Simula and Smalltalk: A Social and Political History

Benedict Dugan © 1994

1. Introduction: Views on Science and Technology

The traditional view of the interrelationships between science, technology, and society holds that scientific research is a purely objective, rational process driven by a humanistic desire to accumulate knowledge. Further, technology is considered a product of science; science generates knowledge, and technology is the mere application of this knowledge. Finally, the accumulation of knowledge and its application, technology, always benefits humankind. This outlook on science and technology, while it has fallen into disfavor in the history and philosophy of science community, is still preeminent in many "hard-science" research disciplines, as well as in the greater, non-technical, social sphere. From a young age, we are inundated with an image of what it is to "do" science. This image is one of a lone, (usually male) scientist sitting in his lab, isolated from the troubles of the real world, running experiments and making discoveries. When we enter school, we are taught about the "scientific method" of hypothesis development, experimentation, unbiased evaluation of data, leading to modification of the original hypothesis, further experimentation and so on. This traditional view is instantiated and reinforced in thousands of high-school chemistry and physics classes on a daily basis.

In this paper, I shall take a different approach, marked by the notion that science and technology are social constructs. First, science is not simply a purely objective, rational process. Second, technology is not simply applied science. While it is true that modern technology owes much to scientific research, history is full of examples of technological developments which relied in no way upon knowledge developed by the so-called "scientific method." Third, because science and technology are inherently social processes, their products (knowledge, technological artefacts) reflect the values of the socio-historical context in which they were conceived. In short, value-free science and technology is an oxymoron. Fourth, science and technology have the potential to greatly alter the course of society. In lieu of this, it is vital that we carefully evaluate the products (be they knowledge or technology) of our enterprises. This necessitates asking more than the traditional questions of,"Is it faster/stronger/lighter/better?" It requires us to investigate and understand the ways in which scientific and technological processes will guide or affect the culture in which they are embedded.

How do we apply this critique to the study of programming languages? A course in the study of programming languages is traditionally a comparative, historical survey whose prime goals include giving students some experience with a range of programming languages -- thereby helping them understand higher level issues related to language design and implementation. Languages, as technological artifacts, are evaluated on the basis of syntactic and semantic properties as well as implementation schemes, performance, and software engineering issues. In this way, the course evaluates the history of programming languages by applying criteria which are exclusively internal to the field of computer science. However, there are problems inherent in this approach. Namely, it can lead the student towards a dangerous, but comfortable ignorance of the broader issues related to technological development. Given my position regarding the necessity of evaluating scientific and technological developments from a broad, cultural perspective, I feel that we must re-evaluate the traditional content of courses such as these. The study of programming languages need not be transformed into a historical or philosophical inquiry into meta-issues related to technology in general and/or programming languages in particular. Rather, we should pause to consider the development of programming languages in the greater socio-historical context, which is what I intend to do herein.

The complete study of a technological artifact entails answering three main sets of questions: (A) Where does this technology come from? What is its history? (B) What are the properties of this technology? How does it work? What does it do? (C) How will this technology affect the social sphere? Generally, traditionalists ignore (C) altogether. When they examine (A) and (B), they do so in the limited, self-referential fashion discussed above. In this paper, I intend to focus primarily on areas (A) and (B) from a "technology as social construct" perspective. This means that I will go beyond developing a "family-tree" or discussing the syntax and semantics of programming languages. Instead, I intend to study the key designers and their philosophical and/or political motivations, expose the cultural backdrop against which these languages arose and discuss their <u>ideological</u> properties. While any language could be studied in this fashion, I will limit my attention to Simula and Smalltalk. I have chosen these two languages because they are usually viewed as the "first" object oriented languages. Further, their stories will suggest the occurrence of a Kuhnian revolution of sorts[1], in which Simula poses a challenge to the existing paradigm of imperative programming, and Smalltalk answers this challenge with the introduction of a qualitatively new paradigm.

2. Simula

2.1 Cultural Background

To understand the background of the development of Simula, we must first discuss the history of simulation. With the Scientific Revolution, came the idea that by developing models of real-world phenomena, we could gain understanding of nature and make predictions about the future. Further, by objectively observing and collecting data about our environment, we could either verify or invalidate those models. Model building is thus central to the scientific method and the rationalist tradition. As the goal of the scientific project was the application of objective, rational analysis to all of nature, it was inevitable that the social realm would eventually fall under its lens. For example, as early as the late 19th C, Fredrick Taylor used these methods to study the work environment and show that he could increase the efficiency of workers by a large margin. Hence, not only are models prevalent in the "hard" sciences (chemistry, physics, etc.), but also have a long history in the "soft" sciences (psychology, international relations, political science, etc).

