficient, nationwide and worldwide wire and radio communications service with adequate facilities at reasonable charge."<sup>79</sup> This legislative obligation put pressure on AT&T to create and support the Bell Labs efforts to push forward the forefronts of computer science, since the most advanced technology and science were needed for AT&T to meet its obligations as a regulated communications utility.

Under pressure from both the New York Public Service Commission (NYPSC) and the FCC, AT&T had to find a way to upgrade their service.<sup>80</sup> In response to these requirements, a push began within AT&T to "use the computer to support the operation of the Bell System Network."<sup>81</sup> Berkley Tague, who would be instrumental in implementing this change at AT&T, describes the broad scope of the effort that AT&T instituted to deal with the service crisis. He lists: "Monitoring and alarms, maintenance and staff management, inventory and order control, revenue data collection and billing, traffic measurement and control, circuit provisioning,<sup>82</sup> etc." and he provides the following illustration: "The data that has to be collected to bill for calls was being stored on a variety of media – e.g. paper AMA tape, a punched paper tape a few inches wide, IBM compatible mag tape, and probably a few others. These tapes had to be mounted, saved, trucked to DP centers for processing,

etc. They could be lost or damaged in the handling. An obvious solution was to send the data over data-links – our own network – to the DP centers. Minicomputer-based systems were designed to collect, preprocess and transmit this data in some cases; in others, the electronic switches themselves were programmed to do the job. The goal was to eliminate the need for people handling physical media – a process that is both expensive and error-prone.”<sup>83</sup>

Software was needed to facilitate these changes. Tague explains how the Unix system, originally created for computer research, also served the needs of program developers. “The difference,” he explains, distinguishing between the needs of researchers and the concerns of developers involved “support, stability and reliability.” He describes how developers have to meet deadlines and so only tolerate absolutely necessary changes to the system. A researcher, on the other hand, according to Tague, will “tolerate a fair amount of upset and change and revision if it advances the toolkit significantly, even if it means you must recode your work.”<sup>84</sup>

The use of Unix by developers for support operations began in the 1972 period. Tague describes how “a number of developers were planning to build their own operating systems – typically their first system and often their first major programs.”<sup>85</sup> Having experience with Unix, he advocated Unix for the job. It was an uphill battle to convince developers to adopt Unix when there was no offered support and it was made up of 12,000 lines of undocumented code.

In 1973, Tague pointed out the need for a central support group and volunteered to form the group. He was given the job. The goal of the USG which Tague created in September 1973 was to “provide the stability and support that would buffer developers from the experiments that researchers were daily imposing on the evolving UNIX system.” He describes how Dennis Ritchie promised him a portable version of Unix, which was delivered as promised by October, 1973.<sup>86</sup> Recoding the Unix kernel from assembler language, which was specific to each computer, into the C programming language, was an important feat. Once the Unix kernel was coded in C, it would become possible to implement Unix on different computers without having to rewrite the majority of the code.

The effort to make Unix machine independent was desirable, so that AT&T wouldn’t be dependent on any one vendor for its computer hardware or software or service. He explains, “A goal of my effort was to remove the vendor lock so we could get competitive bids on the volume buys when we deployed these mini-based systems across the Bell System.”<sup>87</sup>

Every project adopting Unix, Tague reports, added their own features and function to it and it became the responsibility of the USG to choose among or merge the variations. The University of California at Berkeley, according to Tague, performed a similar job with the variations of Unix developed by academic computer scientists outside of AT&T.

## **Creating a Programmer’s Workbench**

While Tague was working on Unix Support Operations, other Bell Labs researchers were planning another major project that would impact on software development and on the use of a new form of tool.

Rudd Canaday (who had designed the original Unix file system with Thompson and Ritchie), Dick Haight, Ted Dolotta, John Mashey, and others formed the Programmer’s Workbench Group (PWB) within the Business Information Systems (BIS) project.<sup>88</sup> Tague explains that the “BIS problem was to get a common ‘workbench’ that would drive code onto any of the three or four commercial vendor’s mainframes that was used by BIS.” Tague writes that “By putting a UNIX®



system in front of the large mainframe systems, developers got the advantages of UNIX® in developing code and a place they could store debugged standard command sequences that drove development and testing on the target mainframes.”<sup>89</sup>

David Nowitz, a Unix pioneer who helped with the development of UUCP, and then went on to collaborate in the creation of honeydanber uucp, explains that when he hired into Bell Labs in 1969, there were programmers working on many different computers with different languages.<sup>90</sup> The problem these different computers represented for those at AT&T, is described by Haight: “We had a real problem to solve at the time. For one thing, we had a fairly large group of software developers at the Labs working on several different main frames. The biggest group, of course, consisted of people working on the IBM 360s or 370s. The programmers working on the IBM 360 or 370 had to contend with batch processing and IBM’s Job Control Language (JCL) problems. Other programmers were working with the UNIVAC and another group with Xerox computers. Only a few were using early but expensive time sharing like TSO (IBM’s timesharing) and Univacs Remand Service.”<sup>91</sup>

“These systems,” Haight relates, “not only offered very different programming environments, but proved to be very expensive to use and very unfriendly.” Describing the task they were charged with, Haight continues, “Basically, we ended up trying to give all these people a cheap text-editing front end that they could use for interactive program entry. For that to happen, we needed to develop a programming environment that was consistent with what they already were using.”

They developed a software subsystem which included five components that all of the programmers working on the different computers could use. The subsystem which they created was called the PWB. It included RJE (Remote Job Entry), an editor, a program for tracking version changes, a word processor and software to simulate test input.<sup>92</sup> Haight believed that Unix was “the best program development system around” and that this “development environment notion was the very concept that got PWB going back in '73.”<sup>93</sup>

The vision of the PWB/UNIX was conceived of in mid April 1973 and installed on the first workbench machine in October 1973 on a PDP 11/45. By 1977 the PWB was serving 1000 users.<sup>94</sup> The programmers working on the PWB rejected the top down software design philosophy, i.e. of creating a fully integrated facility and implementing it, and then making it available to users. Instead they adopted a bottom up philosophy. They designed independent single function tools and made them available to users as early as possible. Users were encouraged to provide feedback. The PWB programmers had the actual feedback to help them determine what modifications were needed to improve the tools. Evan Ivie, a Bell Labs researcher who was one of the creators of the PWB, describes why the PWB/UNIX group adopted the bottom up philosophy of software design: “The idea of designing and building a complete and fully integrated workbench system was rejected for a number of reasons, not the least of which is the fact that no one knows what that system should look like at this point in time. Instead, every effort was made to identify some of the immediate needs of potential users and to develop pieces of the workbench that could satisfy those needs quickly. This approach provided the workbench designers with much valuable user feedback quickly, and it allowed the projects to start picking up pieces of the Workbench to satisfy their most critical needs immediately and thus to start a phased transition to the complete workbench.”<sup>95</sup>

The design for implementing Unix on the IBM System/370 was done in 1979, coding was done in 1980, and by early 1981, the first production system IBM 3033AP was installed at Bell Labs

at Indian Hill. Unix had been suggested as the development system for programmers for the switching system software for the 5ESS switch because the Programmer's Workbench provided "editors, source code control and software generation systems."<sup>96</sup>

In 1981, after a production Unix system was running on an IBM 3033AP in Indian Hill, the application software tools were ported from the PDP-11/70 computer to the 3033AP. Shell scripts were carried over and then users were transferred.<sup>97</sup>

By 1984 the largest user of the Unix system for the IBM System/370 was the development project for the 5ESS Switch. The creation at AT&T Bell Laboratories of software for the 5ESS switch required millions of lines of code.<sup>98</sup> The Unix operating system with the Programmer's Workbench software was chosen for the 5ESS switch software development project because the researchers felt that Unix "provided a better development environment than any other operating system available. In addition, the developers were all trained in using this system and all the software tools had been developed."<sup>99</sup>

The elaborate process of applying Unix to the IBM 3033AP is described by the researchers: "Over 300 tools, written in both C and shell command language, were identified and examined. After careful study, almost half of the tools were found to be little-used and were eliminated for porting to the 3033AP. The C programs required recompiling to generate objects that would run on a 3033AP; in general, they compiled without problems. The shell scripts were carried over with almost no problems. (After testing) System testing, which consisted primarily of generating the system software for the 5ESS switch was then done. The effort to port the application tools was small and again proved the strength and computer independence of the UNIX operating system and the associated application programs."<sup>100</sup>

The rationale leading to the creation of the Programmer's Workbench, is outlined by Ivie. He wrote: "Although the computer industry now has some 30 years of experience, the programming of computer-based systems persists in being a very difficult and costly job. This is particularly true of large and complex systems where scheduled slips, cost overruns, high bug rates, insufficient throughput maintenance difficulties, etc., all seem to be the rule instead of the exception. Part of the problem stems from the fact that programming is as yet very much a trial and error process."<sup>101</sup>

There are at this point, he observed, "only the beginnings of a methodology or discipline for designing, building and testing software. The situation is further aggravated by the rapidly changing hardware industry and by the continuing evolution of operating systems which continues to nullify much of the progress that is made in the development of programming tools. What can be done," he asked, "to move the programming industry toward a more professional and stable approach to software development?"<sup>102</sup>

After enumerating several of the alternatives, he explained that his proposal involved "a very different approach to improving the development process." His recommendation was "...that the programming community develop a program development 'faculty' (or facilities) much like those that have been developed for other professions (e.g. carpenter's workbench, dentist's office, engineer's laboratory)." Such an approach, he explained, "would help focus attention on the need for adequate tools and procedures; it would serve as a mechanism for integrating tools into a coordinated set; and it would tend to add stability to the programming environment by separating the tools from the product (the current approach is equivalent to carpenter's leaving their tools in each house they build)."<sup>103</sup>

The Business Information Systems Programs (BISP) area of Bell Labs which was to use the Workbench program had among the computers it used: an IBM 370/158 and 168, a Univac 1108 and 1110, two Xerox Sigma 5's and several minicomputers.<sup>104</sup> The Workbench encouraged the development of machine independent programming tools. "Each tool must now function," Ivie explained, "for programmers developing code for a number of different vendor machines. There is no room for shortcuts which are dependent on the idiosyncrasies of a given machine (or a given project.) One is thus forced into a more stable and generalized software development approach which should be more applicable to new machines."<sup>105</sup>

Summing up the rationale of the Programmer's Workbench philosophy, Ivie wrote that "the programming profession has yet to produce a software development methodology that is sufficiently general so that it can be transferred from one project to another and from one machine to another." With the development of a Programmer's Workbench, he hoped, "a machine dedicated to the software development and maintenance function can serve as a vehicle for the development of such a methodology."<sup>106</sup>

## **Rationalizing the Process of Software Production**

In their article published in *Scientific American* in 1966, Fano and Corbato noted that the creation of a working computer in 1944 was followed by a series of quantum leaps in both the speed of the electronics and the development of special languages which facilitated human-computer communication. In another article in the same special issue of *Scientific American*, John McCarthy identified the importance of being able to program for making use of the computer. He also noted that spending time mastering a programming language was less important than understanding the functions the software is to serve. Their descriptions demonstrate that the computer is an all-purpose machine that can be directed to do whatever tasks humans can conceive of programs to describe. This poses a dilemma. Since humans are still needed to create the directives for the computer, the challenge raised is how can the process of programming be made more rational and productive.

Perlis had observed that the technical progress of a society was marked by the development of tools and the ever developing set of people capable of wielding those tools. He noted the continuing evolution of the process of producing tools. In the evolution of Unix, the need for software tools became evident and a series of tools was developed to meet these needs. The creation of time-sharing and the development of Unix as a time-sharing system has helped to create a community of people who not only use Unix, but who also continue its development. The contributions to the evolution of Unix by researchers at Bell Labs and then by those at the University of California at Berkeley, and other users and programmers around the world who continued the development of Unix, are an important achievement in the development of the process of software creation.

Writing in 1975, one of the team of researchers creating the PWB, P. J. Plauger, observed that "building on the work of others is the only way to make substantial progress in any field."<sup>107</sup> Describing the process of creating software tools, Plauger explained that each tool is a small programming project requiring anywhere from a few man months to a few hours to complete. He proposed that these tools need to be designed in a standard way so that they can interface with and be combined in a useful way with other tools for a variety of different purposes. "The programmer working this way becomes a tool builder, and his impact extends far beyond the task he may

originally have set out to solve,” he concluded.<sup>108</sup>

Placing the creation of software tools into the broader context of tool production provides a basis to evaluate the achievements and problems of this new stage. Sigvard Strandh, in the *History of the Machine*, describes how mechanical hand tools had acquired a stable form at an early stage. “From the Middle Ages until the era of industrialism,” he writes, “the tools of the carpenter, joiner and smith remained more or less the same.” With the publication of the *Great French Encyclopedia* (between 1751 and 1780), by Diderot and d’Alembert, however, Strandh observes that the development of these tools was given a substantial boost. Publication of the descriptions of the tools used in all the trades extent at the time, in great detail, in the *Great French Encyclopedia*, made it possible for those interested to learn how to build the tools, “so as to establish himself in one of the trades described.”<sup>109</sup>

The publication in the *Great French Encyclopedia* made possible the constant improvement of the old hand tools, and thus made it possible to manufacture the new tools needed for the new technical developments. “Industrialization,” Strandh writes, “means the manual methods of manufacture are mechanized, that is, taken over by machines. The machine tool is the local outcome of the process. Mass manufacture of machine tools,” which he writes, “has been called ‘the industry of industries’, didn’t become widespread until the 1870’s.”<sup>110</sup>

The conditions described by Strandh made possible advances in the technology of tool production in the 19<sup>th</sup> Century. These conditions provide a helpful perspective when examining the achievements of the Unix community in developing the software tool creation process. Bell Labs researchers early on recognized that software production required text editing in place of soldering, and that this substitution represented a significant challenge. The creation by Bell Labs programmers of software tools, making it possible to produce better software more efficiently, is an important development. This development is similar to the ability to improve hand tools that Morgan documents as the technological advance represented by the invention of iron. The success of writing the code for the 5ESS switch at AT&T demonstrates the achievement these software tools have made possible.<sup>111</sup>

Generating software is different from creating hardware and it presents a particular challenge. You can’t see software tools or touch them or handle them as you can mechanical tools. “Software is invisible and unvisualizable,” Brooks points out, since “the reality of software is not inherently embedded in space.” Therefore, he observes, “In spite of the progress made in restricting and simplifying the structures of software, these structures remain inherently unvisualizable and thus do not permit the mind to use some of its most powerful conceptual tools. This lack not only impedes the process of design with one mind, it also severely hinders communication among minds.”<sup>112</sup>

Therefore, there is a particular problem posed when creating, describing, or representing software tools. Diderot was able to catalog the tools and industrial processes of his times, but such a task is much more difficult when it comes to creating such an encyclopedia or presentation of software tools, or in creating the software component catalog that McIlroy once proposed. The development of UUCP and then of Usenet, as a means of cheap communication among those of the Unix community, made it possible to surmount some of the difficulties inherent in the process of software production.<sup>113</sup>