The application of models as a tool for analysis in the social sphere is perhaps most profoundly expressed in the domain of operations research. With the advent of the digital computer, the OR community immediately latched-onto it as a valuable tool for performing the rote calculations necessary to make predictions and optimizations based on their symbolic or procedural models. By the 1960s, the idea that simulation could be applied in a vast array of domains gained wide acceptance in virtually all disciplines which employed computers. From computer science and cybernetics to psychology and political science, simulation came to be viewed as a panacea for a huge range of unsolved problems. Simulation became a unifying theme for these disciplines; they all rallied around the promise of simulation.

The preface to the proceedings of the Third Annual Symposium of the American Society for Cybernetics (TASASC) reads, "The motivation for the conference was to bring together the people with the problems and the people with the technology to see if somehow interaction between the two might suggest some useful courses of action. Several of the attendees were selected because of their experience in developing simulations of major social systems."[2] This is to say, mix equal parts of technology, simulation techniques, and social scientists and we might get a solution. This was not a conference of fringe researchers; the participants included representatives from major universities such as MIT, Harvard, and Yale; industry pace setters such as IBM and UNIVAC; and the federal government.

The papers presented at this conference are rife with examples of how useful computers are or will be in simulating and modeling social and political processes and gathering and maintaining data collected in the real world. Viewed as a whole, they present a kind of utopian dream of the marriage of computing, information science, systems modeling, and the social sciences. Harold Guetzkow, a Professor of Political Science, compares the utility of verbal (natural language) and mathematical (symbolic) models and simulations in the field of international relations. He writes:

> "As we move into the latter third of the Twentieth Century, it seems feasible to catalyze the consolidation of our knowledge about international relations through the use of simulations. Verbal efforts to present holistic integrations of extant knowledge are found in textbooks. . . Yet, their contents are theoretically vague and

> their databases are largely anecdotal. . . Mathematical formulations. . .are more partial in scope, even though they are explicit in structure and systematic in their grounding in data."[3]

It is interesting to note the value-laden language in this passage; he calls verbal models "theoretically vague" and "cumbersome," as if this somehow invalidates their utility. Guetzkow's conclusion is that computerized simulations are and will be very valuable to the study of international relations because they can amalgamate the best of the verbal and mathematical theories. Finally, he calls for general purpose simulation programming languages, with built in support for modularity, presumably so that individual researchers can build subsections of a grand simulation.

Carl Hammer, Director of Computer Sciences at UNIVAC, presents a list of astounding predictions about simulation and the information age. "Laboratories, as we know them today, may go out of style by 1993, as experimentation by computer simulation will be less expensive and more reliable. Laboratories will then *only be used to validate the research done 'on the computer.'* (emphasis added)"[4] He goes on to speak of a "cybernetic culture" characterized by control through simulation and systems modeling:

> "Cybernetics has been defined as the science of information processing, communication, and optimal control in complex, purposive, dynamically stable systems whose human elements provide feedback in a specified environment. While a culture in general, is a complex self-organizing system, cybernetic cultures will be characterized by the introduction of optimizing control mechanisms which react to slowly changing values so as to produce dynamic stability."[5]

This passage advocates the full extension of control methods, including simulation, to all aspects of social life, with the end goal of creating a kind of utopian "cybernetic culture."

Systems and Simulation in the Service of Society, a collection of papers written during the mid-to late 1960s and published in 1971 is filled with similar suppositions. Robert Frosch, an Assistant Secretary of the Navy, in one of the more cautionary papers entitled "The Sin of Simulation," states: "In summary, let me repeat that simulation . . . is essential. There is often no other way for us to learn about large and complex or dangerous and inaccessible situations." He goes on to note the dangers of simulation, but concludes, "However, more than ever, simulation is important. For we feel ourselves on the verge of being able as a society to tackle major societal and environmental problems."[6] This passage reflects not only the overly optimistic predictions regarding the promise of simulation characteristic of this period, but also a cursory dismissal of possible negative social consequences. This is an attitude we will repeatedly encounter; one that was perhaps most succinctly expressed in the keynote address to the TASASC by Lawrence Fogel:

> "Science is non-valuative, the intent being to gain a greater understanding of nature. Technology, a product of science, provides new means for the study of nature, including human behavior. It is entirely appropriate that we turn our attention to how science and technology might best be used to avoid unnecessary violence."[7]

Computers in Modern Society, published in 1974, a textbook intended to educate university level humanities majors about computers, devotes a chapter to computer modeling and simulation. The chapter is entitled "Using Computer Models to Reach the Best Decision." Note the use of the word "best" in the title. At one point, the authors discuss using computer models to simulate the behavior of human beings and social systems, a science they call "sociometrics." They say,

> "This newly arisen discipline attempts to formulate the rules and mores on which a society is built, and on that basis to establish a model that can be used for examining the outcome of various social events. This science, if and when it is refined, will be able to help man create his future with more foresight and accuracy than ever before. Of course, like any other knowledge, sociometrics could be perverted and used to the detriment of mankind. Whatever may happen, it is certain that this new field would be impossible without the aid of the computer."[8]

Here again we encounter a complacent attitude toward the negative effects of technologies on society. Note the disturbing similarity to the "Guns don't kill people -- people do" argument promulgated by the NRA. Further, we see parallels with Hammer's notion of a utopian cybernetic culture.

I have explored so deeply the writings on simulation of this time in order to provide an extensive context for the development of the Simula language. The most significant themes in the above discourse are fourfold: (1) simulation is inarguably the tool of the future, (2) there is a need for computer based simulation languages, (3) simulation itself is value free; any negative effects are entirely due its misuse, and (4) simulation is generally viewed as beneficent to society. This last theme is explicated in varying extremes, but the most extreme view holds that simulation and other control theoretical techniques can and should be applied to control the course of society. Collectively, I will call these themes the ideology of simulation. Simula must be viewed as a product of this intellectual/ideological movement.

1.2 Development of Simula

In focussing on Simula, I will show that the ideology of simulation was central to the development of this language. First, I will show how the principal designers, Ole-Johan Dahl and

Kristen Nygaard, were motivated by this ideology to develop Simula. Second, I would like to show how this ideology is reflected in the Simula language itself. Nygaard and Dahl were first exposed to computers during their conscript service, during which time they worked at the Norwegian Defense Research Establishment (NDRE). From the beginning, Nygaard was involved with simulation. Early on, he worked on Monte Carlo simulations for the design of Norway's first nuclear reactor; later, he became increasingly involved in the field of Operations Research.

In describing their motivations, the Simula designers state, "Simulation is now a widely used tool for analysis of a variety of phenomena: nerve networks, communication systems, traffic flow, production systems, administrative systems, social systems, etc." They go on to say that ". . . simulation programs are comparatively difficult to write in machine language or in Algol or Fortran. This alone calls for the introduction of simulation languages." In this passage we can see a reflection of the themes presented above -- because simulation is a useful tool we must develop simulation languages for computers. Their next statement deserves a little more attention: "However, still more important is the need for a set of basic concepts in terms of which it is possible *to approach, understand, and describe all the apparently very different phenomena listed above.* (emphasis added)"[9] If we analyze this statement, we again see another prevalent theme; namely that simulation, if we just do it right, can capture a very wide range of phenomena, ranging from physical processes all the way up to social systems.

We find more evidence of this underlying ideology in other sources. In a paper on Simula history presented in 1981 at the History of Programming Languages Conference, Nygaard and Dahl state:

> "In large scale Operations Research jobs, simulation once more turned out to be *the only feasible tool for analysis of sufficiently realistic models.* Also it became evident that no useful and consistent set of concepts existed in terms of which the structure and interaction of these complex systems could be understood and described (emphasis added)."[10]

Kristen Nygaard left the NDRE in 1960 to work for the Norwegian Computer Center, of this experience he states:

> "Many of the civilian tasks turned out to present the same kind of methodological problems: the necessity of using simulation, the need of concepts and a language for system description, lack of tools for generating simulation programs. This experience was the direct stimulus for the ideas which in 1961 initiated the Simula development."[11]

In these passages we time and again hear a strong conviction that simulation is the right way, indeed, maybe the only way to go, and that this course necessitates the creation of general purpose simulation programming languages. It is interesting to note the similarities between the sentiments expressed here by Dahl and Nygaard, and Guetzkow and Frosch above. All of these passages associate a nagging sense of urgency with expanding the application of simulation techniques. Finally, Kristen Nygaard makes an explicit reference to the traditional theme of value-free technology: "A main and, at the time, largely undebated assumption in the development of the post-war culture was that 'technological progress happens, it is politically neutral - and good! "[12] This assessment parallels the position taken by the authors of Kochenburger and Turcio.