The Usenet community was able to pioneer the creation of a cooperative network that helped facilitate communications among the Unix software community. “Though large institutions have

been able to avail themselves of communications networks such as ARPANET, the UNIX community has made inexpensive electronic communication available to all its members via Usenet,” remarked John Stoneback, a faculty member at Moravian College, in an article describing the important role that Unix played for members of the academic community.<sup>114</sup>

Describing the importance of the creation of the Unix communication program UUCP, McIlroy writes that “before uucp remote login was the only kind of electronic connection between Unix systems. With uucp, every PDP 11 with dial-up (not even dial-out) access became connected. E-mail and file transfer previously available only among the fortunate few sites on ARPANET, were democratized overnight. This was the brainchild of Mike Lesk.”<sup>115</sup> Lesk’s original plan in creating UUCP was to create an automatic software updating system.<sup>116</sup> That, however, was fraught with difficulties, and instead, with modifications to UUCP by David Nowitz, it became the basis for a broad ranging online computer network that came to be called Usenet. McIlroy, describing how uucp was used by Tom Truscott and other Usenet pioneers to create an online community, writes, “Soon Truscott saw that the new medium could be used for automatic dissemination of information, and netnews was born.”<sup>117</sup>

The Unix system is based on a methodology that helps the user to learn the new language. The methodology is similar to the way children learn to talk. As Unix pioneers Brian Kernighan and John Mashey, write, “The code that people see, adapt, and imitate is usually well structured. People learn to code well in the same way that they learn to speak their native language well by imitation and immediate feedback.”<sup>118</sup>

“People often look at each other’s code,” they explain, “comment on it in person and through interuser communication facilities, and take pieces of it for their own use. The ideas of programming teams and egoless programming fit into the Unix environment well, since they encourage sharing rather than isolation.”<sup>119</sup> Given the frequent sharing and building on each others code, though, “some programs have always been ‘owned’ by one or two people, many others,” however, they note, “have been passed around so much that it is difficult to tell exactly who wrote them....” Also, they point out that “Tools have been built to automate many common programming tasks.”<sup>120</sup>

Kernighan and Mashey explain how in its early days, up through 1974, “Unix best supported a single, cooperative, tightly coupled group of people on each machine.” By 1975, the PWB/UNIX made possible support for a larger number of users. They describe how there have been large groups of over 200 users working on different machines, but that users preferred “to share a machine so they could share procedures and data bases.”<sup>121</sup> They describe how the concept of software tools spread from the Unix Programming environment and PWB/UNIX to more generalized applications described by Kernighan and Plauger in their book *Software Tools* (1976), then to further applications that were ported to other machines, and then to the more generalized applications of the Lawrence Berkeley Laboratory (LBL).<sup>122</sup>

## **Documenting Technological Progress**

In writing about the role that technology plays in the development of society, Morgan observed that oral or written documentation helps to facilitate technological progress. Morgan described how the Greek poet Homer by documenting the civilization that existed prior to his times, provided future generations with a yardstick by which they could measure their own accomplishments.<sup>123</sup>

So, too, documentation of the achievements of the Unix pioneers and of the tools they created, is an important step to be taken by those wanting to understand the progress made in our times. When amateur scientists created the Royal Society in Great Britain in the 1660's, one of their early efforts was to invite Rev. Spratt into the Society and to support him to write a history of the documentation of technology from Homer to their own times. Diderot's work on the Great French Encyclopedia, in the years leading up to the French Revolution, played such a role summarizing the knowledge, including, most importantly, the summary of all the trades and industries with their tools existing up to his time in France.

Unix pioneers often emphasize that scientific progress can only be made by building on what has already been achieved. It is important, therefore, to recognize the legacy of how Unix made it possible to achieve programming projects involving millions of lines of code by creating a workbench of software development tools. The creation of a new form of tool requires a new form of tool production process and though such technology is only in its infancy, there are examples to be studied to know what we are building on. As Henry Spencer and Geoff Collyer recognized when they were working on increasing the performance of Netnews software, it was necessary to systematically record what was happening to be able to see what progress was being made, "To know how to get somewhere, you must know where you are starting from."<sup>124</sup> Similarly, it is important to systematically document the history of the significant developments of the software tool revolution, so that there will be a yardstick by which to measure the accomplishments and to determine the new questions to be explored.<sup>125</sup>

---

#### Notes for Chapter 9

1. Lewis Henry Morgan, *Ancient Society*, Chicago, 1877, p. 42.
2. Alan Perlis, *Introduction to Computer Science*, New York, 1972, p. 4.
3. *Ibid.*, p. 3.
4. *Ibid.*
5. See F. J. Fano and R. M. Corbato, "Time Sharing on Computers" in *Information*, San Francisco, 1966, p. 77. See also C. Strachey, "Time Sharing in Large Fast Computers," *Proc Int Conf on Info Processing*, UNESCO, June, 1959, pp. 336-341. See also Frederick P. Brooks Jr., *The Mythical Man-Month*, Reading, 1972, p. 146. Brooks describes how both E. F. Codd and C. Strachey reported work aimed at time-sharing debugging in 1959. According to Brooks, Codd's system, which emphasized more efficient input-output and interactive debugging was not actually implemented while Strachey's work was built on by Corbato and others at MIT.
6. Early time-sharing systems were developed at Bolt, Beranek and Newman (BBN), a company then specializing in acoustical research, at MIT's Research Laboratory of Electronics (RLE) and at System Development Corporation (SDC). See Fano and Corbato, *Ibid.*, p. 78.
7. *Ibid.*
8. *Ibid.*, p. 94. They go on to explain: "Indeed, the analogy is not farfetched; an editorial board representing the community of users acts as a referee to pass on all new commands that are to be introduced into the system and on all information that is to be stored in the public files."

9. Ibid., p. 95.

10. Ibid.

11. John McCarthy, "Information" in *Information*, San Francisco, 1966, p. 1.

12. Ibid., p. 4.

13. Ibid., p. 8.

14. Ibid., pp. 15-16.

15. Doug McIlroy notes that, "The Multics planners already had seen remote access prototyped at several places, including CTSS and some commercial vendors. Their bigger vision was of an 'information utility,' not just a remote-access system that allowed economies of scale in computer cycles." From e-mail correspondence July 30, 1995. Documenting the need for new hardware as well as software, E. L. Glaser, J. F. Couleur and G. A. Oliver, in "System Design of a Computer for Time Sharing Applications," explain: "Investigation proved computers that were immediately available could not be adapted readily to meet the difficult set of requirements time-sharing places on any machine. However, there was one system that appeared to be extendible into what was desired. This machine was the General Electric 635." (Proc-FJCC, AFIPS, vol. 27, part 1, November 30, 1965, p. 197.)

16. In their article, "Introduction and Overview of the Multics System," F. J. Corbato and V. A. Vyssotsky explain: "For several years a solution has been proposed to the access problem. This solution usually called time-sharing, is basically the rapid time-division multiplexing of a central processor unit among the jobs of several users, each of which is on-line at a typewriter-like console." (Ibid., p. 186)

17. Ibid., p. 185.

18. "Automating Telephone Support Operations: An Interview with Berkley Tague," *Amateur Computerist*, vol. 6 no. 1, Winter/Spring, 1994, p. 8.

19. Ibid.

20. "Interview with Victor Vyssotsky," *Unix Review*, January, 1985, p. 59.

21. Ibid.

22. Ibid.

23. Ibid., p. 60.

24. Dennis M. Ritchie, "The Evolution of the UNIX Time-sharing System," *AT&T Bell Labs Technical Journal*, vol. 63 no 8, part 2, October, 1984, p. 1578. (Hereafter, "Evolution")

25. Dennis Ritchie, "The Development of the C Language", ACM, presented at the Second History of Programming Languages Conference, Cambridge, Massachusetts, April 1993, p. 1.

26. "Interview with Victor Vyssotsky", p. 60.

27. D. M. Ritchie, "UNIX: A Retrospective", *The Bell System Technical Journal*, vol. 57 no. 6, part 2, July-August, 1978, p. 1950. Ritchie refers to the article by R. J. Feirtag and E. I. Organick, "The Multics input-output system," Proc. Third Symposium on Operating Systems Principles (October 18-20, 1971), pp. 35-41.

28. "Evolution," p. 1580.
29. "The Development of the C Language," p. 2.
30. Ibid.
31. Ibid.
32. "Evolution," p. 1580. Also, there is an interesting account by Ritchie of the difference between the work at Bell Labs with Unix as opposed to their work on Multics. (See "Evolution," p. 1587.)
33. E-mail correspondence from Doug McIlroy, July 30, 1995.
34. The BCPL text formatter had been inspired by J. Saltzer's runoff program for CTSS.
35. "Evolution," pp.1589-90. In the lobby of the main building at Bell Labs in Murray Hill, N.J. there is an exhibit of communication breakthroughs which credits Doug McIlroy and Rudd Canaday with the invention of pipes.
36. E-mail exchange with Doug McIlroy. When asked why he had proposed that pipes be introduced into Unix, McIlroy responded: "Data streams were a particularly clean notion of program composition that I'd talked about for a long time. I proposed them in an unclarified way for Multics in 1964. And Joe Ossanna actually wrote some elaborate IO switching for Multics, though I don't think it ever caught on. I had been much taken by Conway's idea of coroutines, which was published (in CACM?) around 1963, and was brought to high development by Dahl in Simula. I thought a lot about stream representations of algorithms. Indeed, the algorithm I am proudest of is the coroutine sieve of Eratosthenes that I described at Cambridge in 1968, and which is now one of the chestnuts of parallel computing." (from e-mail communication)
37. "Evolution," p. 1590.
38. "Interview with Victor Vyssotsky," *Unix Review*, January, 1985. "Co-routines had after all shown up in SIMULA by the end of 1967."
39. In August Mohr, "The Genesis Story," *Unix Review*, January, 1985, p. 5. Mohr explains, "Haight believes that the pipe facility has had a major effect on the evolution of UNIX. Because of the shell and the piping facility, programs running under UNIX can be conversational without being explicitly programmed that way. This is because piping adds to the shell's ability to accomplish this purpose for the system as a whole."
40. M. D. McIlroy, "Unix on My Mind", Proc. Virginia Computer Users Conference (Blacksburgh), vol. 21, September, 1991, pp. 1-6.
41. See M.D. McIlroy, "A Research UNIX Reader: Annotated Excerpts from the Programmer's Manual, 1971-1986". Computing Science Technical Report, No. 139, AT&T Bell Laboratories, Murray Hill, New Jersey, June 1987, pp. 9 and 43.
42. Account provided by Doug McIlroy
43. "A Research UNIX Reader," p. 24.
44. Account provided by Doug McIlroy.
45. See for example Frederick Brooks, "No Silver Bullets," *Unix Review*, November, 1987, p. 41 and also *The Mythical Man-Month*, Addison-Wesley, Reading, MA, 1982, for a helpful statement of the problem in programming research



that needed to be solved.

46. "Unix on My Mind."

47. E-mail correspondence from Doug McIlroy.

48. *The Unix Programming Environment*, Englewood Cliffs, New Jersey, 1984.

49. Brian Kernighan and Rob Pike, "Program Design in the UNIX Environment," in *AT&T Bell Labs Technical Journal*, October, 1984, p 1601.

50. *Ibid.*, p. 1596.

51. M.D. McIlroy, E.N. Pinson, and B. A. Tague, "Foreword", *Bell System Technical Journal*, vol. 57 no. 6, part 2, July-August, 1978, p. 1901.

52. *Ibid.*, p. 1899. See for example, the thunderclap in the Appendix.

53. *Ibid.*, p. 1902. See also Stephen C. Johnson, "UNIX: The Language Forms", in *USENIX Association Winter Conference Proceedings*, Washington, D.C., January 21-23, 1987, p. 18.

54. See Brian W. Kernighan and Rob Pike, *The Unix Programming Environment*, Englewood Cliffs, New Jersey, 1984, p. 200. "idiff is loosely based on a program originally written by Joe Maranzano. diff itself is by Doug McIlroy, and is based on an algorithm invented independently by Harold Stone and by Wayne Hunt and Tom Szymanski. (See "A fast algorithm for computing longest common subsequences," by J. W. Hunt and T. G. Szymanski, *CACM*, May, 1977.) The diff algorithm is described in M. D. McIlroy and J. W. Hunt, Technical Report 41, 1976. To quote McIlroy, "I had tried at least three completely different algorithms before the final one. diff is a quintessential case of not settling for mere competency in a program but revising until it was right."

55. "Foreword", p. 1902.

56. D. M. Ritchie and K. Thompson, "The UNIX Time-Sharing System," *The Bell System Technical Journal*, vol. 57 no. 6, part 2, July-August, 1978, p. 1926.

57. Fred P. Brooks, Jr., "No Silver Bullets", *UNIX Review*, November, 1987, p. 41.

58. *Ibid.*

59. Michael Lesk, "Can UNIX Survive Secret Source Code?", *Computing Systems*, vol. 1 no. 2, Spring, 1988, p. 189.

60. D. M. Ritchie and K. Thompson, "The UNIX Time-Sharing System," p. 1927.

61. Marshall Kirk McKusick, "A Berkeley Odyssey", *Unix Review*, January 1985, p 30.

62. See Peter Ivanov, "Interview with John Lions," *UNIX Review*, October, 1985, p. 51.

63. "Interview with Henry Spencer: On Usenet News and C News," *The Amateur Computerist*, vol. 5 no. 1-2, Winter/Spring 1993, p. 5.

64. "Interview with John Lions," p. 51.

65. *Ibid.*, p. 52.

66. Ibid.

67. Ibid.

68. Ibid., p. 53.

69. John Lions, A Commentary on the UNIX Operating System, The University of New South Wales, (nd), p. 5.

70. Ibid.

71. Ibid., p. 9. A short time later, in "Unix Implementation", *The Bell System Technical Journal*, vol. 57, no. 6, part 2, July-August 1978, Ken Thompson described how a goal in writing the Unix kernel was to write exemplary source code that others would emulate. Thompson writes: "What is or is not implemented in the kernel represents both a great responsibility and a great power. It is a soap-box platform on 'the way things should be done.' Even so, if 'the way' is too radical, no one will follow it. Every important decision was weighed carefully. Throughout, simplicity has been substituted for efficiency. Complex algorithms are used only if their complexity can be localized."(pp. 1931-32)

72. "Automating Telephone Support Operations: An Interview with Berkley Tague," *Amateur Computerist*, vol. 6 no. 1, p. 10.

73. Ibid. Also, Mike Blake-Knox describes how he did development work for Bell-Northern Research (BNR), which was the Bell Labs equivalent for Northern Electric (which is now Northern Telecom). He reports that he had a copy of Lions' book and used it to do Unix kernel modifications. (from e-mail correspondence Feb. 13, 1994)

74. "Interview with Henry Spencer," p. 4.

75. Ibid.

76. Marc Rochkind, "Interview with Dick Haight," *Unix Review*, May, 1986, p. 65.

77. M. D. McIlroy, E. N. Pinson, and B. A. Tague, "Foreword", *Bell System Technical Journal*, vol. 57, no. 6, July-August, 1978, part 2, p. 1902.

78. Alan Stone, *Wrong Number*, New York, 1989, p. 17.

79. Ibid., p. 145. "For example, since 1893, Bell maintenance employees had left unneeded jumper wires attached to main distributing frames [which connect each customer wire pair to the line terminal in a switching center] with no ill effect, but the growth in traffic caused the weight of dead jumpers to impede the operation of the main distributing frames."

80. During this period the NYPSC imposed standards requiring regular reporting from AT&T so that progress in upgrading its service could be monitored. Also in 1970 the FCC required monthly reports on the quality of AT&T service in 20 major cities and by 1972 in 72 administrative areas.

81. Tague situates this push as beginning around 1969.

82. "Interview with Berkley Tague," pp. 7-8.

83. Ibid., p. 8.

84. Ibid., p. 9.

85. Ibid.

86. Ibid. McIlroy in e-mail correspondence clarifies, “The conversion of Unix to C in 1973 was a step toward portability, but was certainly far from achieving it. It took more work to port Unix to the 360 than to make Unix in the first place. Not until 1978 did Richard Miller in Australia and Ritchie and Johnson at AT&T demonstrate effective portability.”

87. See S. C. Johnson and D. M. Ritchie, “Portability of C Programs and the UNIX System,” *The Bell System Technical Journal*, vol. 57 no. 6, part 2, July-August 1978, pp. 2021-2048.

88. The Programmer’s Workbench was first suggested by E. L. Ivie. Ivie, E. L. “The Programmer’s Workbench – A Machine for Software Development”, Unpublished Report, Bell Laboratories, May 19, 1975.

89. “Automating Telephone Support Operations: An Interview with Berkley Tague, *The Amateur Computerist*, vol. 6 no. 1, p. 10.

90. Uucp was first designed and implemented by Mike Lesk of Bell Labs in 1976. It has undergone many revisions and improvements. It was further developed by Lesk and David Nowitz and made available outside of the Labs as part of the V7 distribution. A new revision and improvement known as the Honeydanber UUCP was written in 1983 by Peter Honeyman, David A. Nowitz and Brian E. Redman.

91. “Interview with Dick Haight,” p. 55.

92. The subsystems that became the PWB included: 1) the ed line editor, 2) a modified RJE, 3) a piece to simulate test input to applications for the IBM and Univac and Xerox computers, 4) SCCS software tool for tracking program changes, and 5) the mm macros so that people could use the system for word processing.

93. “Interview with Dick Haight,” p. 63.

94. See “The Programmer’s Workbench” by T. A. Dolotta and R. C. Haight and J. R. Masey, in Proc. Second Int. Conf. on Software Engineering, October, 13-15, 1976, pp. 2180-2181.

95. Evan L. Ivie, “The Programmer’s Workbench – A Machine for Software Development” *Communications of the ACM*, vol. 20 no. 10, October, 1977, pp. 750-751. Ivie’s paper described the implementation of 5 basic components of the workbench: job submission, module control, change management, document production and test drivers.

96. W.A. Felton, G.L. Miller and J. M. Milner, “A Unix System Implementation for System/370,” AT&T Bell Labs Technical Journal, October, 1984, p. 1752.

97. Ibid., p. 1753.

98. Ian Johnstone, who had been the tutor at University of New South Wales working with Professor John Lions, was one of the researchers invited to Bell Labs. He managed the completion at AT&T Bell Labs of the port of Unix to the IBM 370 computer. See “Unix on Big Iron” by Ian Johnstone and Steve Rosenthal, *UNIX Review*, October, 1984, p. 26. Johnstone also led the group that did the port to the AT&T 2B20A multiprocessor system.

99. “A Unix System Implementation,” p. 1753.

100. Ibid., pp. 1763-4.

101. Evan L. Ivie, “The Programmer’s Workbench – A machine for Software Development”, *Communications of the ACM*, vol. 20 no. 10, October, 1977, p. 746.

102. Ibid.
103. Ibid.
104. Ibid., p. 747.
105. Ibid., p. 749.
106. Ibid.
107. P. J. Plauger, AFIPS Conf Proceedings, vol. 44, 1975, p. 281.
108. Ibid.
109. Sigvard Standh, *The History of the Machine*, New York, 1989, p. 79.
110. Ibid., p. 81.
111. See for example, articles in the *AT&T Bell Labs Technical Journal*, July-August, 1985, vol. 64 no. 6, part 2 on “The 5ESS Switching System” and “A UNIX System Implementation for System/370” by W. A. Felton, G. L. Miller, and J. M. Milner, in *Bell Laboratories Technical Journal*, vol. 63 no. 8, part 2, October, 1984, p. 1751.
112. Fred P. Brooks, Jr., “No Silver Bullets”, *The Unix Review*, November, 1987, p. 41.
113. See for example “On the Early days of Usenet,” Chapter 10 of this book.
114. John Stoneback, “The Collegiate Community,” *Unix Review*, October, 1985, p. 26.
115. Doug McIlroy, e-mail correspondence.
116. Mike Lesk, “A Proposal for Operating System Support.”
117. McIlroy correspondence.
118. Brian W. Kernighan and John R. Mashey, “The Unix Programming Environment,” *Computer*, April, 1981, p. 20.
119. Ibid.
120. Ibid.
121. Ibid.
122. Like Burroughs B1726. See C. R. Snow, “The Software Tools Project” in *Software - Practice and Experience*, vol. 8, September-October, 1978, pp. 585-99. See also “A Virtual Operating System” by Dennis E. Hall, Deborah K. Scherrer, and Joseph S. Sventek, in *Communications of the ACM*, vol. 23 no. 9, September, 1980, pp. 495-502.
123. Morgan writes: “The use of writing or its equivalent in hieroglyphics upon stone, affords a fair test of the commencement of civilization. Without literary records neither history nor civilization can properly be said to exist. The production of the Homeric poems, whether transmitted orally or committed to writing at the time, fixes with sufficient nearness the introduction of civilization among the Greeks. These poems, ever fresh and ever marvelous, possess an ethnological value which enhances immensely their other excellences. This is especially true of the Iliad, which contains the oldest as well as the most circumstantial account now existing of the progress of mankind up to the

time of its composition.” (Ancient Society, p. 31)

124. “Collyer kept detailed records of his work on ‘rnews’, so he could see how much progress he was making.” Geoff Collyer and Henry Spencer, “News Need Not Be Slow,” Winter 1987 USENIX Technical Conference Proceedings, Washington, D.C., p. 183.

125. Though there is a rich technical literature documenting Unix development in various technical journals, historical accounts are harder to find and often hard to acquire extent copies. Some of the helpful sources include “Unix Time-Sharing System” Special issue of *The Bell System Technical Journal*, vol. 57 no. 6, Part 2, July-August, 1978, and “The Unix System” special issue of *AT&T Bell Laboratories Technical Journal*, vol. 63 no. 8, part 2, October, 1984. Also, *Life with Unix* by Don Libes and Sandy Ressler, Prentice Hall, Englewood Cliffs, N.J., 1989. Numerous interviews and articles with or about Unix pioneers appeared in early issues of *Unix Review* during the 1980's. Also, the issue of *The Amateur Computerist* celebrating the 25th Anniversary of Unix, vol. 6 no. 1, Winter/Spring 1994 is available online. The URL is <http://www.columbia.edu/~hauben/acn/> and contains interviews with Unix pioneers John Lions and Berkley Tague, as well as an article on the development of Unix. The book *A Quarter Century of UNIX*, by Peter Salus, Addison-Wesley, 1994, contains references to a set of interviews done by Mike Mahoney from “The Unix Oral History Project: Release.0, The Beginning,” ed. and transcribed by Michael Mahoney as well as other interviews.

---

Thanks to Doug McIlroy, David Nowitz, John Lions, and Berkley Tague, for making available information that was helpful in doing the research for this article and thanks to Doug McIlroy, Alfred Aho, Mark Horton and Tom Truscott for commenting on earlier versions.

---

## Appendix

“Like a thunderclap from the sky, the pipeline slays the problem.”

Following is a script provided by Tom Truscott. Such particularly powerful pipelines have been referred to by Ken Thompson as “thunderclaps”.\*\*\*\*

To find how often each word appears in files named book.1, book.2, ...:

```
cat book* | tr -cs '[A-Z][a-z]' '\012' | sort | uniq -c | sort -nr | more
```

-----

Truscott notes “Alas, thanks to so many \*IX variants this pipeline doesn’t always work, on some the underlined text needs to be ‘[\012\*]’.”

# Chapter 10

## On the Early Days of Usenet: The Roots of the Cooperative Online Culture

by Ronda Hauben  
rh120@columbia.edu

“Without a historical perspective, it’s quite easy to get the wrong impression of how all this came to pass. It is the result of the work of a large number of individuals, some of whom have been at it for the last 20 years.”

Lauren Weinstein, 1990

“Even if we have shifted away from discussing human networks, we are getting a first hand EXPERIENCE of what they are through this mailing list. No amount of ‘a priori’ theorizing of their nature, has as much explanatory power as personal experience. By observing what happens when connectivity is provided to a large mass of people in which they can FREELY voice their ideas, doubts, and opinions, a lot of insight is obtained into very important issues of mass intercommunication.”

Human-Nets Mailing List, 03 June 1981, Jorge Phillips,  
Subject: administrivia

Usenet was born in 1979. It has grown from a design conceived of by two graduate students Tom Truscott and Jim Ellis, at Duke University in North Carolina, to a logical network linking millions of people and computers to over 9,500 different newsgroups and millions of bytes of articles available at any given time at hundreds of thousands of sites around the world. Yet little is generally known about how Usenet began and how it developed.

### Computer Chess - The Mini Slays the Mainframe

Tom Truscott had a dream. As a kid he had read the book *Danny Dunn and the Homework Machine*. He decided that it would be neat to have a homework machine. Some things caught his imagination and this particular goal not only set him on a course that would affect his future, but it also would have an unexpected impact on the rest of the world. By the summer of 1970, before his senior year in high school, Truscott had enrolled in a summer computer program that gave him his first chance to use a computer and to learn to program in BASIC. “My first large program played checkers,” he remembers of that summer.<sup>1</sup> “It didn’t play all that well,” he admits, but it introduced him to some of the power of computers. As a college freshman at Duke University the next year Truscott met another student in his chemistry lab who was an excellent chess player. Truscott describes how he told his chemistry lab partner Bruce Wright that “we could write a computer chess program that would beat Bobby Fischer.” Wright “didn’t think so, but we started writing the program anyway.” Truscott continues, “I was interested because of the computing challenge and no doubt the fame we would garner by defeating Fischer, and I guess Bruce was interested because he wanted to learn computing.” Truscott describes how the two undergraduates spent “a LOT of time” writing their chess program and in the process they learned a lot about how not to write programs.

Truscott was interested in how game programs were like robots since they functioned as autonomous creatures. “At tournaments,” he points out, “the program tells me what moves to make for it, asks me how much time it has left on the clock,” etc. And writing a software robot, Truscott observes, “is a lot easier than building a real one.”

Once Truscott and Wright had set their sights on creating a championship chess program, Truscott set out to research what work had been done on the problem. He found that Claude Shannon had written “a very early paper on how to construct a chess playing machine.”<sup>2</sup> “It was remarkably farsighted given the state of computing then,” Truscott remembers. The next oldest paper he found was from 1958 by someone who implemented a program similar to Shannon’s proposal. “It played terribly,” he recalls.<sup>3</sup>

Also, by Spring of 1974, Truscott had joined the Association for Computing Machinery (ACM) to receive notification of the computer chess tournaments. Reading through the journal *Communications of the ACM* in 1974, he came across an article about a new operating system created by research programmers at Bell Labs.<sup>4</sup> In the article, he noticed that a program created by a Bell Labs team ran in the background sopping up idle CPU time and solving simple chess end-games (for example King and Rook vs. King). Truscott explains how there was no chance he and Wright could do something like that on the mainframe computer they were using, since it cost 20 cents per second. But he notes that their mainframe was about the fastest there was and could compute rings around the DEC PDP-11 that the Unix operating system ran on.

He and Wright created their program for an IBM System 370 Model 165 MVT/TSO mainframe computer system at Duke. It had three megabytes of main memory, which Truscott notes was later upgraded to “4 megabytes for a mere \$100,000.” That was, according to Truscott, “Pretty much the top of the line at the time. We did our development in batch mode,” he remembers, “The source code was on punched cards and the compiled code was stored on disk.” And in tournaments, he and Wright used the IBM time-sharing mode TSO.

The first computer chess tournament Truscott and Wright competed in was the North American Chess Championships held at the ACM Annual Conference in San Diego, California in November 1974. By then, Truscott was in his senior year at Duke. He and Wright named their chess program *Duchess*.

Following is Truscott’s description of his first tournament and how he met one of the most respected programmers in the Unix community during that tournament. Truscott writes: “There were twelve teams competing in the tournament. We were on a stage in a large room with seating for spectators. Each team had a computer terminal (something like a dot-matrix printer with a keyboard in front and an acoustic modem on the back). And a telephone. Boy were those phone calls expensive. But the ACM was picking up the tab, and Duke was giving us the computer time.

At the 1974 tournament, we knocked off MIT’s *TECH-II* in the first round. They had come in second the previous year, and we were a newcomer, so that was something of an upset. In the second round we got clobbered by the perennial champ, *CHESS 4.0* from Northwestern University.

In the third round we played Bell Labs’ *Belle*. It was called *T. Belle* at that point. I had met the author earlier, before the second round, when he showed me how good his program was at solving mating problems. I wasn’t that interested in chess, but humored him while he pulled a chess position out of a library and had the program find a mate in 5 (or some such). I guess if I actually played chess I would have been impressed.

So when the third round began, Bruce Wright and I were on one side of a table, and Ken Thompson and someone else from Bell Labs (who years later I realized was Brian Kernighan), were on the other. I noticed that when Ken Thompson logged on, the Bell Labs computer printed: Chess tonight, please don't compute.

I mentioned that that was really neat to be able to get the comp center to put out a notice like that. He said something non-committal in response. So the game began. A few hours and a few thousand dollars later we really had Belle on the ropes. All it had left was a lone king and we were about to queen a pawn! But then our program ABENDED (core dumped) in a way that caused the phone line to drop. We dialed back in and set things up, same thing. Every so often it would actually make a move. But making the phone call was slow (we had to ask for an outside line from the hotel operator) and painful (rotary dial you know) and eventually our program lost on time."