1.3 Analysis

Next, I would like to focus on the Simula language itself and show how its foundational ideologies are reflected in its structures. In particular, I wish to focus on what is today seen as Simula's revolutionary contribution to computer science: the notion of activities and processes. In Simula-67, these were renamed as classes and objects. The object oriented approach, together with class descriptions, is today viewed as a radical departure from the traditional, procedure-data distinction made in languages of the Algol family. This new approach would later form the basis for the first fully object oriented language, Smalltalk.

In reading the original Simula papers, we discover that Nygaard and Dahl apparently did not see the radical implications of their new conceptions. Certainly, they point to the class/object distinction as an important innovation, but they do not pursue its potentialities. The question we must ask is, Why not? Why didn't Nygaard and Dahl see that they had laid the foundations for qualitatively new programming paradigm? The simple answer to this question lies with the fact that the development of Simula was heavily tied to contracts with Univac. As these contracts were for the creation of a simulation language, the designers vision was limited by a vision to satisfy their contract. A somewhat more interesting answer might lie in Simula's relationship to Algol. One might argue that because Simula was designed using an imperative language as a foundation, the designers could not see implications which extended beyond the imperative mode of thinking and looking at the world. This argument is limited however, because as the motivations discussed above will readily show, the designers were attempting to look beyond the imperative paradigm. In fact, it was exactly this desire to look beyond, which engendered the breakthrough; Nygaard states:

> "In the spring of 1963, we were almost suffocated by the single-stack structure of Algol. Then Ole-Johan developed a new storage management scheme [the multistack scheme] in the summer and autumn of 1963. The preprocessor idea was dropped, and we got a new freedom of choice. In Feb, 1964 the process concept

> was created, which is Simula 67's class and object concept."[13]

In other words, Algol was not sufficiently powerful to cleanly express the desire to create a simulation programming language. In truth, the class/object concept was a child of the ideology of simulation. Now, the answer to the original question is clear; just as the ideology of simulation was the driving force which urged Nygaard and Dahl to break out of the imperative paradigm, it was also the limiting factor which hindered their perception of the full implications of what they had created. Simula's use of classes and objects was the solution to a very local problem concerning simulation programming. Quite literally, the class/object innovation was a side effect of the development of the Simula language. The rest of the Computer Science community also missed the implication of this innovation:

> My last visit to the U.S. was in 1970. At that time the class concept only had a certain curiosity and entertainment value. . . Today, it's interesting and pleasant to observe that the situation is different. . . I still think that many people who refer to Simula only grasp parts of it. In fact, those who understand Simula best are not the people in programming research, but rather Simula's simulation users. The computer scientists like Simula as a vehicle for implementing abstract data types."[14]

In terms of a Kuhnian analysis, the dominant programming paradigm within Computer Science was the notion of imperative, sequential, programs which call procedures on data. Simula can be seen as a potential challenge to this dominant paradigm. In order to avoid crisis, the Computer Science community attempted to subsume Simula's class/object notion within their paradigm. They achieved this by focusing on how Simula might be useful to implement abstract data types. In this way, the computer science community made ad hoc adjustments to the predominant theory, and thereby sidestepped the true implications of Simula's class/object concept. Obviously, not everyone in the community viewed Simula in this way. As we shall see, the developers of Smalltalk, our second language of study, proved to be an important exception.