After the tournament was over, Truscott and Wright examined what had happened and they observed that the problem was not with their program, but rather with a bug in the TSO operating system on their mainframe. "Thus was our mighty mainframe slain by a minicomputer," he admitted, as they had lost the competition because the operating system of their mainframe computer had proven inferior to the operating system of the mini-computer used by the Bell Labs Team. "But I didn't realize it was UNIX," Truscott recalls, noting that the victory went to the Bell Labs team and their mini-computer because of the power of the Unix operating system.

Truscott and Wright competed in every ACM Computer Chess Competition [CCC] from 1974 to 1980. The next time he met Ken Thompson was at the 1976 Unix Users Group meeting at Harvard. "That was great fun," he remembers. There were about 80 attendees. "Somewhere along the way I made the connection between Belle and Thompson and UNIX." By this time Truscott was a graduate student at Duke where he and others had just installed Unix Version 6 on the Computer Science Department computer.

"I was also at the 1978 UNIX Users Group meeting at Columbia University, and both Ken Thompson and Dennis Ritchie were there," Truscott continues, "Thompson also competed in the 1978 ACM CCC. He had some special chess hardware but it was no match for the much-improved mainframe programs."

"Because of our mutual interests," Truscott recalls, "Thompson would even call up our computer at Duke from time to time, and 'write' me. That was pretty intense, my trying to pick perfect sentences to send along to the genius at the other end. I think it was during one of those 'write' sessions in early 1979, that he asked if I would be interested in a summer job."

Truscott accepted Thompson's offer and spent the summer of 1979 at Bell Labs in Murray Hill, New Jersey, the birthplace of Unix. That Summer, a distribution of the Unix Operating System, Unix Version 7 was made available to sites with licenses from AT&T to use Unix. Included in the Unix V7 distribution were a number of Unix tools such as "sed" "awk" "uucp" and the Bourne Shell. These tools were very helpful and would prove invaluable in the creation of Usenet.

Truscott found that Bell Labs provided an exciting and supportive environment. Following is his account of this important summer in 1979 that he spent playing volleyball, eating pizza and working on a daily basis with many of the pioneers of the Unix community. He writes: "Woke up at 11am. Got to Bell Labs at noon so I could play volleyball out on the front lawn with Mike Lesk and Steve Bourne and other folks. After a few weeks, the security folks told us they couldn't have a regulated monopoly running around loose like that. Lunch at 1pm in the Bell Labs restaurant. Ken



Thompson and Dennis Ritchie and Greg Chesson were regulars. They had lunch at 1pm because sometimes they didn't get to work until then. Sometimes Dennis Ritchie would entertain us with some horror story about a non-UNIX system he dealt with recently...."

"At 2pm the day began, which involved doing pretty much whatever we wanted. Ritchie was working on 'streams'. I think Ken Thompson was working on typesetting software but mostly working on a chess machine.... Often at 7pm a group would go out for dinner (they liked pizza). Occasionally someone would host dinner at their home. Afterwards I would go back to the Labs and work until midnight. And the next day I would get up 'at the crack of noon' as Thompson put it."

As the summer ended, Truscott left Bell Labs and returned to Duke.

## **Using Unix to Create an Online Community**

Truscott, describing his return to Duke, writes, "Of course when the summer was over and I was back at Duke, one of the first things I did was arrange a uucp connection to research. They called us nightly, which was great." Truscott and Dennis Ritchie set up a uucp connection between "duke" a CS Department computer site at Duke in Durham, North Carolina, and "research" a computer site at Bell Labs in Murray Hill, New Jersey.

The uucp program that was part of the V7 distribution of Unix made it possible to send e-mail and files to other Unix sites using telephone lines as long as the sending computer had an auto-dialing modem and the receiving computer had an auto answering modem.

But these links did not make up for the fact that by Fall 1979, Truscott was back at Duke and no longer in the exciting environment of the birthplace of Unix. After having worked at Bell Labs for Ken Thompson where, as in Truscott's words, "I was in UNIX heaven the whole time, returning to Duke in the fall meant the end of that." Also, that summer he had attended the Unix User's Group meeting in Toronto, Canada. Once back at Duke the primary connection with the Unix community was through the USENIX newsletter; "Login:". This newsletter, however, hadn't appeared in a while. That Fall, another Duke graduate student, Jim Ellis installed the latest Unix (V7) edition on a Duke Computer Science computer. It broke many old programs, including a public domain 'items' program that had provided a local bulletin board. Truscott recalls how the program allowed items to be entered into one of several categories. "It had a number of problems," he explains, "including a 512 byte limit per item, so we were thinking about writing a completely new program. Then we could contribute it to the next user group tape and hopefully achieve some minor level of fame."

Truscott attributes the creation of Usenet to the confluence of these events in Fall 1979. He describes a long rambling conversation he and Ellis had one night considering these circumstances. The idea for Usenet developed during their discussion.

Soon afterwards, Truscott and Ellis met with two other local Unix enthusiasts, Dennis Rockwell, who was a graduate student and worked in the Physiology Department at Duke, and Steve Bellovin, who was a graduate student at the neighboring University of North Carolina (UNC) at Chapel Hill. They decided on the transfer format, i.e., on what an article would look like to make it possible to ship files via computers using uucp, and they agreed on the basic functionality of the software they would need to create an online network.

Bellovin wrote a shell script using Unix to test the design concept. Describing the early work to create Usenet, Bellovin writes: "The release of the uucp program with V7 UNIX provided the initial impetus. So did the Bourne shell. So the very first version of net news was a 3-page shell

script. It supported multiple newsgroups, cross-postings, and subscription lists implemented as environmental variables. As best as I can tell, this script has not survived.”<sup>5</sup>

Bellovin emphasizes how the ease of testing software design facilitated by Unix made it possible to create Usenet. “It’s worth noting now that given the speed (or lack thereof) of the machines we had we utterly relied on the ease of writing shell scripts to experiment with protocol variants. Compilation would have taken much too long.”

Commenting about the early plan for Usenet, Bellovin notes: “We estimated a maximum size of 100 sites, and 1-2 articles a day, net-wide...you couldn’t read things out of order. The goal there (and in many other spots) was to have software free of databases. Instead, we chose to let the file system do the work.”

Bellovin recalls why a news program to replace the one they had used with Unix V6 was needed. “Another motivation,” he writes, “was some sort of local news system. On V6, Duke and UNC had a local news system that came from somewhere. But articles were limited to 512 bytes, and we didn’t carry it forward to V7. A prime requirement was that there be an efficient way to test for the presence of news (hence the checknews program).”

The Duke and University of North Carolina graduate students hoped to contribute their news program to the Usenet community to be used with Unix V7. According to Truscott, the shell script was slow, but worked. They also decided on terminology such as ‘newsgroups’ to describe the subject areas they would have as part of their network. “That was probably due to the newsletter analogy,” he explains since “this was...before the PC and bulletin boards.”<sup>6</sup>

Stephen Daniel, another Duke graduate student, soon became involved and made a substantial contribution to the work. Truscott writes that Daniel “created the dotted newsgroup structure that we know today,” for the newsnaming scheme (i.e. NET.xxxx and dept.xxxx) Also, Steve Daniel wrote one of the earliest versions of the netnews software in the C programming language. This came to be known as “A-News”.

Truscott and Wright continued to participate in the Chess Tournaments and in 1980 they competed in the 3<sup>rd</sup> world Computer Chess Championship held in Linz, Austria. Thompson and Joe Condon, who was a researcher at Bell Labs, were also in the competition. Truscott notes that Thompson and Condon “had completed their hardware chess machine and snagged first place. Duchess came in third. And Claude Shannon was in attendance, and even handed out the trophies at the awards ceremony. Afterwards we all went over to a TV studio to watch a West German TV special on computer chess and the championship. Claude Shannon and his wife were very engaging people. Someone took a photo of all of us, I have a copy buried somewhere.”

When Usenet was created, the newsgroup NET.chess was created as one of the early newsgroups. By developing Usenet, the Unix community became the force behind the creation of an online community to welcome participants into the cooperative culture so important in creating Unix. Graduate students at Duke and the University of North Carolina were able to use Unix to create an online community to provide needed technical and social support. They later named this users network Usenet. The earliest map for Usenet was made up of the first two computers that were sites for Usenet:

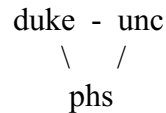
duke - unc

The sites were:

1) duke Duke University

2) unc      University of North Carolina at Chapel Hill

Another computer at Duke joined the network. The site was “phs”. It was in the Physiology Department at Duke Medical School. The map of Usenet then became:

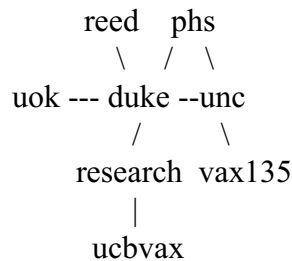


The third site was:

3) phs      Physiology Department of the Duke Medical School

Soon connections were set up with computers at Bell Labs. The computer site “research” and then “vax135” at the Labs were added to Usenet. In the summer of 1980, Mark Horton, a graduate student at the University of California at Berkeley, brought the computer site “ucbvax” onto Usenet.<sup>7</sup>

A map of Usenet during the Summer 1980 shows the sites then connected:



The additional sites were:

- 4) reed      Reed College
- 5) uok      University of Oklahoma
- 6) research    Bell Labs Murray Hill
- 7) vax135    Bell Labs Murray Hill
- 8) ucbvax    University of California at Berkeley

Bell Telephone Labs in Murray Hill, N.J. operating the computer named “research” was the first site to pick up the phone bills for calls between “ucbvax” at the University of California at Berkeley and “duke” at Duke University via “research.” Horton writes: “The first cross country link was from duke to research, then from research to ucbvax, all on research’s nickel.”<sup>8</sup>

Horton recalls how amazed he was to get e-mail messages from Usenet pioneers at Duke and the University of North Carolina just a few hours after he had sent them messages, thanks to the connectivity provided by the Bell Labs computer. “I remember,” he writes, “while at Berkeley, exchanging e-mail with the original ‘A-News’ developers and being amazed that I could get a reply back a few hours later, even though ‘research’ was polling both ‘duke’ and ‘ucbvax’ to pick up waiting mail.”

The first newsgroups on Usenet, according to Truscott, were known as NET.xxxx and dept.xxxx. After Horton joined Usenet, he began feeding mailing lists from the ARPANET into

Usenet. Mailing lists from the ARPANET fed into Usenet were identified as FA.xxxx newsgroups. Truscott notes that, “Only when ‘ucbvax’ joined the net, did ‘fa’ appear.” Truscott explains that he didn’t know about the ARPANET mailing lists until Horton joined Usenet.

At first the Usenet community could only read these ARPANET mailing lists, but couldn’t contribute to them. “It was a one-way gateway – ARPANET into Usenet only, done with recnews, as I recall,” writes Horton.<sup>9</sup> But at least it was possible for the Usenet community to follow the interesting discussions carried on via the ARPANET mailing lists during this early period of Usenet.

Bellovin explains why feeding the ARPANET mailing lists into Usenet was so important for the development both of Usenet and of the ARPANET. “Actually in my opinion,” Bellovin writes, “one of the key elements in the early growth of Usenet was when Mark Horton started feeding the SF lovers and human-nets mailing lists into newsgroups. Those provided a critical mass of traffic and served as a lure to attract new sites.” He describes how “The ARPANET was supposed to be a self-contained entity, and only approved sites were allowed to connect.” Therefore, the connection between Usenet and the ARPANET broke important new ground. Bellovin writes, “Mail to and from Usenet only sites, was an interesting test case that \*wasn’t\* stamped out, though I think it skated on some very thin ice for a while.”

The “ucbvax” site at the University of California at Berkeley provided a crucial gateway between Usenet and the ARPANET. The University of California at Berkeley could provide the gateway because it was also a site on the ARPANET. The CS Department vax computer (csvax) became the site “ucbvax” on the UUCP network. An internal network Berknet was set up to connect “ucbvax” on the UUCP network to “Berkeley” on the ARPANET. Horton explains that Professor Michael Stonebraker and Professor Domenico Ferrari, who were doing research to develop the Ingres data base, had a pair of machines (ing70) and (ingvax) which were sites on the ARPANET. They allowed Horton to use these machines for Usenet. Ing70 was the site known as “Berkeley” on the ARPANET. Horton and two other graduate students, Eric Allman and Eric Schmidt, set up the gateway between Usenet and the ARPANET and made it work. Schmidt created the local net, Berknet, to connect the ARPANET and the UUCPnet. The ARPANET and UUCP computers were tied together by Schmidt’s Berknet. The path, Horton explains, went:

“Any ARPA machine to Berkeley via ARPANET mail  
Ing70 (aka Berkeley) to csvax via Berknet  
ucbvax (aka csvax) to any UUCP machine via UUCP.”<sup>10</sup>

## Human-Nets and WorldNet

The Human-Nets mailing list [known on Usenet as the newsgroup FA.Human-nets] provided a mass of interesting posts to attract Usenet readership at a crucial period in Usenet’s development. The mailing list Human-Nets, Truscott remembers, was a mailing list from the ARPANET for discussing the implications of world-wide ubiquitous networking. “This network of the future,” he recalls, was referred to as WorldNet. “It was a very interesting mailing list and possible only due to the ability of the network itself to permit those interested in this obscure topic to communicate.”<sup>11</sup>

A directory of the ARPANET mailing lists maintained at MIT during this period lists each of the mailing lists. Describing Human-Nets, it notes that this mailing list “has discussed many topics, all of them related in some way to the theme of a world-wide computer and communications network usually called WorldNet. The topics have ranged very widely from something like tutorials,

to state of the art discussions, to rampant speculations about technology and its impact.”<sup>12</sup>

An article on Usenet in October 1982 about Human-Nets explained that “one reader expressed a wish for a ‘World Net’ to tie all sorts of computers worldwide together.”<sup>13</sup>

Another article in October 1982 described how WorldNet “was a nice idea to dream about”, but the writer was pessimistic that it could ever be implemented, at least within the next 10 years. He acknowledged, however, “Still, it’s a fun idea to think about,” and advised, “Maybe it should be tried on a smaller scale first (a distributed network of students with PCs at a university, perhaps a small city, or large community.) Who knows,” the poster observed, “with a PC in almost every home in a few years, maybe it’ll be possible and desirable.”<sup>14</sup>

One of the moderators of Human-Nets maintained how important it was to participate in such online discussion for those interested in developing ubiquitous world wide networking. Responding to a departing moderator’s complaint that the discussion on the list had diverged to a variety of topics, the new moderator disagreed. He retorted: “Even if we have shifted away from discussing human networks, we are getting a first hand EXPERIENCE of what they are through this mailing list. No amount of ‘a priori’ theorizing of their nature, has as much explanatory power as personal experience. By observing what happens when connectivity is provided to a large mass of people in which they can FREELY voice their ideas, doubts, and opinions, a lot of insight is obtained into very important issues of mass intercommunication.”

“The fact,” he continued, “that...dissimilar...topics have been discussed in our own instance of a human network says a lot about its nature and the interests and nature of its members and should not be considered as detracting from the quality of the discussion.”

“A human network,” he concluded, “is a springboard for human interaction and thus for human action. Let’s view it as such and keep repression and censorship at a minimum.”<sup>15</sup>

## **UUCPnet and the “Iron Curtain” of the ARPANET**

In contrast to the vision of ubiquitous human networking via computers discussed on the Human-Nets mailing list, the Usenet community faced a difficult battle when trying to communicate with those on the ARPANET. Posts on Usenet during the 1981 period reflect the constant efforts and the frustration experienced by those on Usenet who wanted to contribute to the ARPANET mailing lists.

Another popular ARPANET mailing list during this early period of Usenet was the Unix-wizards mailing list. It provided for discussion, the sharing of experiences, of problems, and of software, and for the debate over various issues that faced the Unix community. The mailing list was gatewayed from the ARPANET to Usenet and was available on Usenet as the newsgroup FA.unix-wizards.

Recognizing the early difficulty that those on Usenet had in posting to the ARPANET mailing lists, one user asked: “You mean saying -n fa.unix.wizards doesn’t get back to the ARPANET? Does it just get to Usenet? Or does it go anywhere?”<sup>16</sup>

Another post reported the frustration experienced by those on Usenet who were trying to send messages to mailing lists carried on the ARPANET. The person wrote: “With regard to the ARPA/UUCP gateway problem, it appears that ARPANET sites refuse to process mail from UUCP machines, while UUCP machines typically don’t bother checking who stuff comes from before passing it on. In most cases this costs real money in terms of phone rates, use of spool space,

etc.....”<sup>17</sup>

He proposed that UUCP sites retaliate so that transporting messages to Usenet from the ARPANET would be equally difficult: “We could have messages of the type: ‘Gateway to UUCPnet Closed...Service Unavailable’”

He asked others on Usenet for “any ideas what kind of response would result if this was implemented?”

Responding to this proposal, another Usenet user offered his objection: “I’d rather see messages of this form going back to ARPA: ‘Gateway to UUCPnet open...No Iron Curtain here’” “Or some such self-righteous garbage. Seriously, the interchange of information is too useful to get embroiled in hurt feelings. I get mad when Arpa blindly refuses stuff but would rather try to shame them (good luck!) than play the same game.”<sup>18</sup>

There were those on the ARPANET who sympathized with the problems experienced by the Usenet community in trying to contribute to the ARPANET mailing lists. Commenting on the frustration, a user at a U.S. government site that was both on the ARPANET and on Usenet wrote: “I am also concerned about Usenet participants. We really need to be able to interact with them in a better way, yet UUCP gateways to the ARPANET are VERBOTEN”.<sup>19</sup>

Often Usenet users would try to send messages to the ARPANET gateway only to get back notification that their message had bounced. Common messages notifying Usenet users that their efforts to send messages to the ARPANET mailing lists had failed included: “Sorry not an ARPANET gateway: Unable to deliver Mail” “unix-wizards@sri-unix...Mail has been disallowed between the ARPANET and Uucp net” “unix-wizards@sri-unix...Service unavailable”

Other messages on Usenet during this period describe similar problems. For example, one user describes how he sent out 5 e-mail messages to the mailing list FA.unix-wizards and each came back to him undelivered. He then tried to send the messages to the mailing list again, or in frustration gave up and posted them on Usenet in the newsgroup “net.general” so others could see the problems he was having. He reported: “It doesn’t always work, folks! Last week I submitted 5 letters to ucbvax!unix-wizards; and got each one of them back the very next day, saying ‘service unavailable.’ Depending on the message I either shipped it back right away, or just put it in net.general in disgust.”<sup>20</sup>

## **The ARPANET <=> UUCPnet gateway**

The path set up to make it possible for uucp users of Usenet to contribute to the ARPANET mailing list Unix-wizards was via uucp to “ucbvax”, from “ucbvax” along Berknet to “Berkeley”, the UCB site on the ARPANET, and from that site along the ARPANET via e-mail to “sri-unix”, a site on the ARPANET that would distribute the mailing list back to “Berkeley” or send it out on the ARPANET. The site “sri-unix” was a computer at the Stanford Research Institute. SRI was one of the earliest sites on the ARPANET. Describing how this gateway worked, a user from the University of California at Berkeley wrote: “Ucbvax is currently set up such that if you, as a UUCPnet (Usenet) user, send mail to ‘...ucbvax!unix-wizards’, the message will be \*automatically\* forwarded to unix-wizards@sri-unix (via our internal network and then via the ARPANET).”<sup>21</sup>

He describes how ‘sri-unix’ transported the message back to other sites: “The message is then redistributed by sri-unix to all sites on their ‘master’ list, which include ‘csvax.post-unix-wizards@Berkeley’.”

In this way, the message was sent out on Usenet. “When we at Berkeley,” he explained, “receive something addressed to this rather baroque-looking recipient, it is handed over to our network news program. From there, the message is redistributed via UUCPnet to the rest of the world.”

“ARPANET access,” he noted, “is not available (at least through Berkeley) for ‘private communications’, which would include someone on the UUCPnet attempting to respond to an INDIVIDUAL who submitted something via the ARPANET, or vice versa.”

A user at the Ballistics Research Labs (BRL) noted the burden the gateway imposed on both the University of California at Berkeley and SRI and offered to help if necessary. He wrote: “BRL has a strong commitment to UNIX, and we encourage discussions about UNIX. If SRI gets overwhelmed by the burden of distributing the list, or if we ‘clone’ several lists, we will be glad to take the task of mailing the stuff.”<sup>22</sup>

By September 1981, a post indicated that the ucb<=>sri-unix gateway for the Unix-wizards mailing list was being changed. “This is the last message you’ll be receiving on Unix-wizards through SRI-UNIX,” the writer reported. “Now the list will be mailed out of SRI-WARF(host 1/73);” he noted<sup>23</sup>. Posts could still be sent to “sri-unix”, but they would then be forwarded for transporting to “sri.warf”.

Numerous other users commented on the precariousness of this UUCPnet – ARPANET gateway used by the Usenet (uucp) community during this period. For example, Dave Farber, at the University of Delaware, warned, “As to relaying to the ARPANET, communications could be stopped easily by some agency stating to the sites doing the relaying under the table – to stop it.” Farber was part of the effort to have the National Science Foundation set up CSNET as a way to extend access to the ARPANET to NSF supported academic and industrial researchers. He expressed his hope that CSNET would become a force to change the frustrating situation.<sup>24</sup>

Usenet users had to use some kind of gateway to post to any ARPANET mailing list. “Certain newsgroups (fa.all),” a user on Usenet explained, “are not supposed to be posted to by people. Rather, you are supposed to mail to ucbvax!<newsgroup> to get it to the ARPANET people too.... Another reason was the gateway restriction – direct replies didn’t work!”<sup>25</sup>

ARPANET users also encountered difficulties with communication using the ARPANET. Describing the problem MIT experienced as a result of its efforts to support the ARPANET mailing lists, a user at MIT wrote: “There is always a threat of official or public accusations of misuse of the networks for certain mailing lists. This actually happened with a list called WINE-LOVERS and Datamation [a technical journal].... The fiasco, nearly resulted in MIT being removed from the network, and cost us several months of research time while we fought legal battles to show why our machines should not be removed from the ARPANET. We are all in the hands of our neighbors. The best thing to do is to ensure that we are all educated as to how to take care of each other and ourselves.”<sup>26</sup>

## **Usenet as a Public Computer Users Network**

While the ARPANET was subject to the regulations and policies set by the U.S. Defense Communications Agency (DCA) during this period, Usenet was considered a public computer users network. Policies were proposed, and then were subject to discussion by the Usenet community.

For example, in October, 1981, Horton proposed the following statement of policy for

Usenet:

Usenet is a public access network. Any User is allowed to post to any newsgroup (unless abuses start to be a problem). All users are to be given access to all newsgroups except that private newsgroups can be created which are protected. In particular, all users must have access to the net and fa newsgroups, and to local public newsgroups such as general [net.general].

He continued: "The Usenet map is also public at all times, and so any site which is on Usenet is expected to make public the fact that they are on Usenet, their Usenet connections (e.g. their sys file), and the name, address, phone number and electronic address of the contact for that site for the Usenet directory."<sup>27</sup>

In another post, the writer describing the wide range of topic areas on Usenet, explained: "The net represents a wide spectrum of interest (everything from the latest kill-the-millions-hardware to the latest sci-fi movies)."

He also noted the broad range of sites on Usenet, "The participants of the net, include major (and not so major) universities, corporations, think tanks, research centers, and the like."

"All these people seem to have one thing in common – the willingness to discuss any idea, whether it is related to war, peace, politics, science, technology, philosophy (ethics!), science fiction, literature, etc. While there is a lot of flame," he commented, "the discussion usually consists of well thought out replies to meaningful questions." (He gave examples such as "Should the Postal Service be allowed to control electronic mail?...")

And he added, "I am told that a lot of traffic on the net is not discussion, but real honest-to-goodness work. (Code, applications, ideas, and such.)"<sup>28</sup>

Those posting to Usenet included Unix users, ARPANET users, Usenet users working at Bell Labs, at other industrial sites, at University sites, at government sites, etc. For example, both Thompson and Ritchie, creators of Unix, sometimes responded to Usenet discussions. Thompson contributed to the NET.chess discussions and Ritchie contributed occasionally to fa.unix-wizards, among other newsgroups.

Following is a description of Usenet posted in March 1982.

"Usenet is an international network of UNIX sites with hookups into the ARPA network, too. It is basically a fancy electronic Bulletin Board System. Numerous BTL [Bell Telephone Labs] machines are connected at HO, IH, MH, with a few elsewhere, too. In addition, there are major sites at universities: U C Berkeley, Duke, U Waterloo, and so on (...) And at industry nationwide: DEC, Tektronics, Microsoft, Intel, etc. There are numerous bulletin board categories, set up in a hierarchy."<sup>29</sup>

The article describes how the "fa" newsgroups on Usenet "can reach a very large user community, including Usenet, sites on UUCP, Berknet, BLN, and the ARPANET, as well as sites on the ARPANET which are not on Usenet who get the news via direct electronic mailing." It explains that "Net.all newsgroups are available to all people on the entire network who read netnews." Though not all sites got every newsgroup, "Usenet is defined as all sites that net.all reaches."

Characterizing Usenet as a logical network, as opposed to a physical network, Horton explains that Usenet is a network of sites running Netnews software: "For those of you who don't know, Usenet is a logical network of sites running netnews. Netnews is a network oriented bulletin



board, making it very easy to broadcast a query to a large base of people. Usenet currently has about 50 sites and is growing rapidly.”<sup>30</sup>

Horton emphasizes that Usenet is a users’ network. He explains: “Usenet exists for and by the users, and should respond to the needs of those users.”<sup>31</sup>

He also notes that “Usenet is a cashless network.” This meant that “No person or organization may charge another organization for news, except that by prearrangement.” He explains that a site could charge only for the extra expenses incurred in sending Usenet to another site. And almost every site that received news had to be willing to forward it to at least two additional sites.

Horton’s policy proposal suggested that articles should be of high quality, signed, and that offensive articles shouldn’t be posted. “Peer pressure via direct electronic mail will, hopefully, prevent any further distasteful or offensive articles. Repeated violations,” he noted, “can be grounds for removing a user or site from the network.”

Common to many of the posts in these early years, is the encouragement that users participate and voice their concerns and opinions, both in the ongoing discussion in various newsgroups, as well as in determining the practices and policies guiding how Usenet functions. For example, Adam Buchsbaum, a high school student who played an important role in early Usenet, started the NET.columbia newsgroups, a newsgroup about space issues. He posted the opening message inviting participation: “Greetings fellow space enthusiasts! This newsgroup was designed to inform people on developments in our space program. Although named ‘columbia,’ it will contain articles about the entire space program, including the shuttle for which it is named. Please feel free to reply, comment, criticize, and submit your articles. Also, I hope this will serve as an open ground for discussion about events in the space program. Comments, etc. can be mailed to myself (research!sjb) or submitted directly into the newsgroup. In all, I hope that this will provide an atmosphere for people who are interested in the space program to discuss it and be informed of new events.”<sup>32</sup>

Such articles on Usenet, welcoming contributions from all participants, helped to set a firm foundation for interesting and lively discussion on early Usenet newsgroups.

## **Changing to B News**

The continuing expansion and popularity of Usenet was creating the need for changes in the software. Explaining some of the problems that the ever larger number of posts were creating for those using Netnews, Horton describes how A news recorded subscriptions as a one-line pattern, and a timestamp recorded which messages were read so that you were expected to read all new Netnews at once. He writes: “In the Spring of 1981, Usenet had grown to the point where it was awkward to use A News. It was important to read news in newsgroup order (not by time of arrival) and to quit in the middle leaving some news unread. Also, the user interface of A news resembled V7 /bin/mail, and users were expressing a preference for other e-mail styles (Mail, MH, etc.) and for the Berkeley msgs program.”<sup>33</sup>

At the time, Horton was finishing up his dissertation so he didn’t have the time to do the needed work. Fortunately, however, as Horton recounts, “One day, into my office walked Matt Glickman. He was a local high school student on spring break, looking for a computer project. We teamed up to design B news, and he did most of the coding that week. (The actual production release of B news was announced by Matt at the Winter 1982 USENIX.) I’ll never forget the smile on Matt’s face when he told me, “You know, you’ve made my spring break!”

Horton explains, “B News was patterned after the Rand MH e-mail program, and designed to be compatible enough that MH could be used to read the news. It put each newsgroup in a separate directory (causing a 14 byte limit on newsgroup names that lasted until years later when subgroups made subdirectories) and used a ‘.newsrc’ file to record newsgroup subscriptions and which messages were read.” He notes, “It defaulted to a msgs-style user interface and provided a read-it-all-now escape to a mail program like Mail. In those days it was also reasonable to dump it all to a printer and read it like a newspaper.”

In a post announcing B News, Glickman described the features of the new version of Netnews software that was being written: “I’m working on a new netnews. It is not ready. It is taking a lot longer [then]...it should. I hope to have a rough version running locally this week. Initially, the major new features will be: 1) No more .bitfile, .uindex, or .nindex. Everyone has a .newsrc file in their home directory which contains a list of the articles they’ve already read. This will allow skipping articles and coming back to them later: random-access. The same interfaces are around: /bin/mail, msgs, and print. The -c option still works in the same way, but I’m beginning work on an improved interface with the Berkeley Mail program so that netnews will know which articles were looked at during Mail.”

Among the features Glickman describes are a new article format, an expire feature so articles could be read out of order, but would be cancelled at a determined date, and the netnews command was to be split into two commands, inews, to insert news, and readnews to read news. He also describes how B News provided directories for each newsgroup in a spool directory and all the articles had sequentially numbered filenames in their directories.