3. Smalltalk

3.1 Cultural Background

Now we turn to the vastly different story of Smalltalk. As we have seen, the design and implementation of the Simula occurred within the context of a very specific socio-scientific agenda, namely the drive to apply simulation techniques to a wide range of scientific, technical, and social problems. Smalltalk arose from the culture of the turbulent Vietnam-War era, or the "60s."[15] At this point the insightful reader will object that the creation of Simula also occurred during the 1960s. In fact, Simula was still under development as late as 1968, a year when things were really starting to "heat up" (student revolt in France, RFK assassination, MLK assassination, Democratic Convention in Chicago). So how can I argue that Simula and Smalltalk truly emerged from different socio-political and ideological backgrounds? I can argue this for three reasons: First, it is most important to look at when the conception of a language occurred. For Simula, it is clear that this occurred during the late 50s and very early 60s. Smalltalk, as we shall see, was conceived almost exactly a decade later. Second, there is an important generation gap between the respective designers. Biographical data tells us the following: Nygaard and Dahl were born around 1930, completed their theses around 1956, and became prominent researchers by the early 60s. Judging by the status of the persons cited in our discussion of Simula's cultural context, it is clear that they were contemporaries of Nygaard and Dahl. The Smalltalk generation, on the other hand, was almost exactly 10 years younger. Alan Kay was born in the early 1940s (1942 is my best estimate), received his PhD in 1969, and became a prominent researcher with Xerox Parc by the early 1970s. Third, as I will show below, the university/research community, especially as situated in the San Francisco Bay Area during the late 60s and early 70s, was undoubtedly a qualitatively different environment than that which existed in Norway during the early to mid-1960s.

While it is probably not necessary to refresh the reader's memories of this period, I will take a moment to point out a few general trends. Later, we will revisit these trends, but within the context of the design of Smalltalk. We take our definition of what constituted the 60s largely from popular culture. It was a time of the civil rights movement, student unrest, political assassinations, and rock music. Many people (especially the middle or upper class university students) saw themselves at a turning point in history. Estranged from their parents and the culture of WW II America, they sought to break free and change the status quo in order and make lasting changes which would hopefully yield a future more livable than the present or the past. This was a period of political and social awareness; it was a time when many sought to study and expose the structure, sources, and patterns of the power relationships which dictated their daily existence. In light of the problems they found, the suggested solutions were as numerous as the problems themselves. Some chose to escape and attempt to create radically new ways of living; others sought change the system form within; still others tried to confront it directly and perhaps destroy it. Whatever the means, the ends were often very similar: to forever alter the inequitable power relationships which constituted daily life. This categorization might make it sound like everyone was either marching on Birmingham, or growing organic lentils on a commune, or building bombs for the Weather underground during the 60s. While clearly not the case, we cannot deny that this attitude permeated the general culture in various ways.

Now I would like to narrow the focus somewhat to the level of the university campuses. Campuses in this nation became centers for unrest and debate. A survey conducted by a Department of Health, Education, and Welfare study group on the roots of campus unrest during 1969 found four major categories of concern:

1. dehumanization of society
2. inequitable distribution of wealth, power, and prestige,
3. social and cultural exclusion
4. educational irrelevance[16]

Raymond Tanter, in a study entitled "Comparative Statistics on Campus Turmoil" found 108 incidents of campus related unrest on 50 different campuses, involving as few as 4 people up to as many as 200,000, between May 1964 and August 1968.[17] Although this survey gives us an incomplete picture of what was going on (because of its limited time span), it is sufficient to give us a flavor of the atmosphere on university campuses during the 1960s.

This era was characterized by the realization that technology and knowledge played a critical role in the power relations. For some, the realization that we could apply information technology to many aspects of life held great promise and for others, great threats. Information was viewed as a new substrate of power,[18] and therein lay the great promise as well as the danger. Whoever controlled the information technology controlled the access to information, and therefore also power. Up until the late 1960s, due to its high cost, information and the technologies to process them were controlled largely by the government, large corporations and universities (who were generally supported by either the government or large corporations), implying control by a capitalistic oligarchy. As we have already noted, certain social and cultural groups (in particular, students) felt compelled to attempt to equitably re-distribute power, make education and research more relevant, and to re-humanize society. Computers became a major factor in this equation; while they acted as storehouses, gatekeepers, and processors of information (and by extension, power), they also dehumanized society by reducing it down to standard data which could be processed by a computer. In this way, the computerization of society was viewed by some as the next logical step in the dystopias foreseen by social theorists such as Marcuse, Horkheimer, Adorno, and Habermas, in which all social processes succumb to the instrumental reason of scientific analysis. These concerns were shared, in varying degrees of extremity and explication, by a large segment of the young, liberal-minded, social scene. I would like to present Smalltalk as an effort by a group of researchers to address some of these issues.

3.2 Xerox Parc and the New Computing

Xerox Parc, the birthplace of Smalltalk, was neither a university, nor a radical, hippie commune, set to reorganize daily existence around equality and personal freedom. In fact at the corporate level, Parc's purpose was purely capitalistic. High level executives and planners, having the foresight to see the eventual limits of Xerox's paper copier monopoly, and realizing that the future medium of information would be computers and not paper, set out to make profitable inroads into this potentially huge market. In order to develop the computer systems of the future, Xerox decided to follow a formula so successfully used by other groups, such as ARPA: Gather together some of the best, most ambitious minds in the field, add money, and sit back and wait-- like a farmer watching the corn grow.