“I’ll try to keep you posted on late-breaking developments,” Glickman promised.<sup>34</sup>

## **Automating AT&T and Usenet**

In the summer of 1981, Horton received his graduate degree from the University of California at Berkeley and went to work in Columbus, Ohio at a Bell Labs facility there.

During this period AT&T was automating much of its operations and it recognized that helping to develop and participate in Usenet, and the UUCPnet that was being developed along with Usenet, could help AT&T solve some of the problems raised by its pioneering efforts developing large scale software systems.

Bob Rosin in a post on Usenet, described the difficulties that those working on large-scale software projects encountered and the important technological problem this represented: “There is no cheap, easy way to accumulate the years of experience necessary to deal with complex software based systems. One need only examine the ugly reinventions of assembly language generated by ignorant non-converts and to watch thousands of neophytes wallow in the pits of personal computer assemblers to realize that, while software is in its infancy, people who have studied and built software are way ahead of the great unwashed.”<sup>35</sup>

Recognizing difficulties inherent in large scale software projects, there were those at Bell Labs who labored to encourage management to improve the software development environment. This included adopting and spreading Usenet and e-mail among programmers. One such article posted on Usenet described these efforts: “There is a lot of effort going on now to try to convince management in Bell Labs to improve the software work environment. Good electronic mail and bulletin boards are an important part of that environment. There is a lot of interest in netnews here,

with lots of people from management and even the legal department looking at it.”<sup>36</sup>

During this period, Bell Labs was doing work to develop and implement the 5 ESS switch [Electronic Switching System]. Describing how the 5 ESS was an all purpose electronic switch that would replace the other switches that had been developed for particular purposes. John Hobson wrote in Human-Nets: “Yes, there is such a thing as a #5 ESS. This is a bigger and better ESS, designed to be a replacement for all others. That is, there is one basic configuration, and different versions depending on the capacity needed. This is an improvement over the #1/1A, #2, #3 and #4 ESSes, which are fundamentally different machines, each designed to cover one range of live trunk numbers. (#1/1A is used in large, metropolitan switching offices, #4 in small, rural ones.) The #5 ESS is expected to be out in the field sometime next year.”<sup>37</sup>

The 5 ESS project was a large scale programming project involving many programmers and millions of lines of computer code. Describing the 5 ESS project in a post that appeared on Usenet, the writer explains: “Our project (#5 ESS) uses a lot of remote command execution to support our multi-machine development scenario (13 11/70's + 2 VAXes + 1 IBM 3033 - AP). This environment is treated as though it is what it isn't, a single machine. That is we have developers spread across 7 - 9 PDP-11's + a 370 and they all work on the same project [We produce 'load modules' for 3 processor types...that way.]”<sup>38</sup>

Several articles on Usenet describe how difficult it often was for system administrators to convince their management that it was worthwhile to support Usenet at a work site. For example, describing the situation at Bell Labs, one poster wrote: “Much of the netnews distribution within bell labs is done without any explicit approval. I would be surprised to learn that many of other of the corporate participants in Usenet had explicit approval from management. This makes us all very vulnerable.”<sup>39</sup>

Another poster from ‘cincy’, a site at the University of Cincinnati, in the Department of Computer Science and Engineering, verified that this was the situation elsewhere. He wrote: “When I was at cincy, we had a HARD fight to get the administration to pay the bill.”<sup>40</sup>

Because of the difficulties that those at commercial sites had maintaining their participation in Usenet, a debate developed between those who felt that Usenet should be uncensored and those who felt that an uncensored Usenet might lead their management to cut off access to Usenet. One poster from “tektronix” explained the dilemma: “I am beginning to wonder about Usenet. I thought it was supposed to represent electronic mail and bulletins among a group of professionals with a common interest, thus representing fast communications about important technical topics. Instead it appears to be mutating into electronic graffiti. If the system did not cost anything, that would be fine, but for us here at Tektronix, at least, it is costing us better than \$200 a month for 300-baud long distance to copy lists of people's favorite movies, and recipes for goulash, and arguments about metaphysics and so on. Is this really appropriate to this type of system?”<sup>41</sup>

There were also those at University and government sites who were fearful that certain types of posts might jeopardize grants their sites received. Others maintained that Usenet should be uncensored, but that sites could decide what newsgroups they would carry or what posts they might read. For example, one Usenet user wrote: “What I would really like is to work out methods that would allow as free a flow of information as possible. Some of the problem with the lack of control we have now (i.e. either too many newsgroups/lists or too many messages on one list) may be solvable by implementing new tools and conventions without resorting to brute force. I believe that

there are limits to how much the group of users on one machine can store and comprehend, and that we ought to try to have this be what moderates groups (along with a certain amount of peer-pressure to keep the quality up). Something more along the lines of democracy or physical law than dictatorship, anarchy or even socialism.”<sup>42</sup>

Some sites felt that the content of Usenet should be restricted to topics that management or funding agencies would approve of. Others argued that a site could choose which newsgroups to carry, but that shouldn't limit the broad range of newsgroups that would be available. In summarizing a discussion on this issue that took place at Usenix, Horton noted that newsgroups that seem trivial to one site might be important to another and he reported that those discussing the problem at the USENIX meeting felt that sites could determine what they would carry, but shouldn't impose their tastes on all of Usenet.

A similar debate occurred on the Unix-wizards mailing list. A post reports that some Unix-wizards had dropped off the mailing list complaining about the trivia on the list. Others responded that they didn't want anyone deciding what they could read or not read, so they wanted the list to remain uncensored.

## **Cross Atlantic and Intercontinental Links**

Not only were links within North America difficult to establish, but Dik Winter, from Amsterdam in the Netherlands, describes how the first cross Atlantic Usenet link was delayed until 1982/83 because of the difficulty of acquiring an auto-dialer modem that conformed to European standards. “In Europe,” he writes, “the two people responsible for the link were Teus Hagen and Piet Beertema,” both at the Mathematisch Centrum, a research site in Amsterdam (now called CWI). The Mail link was between decvax <-> mcvox . It connected the site ‘decvax’ at Digital Equipment Corporation (DEC) in the U.S. with ‘mcvox’ in the Netherlands.

Beertema recounts how the early transport of News into Amsterdam was from “philabs” a site at Philips Laboratories, a North American research laboratory for the Dutch company Philips.

Hagen writes that European Unix users who met in European DEC meetings began to do networking in the late 1970's. He describes how relationships were established between Peter Collinson from the University of Kent in England, Keld Simonsen from the University of Copenhagen in Denmark, and the Mathematisch Centrum in Amsterdam.

Timothy Murphy from Trinity College in Dublin, Ireland, explains that a relationship was established between Peter Collinson at the University of Kent and the site “tcdmath” at Trinity, to connect Ireland and England. He remembers how he was one of the founding members of the EUUG (European Unix Users Group) when it was set up at University College in London, England. Describing how the link at Trinity was set up by Brendon Lynch, the first system administrator of the Maths Department unix system, Murphy notes that this link “worked remarkably well – it was far more reliable than its successors, which used to be out of action for weeks at a time.” Murphy remembers that Brendan Lynch at Trinity, “set up an incredibly complicated link from our machine to Kent, which ran on X.25 via the University Dec-20. Our Unix box communicated with the Dec-20, which then communicated with Kent.” He writes that the “tcdmath” was the Irish backbone site for about 4 to 5 years, “maybe from 1980-1984.”<sup>43</sup>

Hagen recounts how links were established with others, including Yves Devilles from INDRIA in Paris, France, Johan Helsingius of the University of Helsinki in Finland, Daniel

Karrenberg of the University of Dortmund in Germany and with other university and technical sites like the Technical University in Vienna, Austria, the University of Stockholm in Sweden and Siemens and Olivetti. Eventually email via UUCP was established with support from Armando Stettner at DEC laboratories. Those involved wanted also to have “a regular exchange of news articles (Usenet) as well,” Hagen adds. Usenet in Europe, he explains, “was born from a tape I took with me from [the] San Francisco USENIX conference...back to Amsterdam.” At a USENIX conference in San Francisco, Hagen met Dan Lorenzeni. Since Lorenzeni worked with Philips, whose Mother firm was from the Netherlands, and Hagen was from the Netherlands, an agreement was made to have Lorenzeni send Hagen tapes of news articles. Hagen describes how a 1200 baud UUCP intercontinental link was set up between “philabs” in the U.S. and “mc” in the Netherlands. He explains that they couldn’t use 2400 baud modems as that “equipment was unreliable, expensive, and modems from different manufacturers could not talk to each other.” On one occasion, Hagen remembers he came into the office “rather early (9:30 am) and noticed that the 1200 baud modem [was] still running. UUCP U.S. and UUCP Holland were sending each other resync messages.” It was running from 7 pm the previous night to the [next] morning. And phone charges were six dollars a minute. “Within 5 minutes,” Hagen remembers, he was in the Director’s Office “trying to explain the high phone bill” which they had run up using “equipment which was not even allowed” as the law in Holland didn’t allow use of a 1200 baud modem. “After that,” Hagen continues, “we made an arrangement with Dan to share more of the costs.”<sup>44</sup>

Lorenzeni, who helped to set up the news link between “philabs” and “mc” concurs. He describes how he worked with Hagen and Beertema to set up the link. “From the beginning,” he writes, “they only wanted certain newsgroups. So they supplied me with the list.” Lorenzeni notes, “From the start, I thought Usenet was a great thing and promoted it as much as possible. Over time the S/N [signal/noise] ratio got worse and worse, but it was always fun.”<sup>45</sup>

Hagen describes some of the frustration that European participants in Usenet experienced. “I can remember a fight in net.general,” he writes, “when someone in the U.S....complained about posts from Europe. The person,” Hagen recounts, said “we were dummies as we introduced errors in the date/time stamp” on the posts from Europe. “He was complaining,” Hagen continues, about “the fact that he was reading news articles which were replies” to posts though they were dated “a day earlier [than] the original post.” He forgot, Hagen notes, that the U.S. was in a different time zone.

Hagen details several other problems faced by the European Netnews community, such as high phone costs, leading them to work out a way all would share in the costs. This led to a well organized network of “backbones” connecting UNIX user groups in different countries. Also, language differences were a problem to be dealt with. One of the results, Hagen remembers, was in a message to all news readers noting that international meant “not everyone is speaking their own national language.” Hagen also describes how he presented the potential of a European net at a conference of EUUG in Paris with a presentation where he showed e-mail and news and made available some illegal modems which were subsequently spread throughout Europe.

In the following post from 1983, Jim McKie at Mathematisch Centrum, discusses some of the difficulties confronting these early European Usenet users. He writes: “Well, the net isn’t collapsing over here, and is already run on a pay-as-you-read basis. I can’t speak for the U.K., and I am sure, as in all things, the U.K. would not like somewhere else in Europe to speak for her (the

U.K. is only GEOGRAPHICALLY close to Europe), but the U.K. gets it's news free from vax135; I don't know how much they get. And we get a small number of groups through 'philabs', ones which people asked us to get, not a blanket coverage anymore. Hopefully we will soon be getting some more news groups from 'decvax', and to those sites which ask for them, we will redistribute. Another major manufacturer has offered some free satellite time, which we are investigating.... We are in the fortunate position of starting up late and having someone (Teus Hagen) who put things on a nice footing.... But it means we have to keep trying to find cheaper ways to obtain the groups, so we can afford to make some mistakes and chuck them later. However, the real problem is that the (soon to be) 3 news feeds supply different groups, and there is no net.anything passed between the U.K. and Europe, so we would perhaps not get a fair and unbiased choice...."<sup>46</sup>

Several of the European Usenet pioneers report that Armando Stettner of DEC soon became involved in helping to get Usenet to Europe. DEC was willing to pay the intercontinental phone bills so e-mail and news traffic were shifted to DEC.

Winter also describes the difficulties that those working to provide a Usenet link to Australia faced to provide Australian – North American connectivity. Robert Elz, at the University of Melbourne in Australia, describes how working with Piers Lauder, news distribution was set up in Australia. The earliest international link was created when Ian Johnstone from UNSW [the University of New South Wales] was invited to Bell Labs in Murray Hill, N.J. in 1980 or 1981. "In any case," Elz writes, Johnstone "arranged a link from Bell Labs...to an IBM mainframe...at the University of Waterloo. The University of Sydney (or UNSW) connected to there using X.25 (which was why Waterloo was chosen....) This link," Elz remembers, "was basically pathetic – messages lost, and lots of manual work involved in transferring what did get transferred, yet it did allow messages through, and was kind of linked to the UUCP net in the U.S. (and Canada)."<sup>47</sup>

Elz explains that "It was probably '83 when the first Usenet news reached here (well, actually, before then I had dialed into Berkeley (ucbvax etc), saved news from time to time in my directory there – anything that looked interesting, and then had it added to the next tape coming back this way, either one I brought after a visit there, or one they were mailing me for some reason). I doubt that counts as a real Usenet connection, but it is probably responsible for a rumor that occasionally makes the rounds about Australia getting news via mag tape, which never really occurred in any meaningful fashion."

Piers Lauder writes that, "All news arrived via Robert's machine in Melbourne University called 'munnari' which still exists in name, if not in original form...munnari acted as the gateway to the rest of Australia."<sup>48</sup>

During this period, Elz attended Usenix conferences in the U.S., usually the summer meetings. While in the U.S. he would usually also spend some time at the Computer Science Research Group (CSRG) at the University of California at Berkeley. During one of the Usenix meetings, Elz writes that he and several UCB related people were hanging around the DEC stand at the conference with Armando Stettner, trying to get BSD unix to work on the (then) new Vax 11/730. "While doing that," Elz writes, "Armando heard of our tenuous net link to the world, and offered to have 'decvax' call us for e-mail transfer.... I wasn't about to say 'no' to that offer."

"Having this free link (to us) available greatly increased use of the net in Australia," Elz notes, crediting DEC, and Stettner, for "helping spread the network into the world outside North America." And he points out that "the free links available to Australia, and Europe, without question

encourage use that would never have happened had there not been this sort of access available – justifying paying for traffic without seeing how useful it can be is very hard to do. On the other hand,” he adds, “having this period of uncharged use allowed people to see the benefits, and get accustomed to it, which then allowed people to be able to justify meeting the bills when that eventually was required.”

Elz explains why it was fortunate that it was possible to have Usenet along with e-mail. “Usenet was just a ‘free optional extra’ (more or less) that came with the e-mail links. If it had ever been much in the way of particular effort, it might never have survived. Still, it did allow us to keep in touch much more actively with what the rest of the world was doing. Being a communal medium it allows one to notice things by accident, which person to person e-mail might never reveal,” he concluded.

## **Setting a Foundation for the Future**

Many of the academic, industrial and government sites participating in the early days of Usenet were involved with computer software or hardware research. The developing network of Usenet sites helped to provide the Unix community with the technical and social support they needed to keep computers functioning and to deal with the perennial upgrades as computer development advanced. Often people online would ask for advice or offer information or programs to others so that people could build on each other’s experiences, rather than “reinventing the wheel.”

In addition to such technical cooperation, newsgroups were developed to discuss a wide range of topics, including world-wide ubiquitous networking in the future (Human-Nets), science fiction (sf-lovers), computer games (NET.games), etc. Socializing was encouraged in NET.singles (or NET.social), recipes were exchanged in NET.food. Music was discussed and recommended in NET.music. The developments and problems of the space program were discussed in NET.columbia (on Usenet) and NET.space (an ARPANET mailing list).<sup>49</sup>

As the interests of people were reflected in their suggestions for new newsgroups, online discussions developed over how to create a process that would make the desired groups possible. The early development of a newsgroup creation process and the discussion over how to structure that process help to demonstrate that a great deal of effort by many people was expended to create functional and democratic procedures for the early Usenet. The earliest newsgroups were all unmoderated. Everyone had the right to participate and contribute their views. A rich and interesting content emerged that surprised even the participants themselves.

The development and spread of computers require new means of communication like Usenet. A great deal of effort and discussion went into creating Usenet. This has provided Usenet with the strong foundation needed to support the technical and educational needs that result from the increasing use of computers in our times. Usenet has grown and flourished and in turn serves the needs of those using and developing computer technology.

The Unix community gave the world high tech software tools that could perform wondrous feats with simple programs.<sup>50</sup> The Usenet community took these tools and used them to open up and create channels for communication so that those in the online Unix community could help each other wield the tools. In a society that hopes to progress in this era of rapidly developing computer hardware and growing demands for computer software, more and more of the population needs to have access both to the tools and to the means of communication needed to wield these tools. This

is the foundation of the cooperative and democratic culture that Usenet has pioneered and made possible. It is important to understand and build on these roots and to nourish and expand this cooperative culture. It is important to make this cooperative networking culture, this marriage of an ever larger network of computers and people, available to ever broader sectors of the population if the promise of computer technology to provide a better and more productive world is to be realized. We are much closer to the dream of a WorldNet today, than we were in 1979, thanks to the hard work of the Usenet pioneers in setting a firm foundation. We will need to build on the foundation they set if we hope to make the dream of a WorldNet, of ubiquitous computer networking, a reality.

---

#### Notes for Chapter 10

1. The following account is from e-mail correspondence from Tom Truscott, which has been compiled into an unpublished interview “Interview with Tom Truscott: On the Environment and Early Days of Usenet News.”
2. The paper was by Claude E. Shannon, “A Chess-Playing Machine”, *Scientific American*, February 1950, p. 48.
3. The next oldest paper Truscott found was by Alex Bernstein and M. de V. Roberts, “Computer versus Chess-Player,” *Scientific American*, June 1958.
4. This was the July, 1974 paper by Dennis M. Ritchie and Ken Thompson, “The Unix Time-Sharing System”, published in *Communications of the ACM*, Vol. 17 no. 7, pp. 365-375. A reference to chess is on p. 375.
5. Usenet Archives, Steve Bellovin, Wed Oct 10 19:48 PDT 1990: Available via ftp: weber.ucsd.edu <usenet.hist>.
6. E-mail correspondence from Tom Truscott. Ward Christensen and Randy Seuss had set their bulletin board up in Chicago on February 16, 1978, predating Usenet. They operated on a North Star Horizon 4 Mhz Z-80 CP/M machine with a 5 Mb drive for posting and reading of messages. (See “The Online User’s Encyclopedia” by Bernard Aboba, Reading, MA, p. 59)
7. E-mail correspondence with Mark Horton, August 1995. Horton like Truscott was introduced to programming using BASIC as a high school student in 1970. He writes that he learned to program in BASIC, “first on the GE system, but that was expensive. First Portland and then San Dieguito HS’s got access to HP 2000 BASIC systems with unlimited usage.”
8. Mark Horton, Mon Oct 15 19:49 PDT 1990, Usenet History Archives Available via ftp: weber.ucsd.edu <usenet.hist>.
9. Mark Horton, Tue Nov 24 04:51 PST 1992, Usenet History Archives.
10. E-mail communication from Mark Horton.
11. E-mail communication from Truscott.
12. Rich Zellich, 16 Feb. 1982, posted on Usenet in post by btempleton, watmath.2114, Subject: ARPANET mailing list directory.
13. 17 Oct 1982, Zaleski at Ru-Gren, Subject: Why not AT&T for World Net by Michael Zaleski.
14. 19 Oct 1982, Greg Skinner, <uc.bds at MIT-EECS at MIT-MC>, Subject: Worldnet responses.



15. 03 June 1981, Jorge Phillips, Subject: administrivia.
16. cincy.151, fa.unix-wizards, cincy!chris, Tue Apr 7 13:16:12 1981, Subject: to unix-wizards.
17. A. Feather, pur-ee.123, net.general, pur-ee!aef, Mon Aug 24 15:13:14 1981, Subject: UUCP gateway.
18. esquire.127, net.general, Wed Aug 26 09:48:51, UUCP gateway, Re: A Feather's suggestion.
19. ucbvax.2946, fa.unix-wizards, Re: PROPER FORUM, mike@bmd70@BRL, Fri Sep 4 14:55:10 1981.
20. ucbvax.2858, Sat Aug 29 10:17:34 1981, purdue!cak.
21. Geoff Peck, ucbvax.2842, fa.unix-wizards, ARPANET access.
22. ucbvax.2946, fa.unix-wizards, Re: PROPER FORUM, mike@bmd70@BRL, Fri Sep 4 14:55:10 1981.
23. FA.unix-wizards, ucbvax.3198.
24. ucbvax.2955, Sat Sep 5 07:34:34 1981, from farber@udel. See description of CSNET in Appendix IV.
25. Net.news, cbosgd.113, Sat Oct 3 19:51:41 1981, Re news.
26. ucbvax.5782, fa.digest-p, Thu Jan 14 05:46:13 1982, From cStacey@MIT.AI.
27. NET.news, cbosgd.120, Tue Oct 13 20:56:30 1981, cbosgd!mark, Subject: Whether the sys and uname files are public.
28. NET.news, wolfvax.53, net.news, wolfvax!jcz, Mon Nov 2 21:47:32 1981, Net Names, In Real Life: Carl Zeigler, Location NCSU, Raleigh.
29. ucbarpa.1182, net.sources, Subject: ARPAVAX: Usenet, Tue, Apr 20, 1950:48 1982, misc/newsinfo, from eiss!ladm, Fri Mar 19 16:20:27.
30. Mark Horton, fa.unix-wizards, ucbvax.4080, Sun Sep 27 22:04:41 1981, Usenet membership.
31. NET.news, cbosgd.794, Wed Dec 23 21:28:32 1981, Subject: Proposed Usenet policies.
32. net.columbia, research!sjb, Thu Sep 17 07:28:50 1981, Adam Buchsbaum kept the official list of newsgroups and published it regularly to the net for several years in the mid 1980s.
33. E-mail correspondence from Mark Horton, Mon Jul 24 1995. MH "Mail Handler" was developed by Bruce Borden of the Rand Corp, with help from Stockton Gaines and Norman Shapiro, to be used on the ARPANET. (See "UNIX Communications" by Bart Anderson, Byran Costales & Harry Henderson, Carmel, Indiana, 1991, p.198.)
34. Aucbonyx.118 NET.news utzoo!duke!decvax!ucbvax!Onyx:glickman Fri May 16 10:29:40 1980 New Netnews.
35. Bob Rosin, Bell Labs, Linroft, N.J., houxf.148, NET.general, houxf!rosin, Fri May 7 09:26:53 1982, Re: debugging microcode in writable control store.
36. NET.news, ihns.995, net.news, ihns!warren, Subject: Misconceptions about Bell Labs Netnews Content.
37. Human-Nets Digest, 28 May 1981.

38. NET.blfp, alanr, Subject: Remote Command Execution, File Installation, Tues Jul 21 10:42:15.
39. ihns!warren, ihns.995, Subj: Misconceptions about Bell Labs, Netnews Content.
40. purdue.139, net.general, net.news, cak, Sat Dec 17 19:27:08 1981, Subj: Freedom of the dataways.
41. NET.misc, dadlaA.98, net.misc, dadlaA!steve, Mon Mar 15 21:56:49 1982, Subject: Trivia on the Net.
42. Asri-unix.429, net.news, utzoo!decvax!ucbvax!menlo70!sri-unix!knutsen Tue Jan 5 17:46:42 1982, USENET policy, reposted from Date: 15 Dec 1981 at 1522-PST, From: Andrew Knutsen <knutsen@SRI-UNIX>, Subject: Re: read-only newsgroups (net.news cbosg.193)
43. E-mail from Timothy Murphy at the Trinity College. Murphy did the technical work to split uucp into 2 processes, which was necessary “as it was too large to run in the 64k (data+text) allowed on our pdp-11 under Unix edition 6,” he explains.
44. E-mail correspondence from Teus Hagen, August 1995.
45. E-mail correspondence from Dan Lorenzeni, August 1995.
46. Dec. 15 1981 at 1522, Andrew Knutsen <knutsen@SRI-UNIX>. From: rti!mcnc!unc!duke!decvax!linus!philabs!mcvax!jim Wed, 3-Aug-83 01:12:41 EDT Jim McKie Mathematisch Centrum, Amsterdam ... {decvax|philabs} !mcvax!jim (mcvax.5322) net.news : Re: cost of sending netnews to aliens.
47. E-mail from Robert Elz, October, 1995. Elz remembers that “Johnstone arranged a link from Bell Labs. This one worked by Bell Labs using cu [call unix, i.e. an application to allow dialing out from one host to connect to another – telnet over phone lines, and had some primitive capture and send mechanisms, with no correctness checking] – with a back end process filtering the output so as not to overrun the IBM after end of line and such.”
48. E-mail from Piers Lauder.
49. A listing of all the newsgroups available by March, 1982 is in the appendix.
50. See for example the thunderclap in the Appendix III.

---

Thank you to Tom Truscott, Mark Horton, Rob Scott, Dik Winter, Russell Lowell and others on Usenet for their comments on an earlier draft and their helpful suggestions. In addition, thank you to Teus Hagen and Dan Lorenzeni for their helpful info about setting up the Cross-Atlantic link, Timothy Murphy for background on the link to Ireland, and to Robert Elz and Piers Lauder for information about the link from North America to Australia. Also, thanks to Henry Spencer and others at the University of Toronto for archiving early Usenet posts so folks can understand the early days of Usenet when it was possible to read every post. Also thanks to the Usenet pioneers and to Bruce Jones for setting up the Usenet history Archives at weber.ucsd.edu <usenete.hist> and for making material available online.

---

## Appendix I

One of the Usenet pioneers, Henry Spencer, at the Zoology Department at the University of Toronto in Canada, archived Usenet from the date his site “zoo” joined Usenet in May 1981 to recent times. The earliest posts he archived are contained in the A-News archive, which covers posts that appeared on Usenet from May 1981 to 1983.

Another important source of early Usenet history has been gathered by a graduate student at the University of San Diego (UCSD) in California. Bruce Jones who began work to document the sociology of Usenet for his Ph.D. thesis,

collected recollections and background from several of the Usenet pioneers and made them available online via anonymous ftp from weber.ucsd.edu in the directory <usenet.hist>.

---

## Appendix II

### Two Early Lists of Newsgroups Appearing in Usenet in 1982

```
net.news.group
utcsrgv!utzoo!decvax!duke!chico!harpo!mhtsa!ihnss!cbosg!teklabs!t ekmdp!azure!curts
Tue Jan 26 13:50:13 1982
grouplist
```

FA groups are “from the ARPANET” and are mostly copies of mailing lists or “digests” distributed on that network. (A digest is a collection of mail put together by an editor and sent out every so often. It is much like a newsletter.)

A special convention applies to submissions to FA newsgroups. As previously described, you should not post directly to the newsgroup, since this will be seen by people on Usenet but not by the people on the arpanet who get the list directly mailed to them. Instead, send mail to the name of the group on site.

For example, to post an article to fa.human-nets, you might mail to chico!ucbvax!human-nets (if chico is the proper route to get to ucgvax – this route varies depending on your system). FA groups and their corresponding mailing lists can reach a very large user community, including Usenet sites on UUCP, Berknet, BLN, and the ARPANET, as well as sites on the ARPANET which are not on Usenet, who get the news via direct electronic mailing.

The following is a list of digests:

---

NEWSGROUP	Description
fa.arms-d	Discussion and info on strategic weapons.
fa.arpa-bboard	Announcements that are posted to all arpanet bboards   are also fed into this newsgroup.
fa.digest-p	People who deal with digests. Mostly the people who   moderate them.
fa.editor-p	Interest group in computer editors, both text and program.
fa.energy	Topics relating to alternate energy production,   conservation, etc.
fa.human-nets	A daily moderated digest with discussions of computer-aided   human-to-human communications. Probably the most widely read   ARPANET publication.
fa.info-cpm	CP/M and other operating systems for micro computers.
fa.info-micro	Microprocessor and microcomputer discussions.
fa.info-terms	Opinions/queries about what’s a good/bad computer terminal.
fa.info-vax	VAX interest group. Seems to be mostly VMS issues, but some   hardware discussions too.

fa.poli-sci	Political Science discussions digest.
fa.sf-lovers	Science Fiction book/movie reviews, etc.
fa.space	Digest containing comments on the space program and outer space in general.
fa.tcp-ip	Digest relating to the TCP and IP network protocols.
fa.telecom	Technical topics relating to telecommunications, especially the telephone system. A digest recently spun off from fa.human-nets.
fa.teletext	Teletext discusses all aspects of "esoteric" data systems. This includes teletext, viewdata, closed-captioning, and digicasting.
fa.unix-cpm	CPM/UNIX discussions.
fa.works	Interest group on personal workstations (e.g. Apollo, Perq, Xerox Star, etc).

Newsgroups are intended to be available to all people on the entire network who read netnews. This does not mean they go to every machine, since some machines restrict the volume of news that comes in. It is assumed that users of such restricted machines can read news on another machine on which they have a login. Newsgroups reach all of Usenet (including Usenet sites on the ARPANET) but do not reach any sites that are not on Usenet. That is, Usenet is defined as all sites that net.all reaches.