Located at the very origin of the silicon revolution (rubbing shoulders with Stanford University), Xerox Parc was able to tap into a great talent base of creativity, knowledge, and motivation. While the Stanford environs have always been a rather peaceful place, it is important to note that the San Francisco Bay Area was arguable one of the most liberal, turbulent, diverse population centers in the country. Indeed, San Francisco and Berkeley went a great distance towards defining the "era" of the 1960s.

Just as Parc's location would be of benefit, so would be its timing. University labs, funded largely with ARPA money, had always supplied researchers with a work environment largely free from corporate politics, deadlines, and the requirement to see a direct return from investments. However, as the 1960s drew to a close, the federal government was beginning to feel that research should be more directly applicable national interests such as defense. They sought to achieve this through the Mansfield Amendment, which would restrict research to more "applications" oriented projects ("applications" is military-speak for "things that might be useful for fighting and winning wars"). With the possibility of only restricted access to research funds, university researchers began to see working in the corporate environment as a more palatable alternative. This gave Xerox access to researchers who would have never considered leaving the university setting, such as Alan Kay-- who would become head of the Learning Research Group at Xerox Parc, the sector responsible for the development of Smalltalk.

While Xerox Parc was created to fulfill purely entrepreneurial goals -- to do the research and invent the technology that would enable Xerox to monopolize the digital medium, just as it had the paper medium -- the climate on the inside was really quite different.   The Rolling Stone Magazine article, "Fanatic Life and Symbolic Death Among the Computer Bums," by a prominent free-lance writer, Stewart Brand, provides much information on the climate of the Learning Research Group. Brand focuses on what some of the researchers at Parc did with their spare time. He states: "How mass use of computers might go is not even slightly known as yet . . . One informative place to inquire is among the hackers, particularly at night when they're pursuing their own interests."[19] With his investigation, Brand is not trying to determine the future course of computers, but rather, the to uncover the hopes and dreams of the researchers who were then designing them. In this way, we gain some insight into the political and social underpinnings of the work done at Parc.

So what were the researchers at Parc doing in their spare time? Brand finds them writing programs which tally the accounts for communes or create astrology charts, building their own computers, and using word processing programs to automate the creation of anti-War form letters. An even more ambitious project, Research One, attempted to bring on-line computer access and information retrieval to the people of San Francisco and maybe beyond. The politics of Resouce One clearly had a leftward slant. The project's goals included automating the data retrieval for San Francisco's many free clinics, using government data to expose things like inequitable bank loan trends in low-income areas such as San Francisco's Mission District, and finally, educating the community about computers. Pam Hart, an organizer behind the project, says, "People want to know about computers, not how to use them, necessarily, but how they're used against them."[20] Alan Kay himself states, "A lot of time was spent outside of Parc, playing tennis, bikeriding, drinking beer, eating Chinese food, and constantly talking about the Dynabook and its potential to amplify human reach by bringing new ways of thinking to a faltering civilization that desperately needed it (that kind of goal was common in California in the aftermath of the sixties)."[21]

In his conclusion, Brand uses the early graphical game Spacewar as a parable of what he sees as the future of computing. He states:

> "It (Spacewar) was the illegitimate child of the mating of computers and graphics displays. It was part of no one's grand scheme. It served no grand theory. It was the *enthusiasm of irresponsible youngsters*. It was disreputably competitive . . . It was an administrative headache. It was merely delightful."[22]

He goes on to list future themes: real time interaction; user programmability; high bandwidth human interfaces; computers as communication media; games; personal, stand-alone equipment (as opposed to timesharing); and serving human, rather than computer, interests. We must remember that Brand's ideas are the products of his interaction with the computer sub-culture at universities and corporate research labs like Xerox Parc.