NEWSGROUP	Description
net.general	General information.
net.applic	Applications programs for UNIX. Discussions seem to center around functional programming languages.
net.auto	General Information for automobile owners.
net.auto.vw	Subgroup net.auto – for owners of Volkswagon Rabbits.
net.aviation	General information about aviation.
net.bugs	General information about bug reports and fixes.
net.bugs.2bsd	Subgroup net.bugs – 2nd Berkley Software distribution
net.bugs.4bsd	Subgroup net.bugs – 4th Berkley software distribution
net.bugs.v7	Subgroup net.bugs – Version 7 or UNIX System III
net.chess	General information about computer chess. Gatewayed to ARPANET mailing list but appears as newsgroup rather than a digest.
net.columbia	General information on space shuttle and space programs

-----  
net.cycle | General information about motorcycles.  
-----  
net.eunice | General information for sites running SRI Eunice system | which simulates UNIX on VMS.  
-----  
net.games | Information and discussion on computer games.  
-----  
net.games.rogue | Subgroup net.games – rogue  
-----  
net.games.frp | Subgroup net.games – fantasy role playing games  
-----  
net.games.trivia | Trivia contests and results.  
-----  
net.ham-radio | Topics of interest to amateur radio operators.  
-----  
net.jokes | The latest “good” joke you’ve heard?  
-----  
net.lan | Local area network interest group.  
-----  
net.lsi | Large Scale Integrated Circuit discussions.  
-----  
net.misc | Miscellaneous discussions that start in net.general but | are not permanent enough for their own newsgroup.  
-----  
net.movies | Movie reviews by members of Usenet.  
-----  
net.music | Computer generated music.  
-----  
net.news | Discussion of netnews itself.  
-----  
net.news.b | Subgroup net.news – specific to bnews.  
-----  
net.news.directory | to post all or part of the Usenet directory,  
-----  
net.news.group | for discussions about proposed new newsgroups,  
-----  
net.news.map | for discussions about maps of newsites.  
-----  
net.news.newsite | to announce a new site.  
-----  
net.news | for discussion of Usenet policies.  
-----  
net.oa | Office Automation/Word Processing interest group.  
-----  
net.periphs | Queries and discussions about particular peripherals. (“Does anyone have a driver for a framus-11?”)  
-----  
net.rec | General info on recreational (participation) sports.  
-----  
net.rec.bridge | Subgroup of net.rec – contract bridge.  
-----  
net.rec.scuba | Subgroup of net.rec – scuba diving.  
-----  
net.rec.ski | Subgroup of net.rec – skiing.

```

-----
net.records    | Info and opinions about records (and tapes ?).
-----
net.rumor     | For posting of rumors.
-----
net.sources   | A place for sources and the distribution of material
              | in large volume. More for software distribution than | for general info.
-----
net.sport     | General info about spectator sports.
-----
net.sport.baseball | Subgroup of net.sport – for baseball.
-----
net.sport.football | Subgroup of net.sport – for football.
-----
net.sport.hockey | Subgroup of net.sport – for hockey.
-----
net.taxes     | Tax advice and queries.
-----
net.test      | Test messages are posted here.
-----
net.travel    | Requests, suggestions, and opinions about traveling
-----
net.ucds      | Circuit drawing system.
-----
net.unix-wizards | ARPANET mailing list for UNIX Wizards. Anything and
              | everything relating to UNIX is discussed here. This
              | list is gatewayed to the ARPANET mailing list
              | but appears like a regular newsgroup to Usenet.
-----

```

This is the first pass at establishing a list of newsgroups. My intent is to update the list every week or so. Although this list is incomplete, it seemed that a partial list at the right time might be better than a complete list that arrives too late. If you have additions, corrections, or suggestions, please send them to me at:

ucbvax!teklabs!tekmdp!curts

Curt

>From cbosg!harpo!npois!eiss!ladm Fri Mar 19 16:20:27 1982  
 Subject: newsinfo.shell  
 Newsgroups: net.sources

Newsgroup naming conventions:

NO prefix= LOCAL ONLY

btl. = Bell Labs

net. = USENET wide categories

fa. = from ARPA-Net (no return feed, except via mail)

\_\_\_\_.all= everything in category "\_\_\_\_".

The netnews newsgroups of most interest are:

general: local general information

btl.all : BTL Everything.

net.general: general net-wide announcements

net.bugs.v7: reports of bugs and/or solutions to UNIX V7

net.news.b: news about our version of netnews

FA groups are "From the ARPANET" and are mostly copies of mailing lists or "digests" distributed on that network. (A digest is a collection of mail put together by an editor and sent out every so often. It is much like a newsletter.)

NEWSGROUP | Description last update 3/19/82

---

fa.arms-d | Discussion and info on strategic weapons.  
fa.arpa-bboard | Announcements that are posted to all arpanet bboards  
| are also fed into this newsgroup.  
fa.digest-p | People who deal with digests. Mostly the people who  
| moderate them.  
fa.editor-p | Interest group in computer editors, both text and program.  
fa.energy | Topics relating to alternate energy production,  
| conservation, etc.  
fa.human-nets | A daily moderated digest with discussions of computer-aided  
| human-to-human communications. Probably the most widely read  
| ARPANET publication. AVAILABLE PRINTED ONLY.  
fa.info-cpm | CP/M and other operating systems for micro computers.  
fa.info-micro | Microprocessor and microcomputer discussions.  
fa.info-terms | Opinions/queries about what's a good/bad computer terminal.  
fa.info-vax | VAX interest group. Seems to be mostly VMS issues, but some  
| hardware discussions too.  
fa.poli-sci | Political Science discussions digest. TURNED OFF.  
fa.sf-lovers | Science Fiction book/movie reviews, etc. PRINTED ONLY.  
fa.space | Digest containing comments on the space program and outer  
| space in general. This is fed to net.space, ALSO PRINTED.  
fa.tcp-ip | Digest relating to the TCP and IP network protocols.  
| TURNED OFF.  
fa.telecom | Technical topics relating to telecommunications, especially  
| the telephone system. A digest recently spun off from  
| fa.human-nets. PRINTED ONLY.  
fa.teletext | Teletext discusses all aspects of "esoteric" data systems.  
| This includes teletext, viewdata, closed-captioning, and  
| digicasting.  
fa.unix-cpm | CPM/UNIX discussions.  
fa.works | Interest group on personal workstations (e.g. Apollo, Perq,  
| Xerox Star, etc).

---

NEWSGROUP | Description last update: 3/19/82

---

net.general | General information.  
net.followup | follow-up articles to those posted in net.general  
net.applic | Info - applicative language and related architecture.  
net.auto | General Information for automobile owners.

net.auto.vw | Subgroup net.auto - for owners of Volkswagon Rabbits.  
net.aviation | General information about aviation.  
net.bugs | Genrerall information about bug reports and fixes.  
net.bugs.2bsd | Subgroup net.bugs - 2nd Berkley Software distribution  
net.bugs.4bsd | Subgroup net.bugs - 4th Berkley software distribution  
net.bugs.v7 | Subgroup net.bugs - Version 7 or UNIX System III  
net.columbia | General information on space shuttle and space programs  
net.cooks | Interest group - food, cooking, cookbooks, and recipes.  
net.cse | Computer Science Education  
net.cycle | General information about motorcycles.  
net.dcom | data communication - modems,multiplexers,port selectors etc.  
net.eunice | Info on sites using SRI Eunice - simulates UNIX on VMS  
net.games | Information and discussion on computer games.  
net.games.rog | (net.games.rogue) Subgroup net.games - rogue  
net.games.frp | Fantasy Role Playing games  
net.games.triv | (net.games.trivia) Trivia contests and results.  
net.ham-radio | Topics of interest to amateur radio operators.  
net.jokes | The latest "good" joke you've heard?  
net.lan | Local area network interest group.  
net.lsi | Large Scale Integrated Circuit discussions.  
net.math | mathematical discussions (eg. what is  $\lim_{x \rightarrow 0} \log(x) - \log(x)$ )  
net.micro | micro-computers, see also fa.info-micro.  
net.misc | Discussions not permanent enough for a newsgroup.  
net.movies | Movie reviews by members of USENET.  
net.music | Computer generated music.  
net.news | Discussion of netnews itself, and its policies  
net.news.b | Subgroup net.news - specific to bnews.  
net.news.direc | (net.news.directory) all or part of the USENET directory  
net.news.group | for discussions about proposed new newsgroups.  
net.news.map | for discussions about maps of newsites.  
net.news.newsi | (net.news.newsite) to announce a new site.  
net.oa | Office Automation/Word Processing interest group.  
net.periphs | Queries and discussions about particular peripherals.  
net.rec | General info on recreational (participation) sports.  
net.rec.boat | boating (sail and motor???)  
net.rec.bridge | contract bridge.  
net.rec.scuba | scuba diving.  
net.rec.ski | skiing.  
net.rec.skydiv | (net.rec.skydive) sky diving  
net.records | Info and opinions about records (and tapes ?).  
net.rumor | For posting of rumors.  
net.sf-lovers | Science Fiction Lovers - undigested from fa.sf-lovers  
net.sources | For large volume material, source program distribution  
net.space | Space programs and research - undigested from fa.space  
net.sport | General info about spectator sports.  
net.sport.base | (net.sport.baseball) Subgroup of net.sport - for baseball.  
net.sport.foot | (net.sport.football) Subgroup of net.sport - for football.  
net.sport.hock | (net.sport.hockey) Subgroup of net.sport - for hockey.  
net.taxes | Tax advice and queries.  
net.test | Test messages are posted here.  
net.travel | Requests, suggestions, and opinions about traveling  
net.ucds | Circuit drawing system.



net.unix-wiza | (net.unix-wizards) Discussion of UNIX - gatewayed to ARPANET  
net.wines | Info and recommendations about wines and alcoholic beverages

---

from ucbarpa.1182, net.sources,  
utzoo!decvax!duke!harpo!npois!ucbvax!ARPAVAX:usenet, Tue Apr 20  
19:50:48 1982, misc/newsinfo

---

### Appendix III

“Like a thunderclap from the sky, the pipeline slays the problem.”

Following is a script provided by Tom Truscott. Such particularly powerful pipelines have been referred to by Ken Thompson as “hunderclaps”.\*\*\*\*

To find how often each word appears in files named book.1, book.2, ...:

```
cat book* | tr -cs '[A-Z][a-z]' '\012' | sort | uniq -c | sort -nr | more  
-----
```

Truscott notes “Alas, thanks to so many \*IX variants this pipeline doesn’t always work, on some the underlined text needs to be ‘[\012\*]’.”

---

### Appendix IV

1983 Post on CSNET

Relay-Version: version B 2.10 5/3/83; site utzoo.UUCP  
Posting-Version: version B 2.10 5/3/83; site utcsrgv.UUCP  
Path: utzoo!utcsrgv!peterr  
From: peterr@utcsrgv.UUCP (Peter Rowley)  
Newsgroups: net.news  
Subject: Usenet Inc == CSNET ?  
Message-ID: <1857@utcsrgv.UUCP>  
Date: Sun, 31-Jul-83 05:48:11 EDT  
Article-I.D.: utcsrgv.1857  
Posted: Sun Jul 31 05:48:11 1983  
Date-Received: Sun, 31-Jul-83 08:27:27 EDT  
Organization: CSRG, University of Toronto  
Lines: 65

>From literature and a presentation given at the Toronto USENIX, my impression is that CSNET is...a form with as little bureaucracy as possible, and with non-profit status. Some excerpts from “csnet news”, no. 1 (may 83):

“CSNET was established in 1981 with a 5 year grant from the National Science Foundation. From the beginning, the goal of the project has been to create an independent network, fully supported by membership dues and service fees. With this in mind, NSF has adopted a schedule of dues and fees for 1983, and the Coordination and Information Centre (CIC) has developed models of expected service charges. ...

CSNET dues support software maintenance and development, hardware, tech.staff, and other expenses associated with shared resources such as the PhoneNet relays, the Name Server facility, and the CIC. Dues also defray the costs of documentation, network management, and network governance activities. Each member of CSNET is required to pay

yearly dues to support CSNET operation.”

Here are the current dues:

Industrial: \$30K/yr    Government: \$10K/yr    Univ: \$5K/yr

The two relays mentioned are at Rand Corp. in Santa Monica and U. Delaware. A PhoneNet site dials into the closest relay, except where a site has been moved to the other relay for load balancing. An X25Net (Telenet) site accesses CSNet by buying special hardware from Telenet, getting X.25/TCP-IP sw from Purdue (runs only on BSD) and paying Telenet \$1000/mo. for a 4800 baud line, packet charges not included. The break-even point for phone/telenet is about \$22K/yr.

PhoneNet sites pay service fees too:

	Day	Evening/Night
Dial-out	.80	.40
Dial-in	.10	.10

These are on top of any telco charges and are in terms of \$/connect-minute. The CSNet model predicts service fees of between \$125 and \$625 a year for light-heavy PhoneNet mail users (note that there is no news). X25 service fees have not been established yet.

All the details can't be gone into in the space of a news article, but it appears that CSNET provides the reliable mail and name server services desired, and could handle news.

The bureaucracy involved is the 6 member management committee, which appears to be responsible to the NSF, and the >=4 member staff of the CIC, located at BBN in Cambridge Mass.

-----  
Comment:

It appears that a lot of work has gone into setting CSNET up, together with a good deal of money from the NSF. For this to happen again with Usenet would probably require private, for-profit funding (public funders would say "Use CSNET"), resulting in a for-profit organization probably more expensive than CSNET.

The obvious statement to make is that sites who want a "Usenet Inc." should cough up the CSNET dues and join that network, then help/urge them to get news going. Note that this would not satisfy the user-pay advocates, as univ's (and government sites) get a break on dues. I would suggest, however, that Usenet as it is, with no bureaucracy at all, is a valuable thing to preserve. It has an active, informed community capable of contributing software and manpower to the net. If a new news/mail pkg is created, no bureaucracy need be convinced of its worth...all that must be done is to post it to net.sources. Those who feel it can be used, in the given environment (with all requisite compatability problems), may use it. This seems as democratic as one could hope for.

peter rowley, U. Toronto CSRG  
{cornell,watmath,ihnp4,floyd,allegra,utzoo,uw-beaver}!utcsrgv!peterr  
or {cwruecmp,duke,linus,lsuc,research}!utzoo!utcsrgv!peterr

Relay-Version: version B 2.10 5/3/83; site utzoo.UUCP  
Posting-Version: version B 2.10 5/3/83; site umcp-cs.UUCP  
Path: utzoo!linus!decvax!harpo!seismo!rlgvax!cvl!umcp-cs!chris  
From: chris@umcp-cs.UUCP  
Newsgroups: net.news  
Subject: CSNet  
Message-ID: <1341@umcp-cs.UUCP>  
Date: Mon, 1-Aug-83 18:09:10 EDT  
Article-I.D.: umcp-cs.1341

Posted: Mon Aug 1 18:09:10 1983  
Date-Received: Mon, 1-Aug-83 22:49:50 EDT  
Organization: Univ. of Maryland, Computer Science Dept.  
Lines: 18

CSNet, as it stands now, would collapse instantly under the weight of netnews. Without some way of distributing the load, the two relays would not be able to handle all the traffic. That's the basic trouble with a centralized system, and is the reason for all the research into distributed computing. You can only squeeze so much into a single system.

- Chris

PS Don't get me wrong, I think CSNet is actually doing quite well. It hasn't been around long enough to solve the initial problems (like slow software). Supposedly MMDF II is a big step in increasing speed. - ACT

--

In-Real-Life: Chris Torek, Univ of MD Comp Sci  
UUCP: {seismo,allegro,brl-bmd}!umcp-cs!chris  
CSNet: chris@umcp-cs

---