What is most important to extrapolate from this discussion is not that all computer scientists in the 60s were leftist radicals, bent on using computer power to bring down the establishment. We know that this is not the case. However, what is clear is that a widespread attitude existed amongst the younger generation of computer scientists that they were on the brink of what has been called a "new computing," or less dramatically, "personal computing." This attitude parallels and reflects a very similar trend evident in the greater cultural sphere of the late 1960s to mid 1970s. The new computing was a way to address (if not entirely consciously) the multitude of profound problems facing American society at the time. The new computing would help redistribute power by providing heretofore unimagined access to previously unimagined volumes of data. It would re-humanize the interactions of humans and computers, through games and innovative, "user-friendly" interfaces. Further, the new computing would encourage artistic and creative expression, through painting and music programs, as well as through as yet undiscovered artistic media.

Technically speaking, much work remained to bring about this new computing. The new computing would demand widespread, personalized access to computer cycles. By 1970, it was clear that the necessary hardware revolution was just around the corner: It was evident that by 1975, many of these dreams could run on computers which could fit on a person's desk; by 1985, in their backpack or briefcase; and by 1995, in the palm of their hand. Many such systems were either imagined or actualized by 1975 at Xerox Parc and other labs. However, hardware was just part of the equation.

And this, finally, brings us to software, and the language that would implement these dreams -- Smalltalk. I will show from a variety of sources that Smalltalk was a product of the goal to create a new, personal, computing, which in turn was a means of addressing the social and political concerns of the Vietnam War era.

3.3 Smalltalk: Philosophical Underpinnings

In the opening paragraphs of of "Personal Dynamic Media," Alan Kay and Adele Goldberg, two of the primary designers of Smalltalk, divulge their philosophical underpinnings and design motivations. The primary focus of this paper is the Dynabook-- the ultimate, portable, powerful, personal computer of the future. Smalltalk is seen as the "communication system" which will allow users full flexibility for the creation and manipulation of knowledge. Goldberg and Kay state:

> "Every message is, in one sense or another, a simulation of some idea. It may be representational or abstract. The essence of a medium is very much dependent on the way messages are embedded, changed, and viewed. Although digital computers were originally designed to do arithmetic computations, the ability to simulate the details of any descriptive model means that the computer, viewed as a medium itself, can be *all other media* if the embedding and viewing methods are sufficiently well provided."[23]

This passage strongly echoes the words of Marshall McLuhan. His famous slogan, "The medium is the message," radically altered the dominant notions of the relationships between media, messages, senders and receivers. His essential message was that the medium of communication has the power to shape our patterns of interaction in a powerful manner. This opinion ran strongly contrary to contemporary traditional notions in the communications field, which held that message itself constituted the essential aspect of the communications act. McLuhan's work in communications theory was perhaps most widely acknowledged during the 60s and it is clear from Kay's numerous references in this and other papers[24] that he was well versed in McLuhan's work. Finally, Kay's renaming of the Smalltalk programming language as a "communication medium," gives the Dynabook system a very different connotation. "Programming" implies a computer oriented process which entails rigid, strict, exact, and linear control, whereas "communication" implies a human oriented process which entails understanding and consensus and is often inexact.

In "Design Principles Behind Smalltalk," Dan Ingalls, another member of the Smalltalk team, discusses what appear to be design principles and motivations, but upon closer inspection, look a lot more like a philosophy of life: "The purpose of the Smalltalk project is to provide computer support for the creative spirit in everyone. Our work flows from a vision that includes a creative individual and the best computing hardware available." He continues, ". . . the human potential manifests itself in individuals. To realize this potential, we must provide a medium that can be mastered by a single individual. Any barrier that exists between the user and some part of the system will eventually be a barrier to the creative spirit."[25] Ingalls' focus on the potentialities of the individual and the need to foster the creative spirit were certainly prominent liberal themes during this period.

Reflecting on these times in the 1993 paper "The Early History of Smalltalk," Kay refers to the "new computing" when he says

> "It [Smalltalk] became the exemplar of the new computing, in part, because we were actually trying for a qualitative shift in belief structures - a new Kuhnian paradigm in the same spirit as the invention of the printing press - and thus took highly extreme positions . . ."[26]

Here Thomas Kuhn's ground-breaking work in the study of competing paradigms and scientific revolutions is blended with McLuhan's notion of the role of media in affecting the structure of language, communication, and thought processes.[27] The notion of revolution was another prominent 1960s theme, which we shall encounter time and again in Kay's writings. It is clear that Kay, Ingalls, and others believed they were in the midst of a revolutionary transition into a new era of computing.

3.4 Smalltalk: The Focus on the User

Another facet of the Smalltalk history that is interesting to this investigation is the focus on the user. The particular user group that received the most attention was children. Kay and Goldberg's proposed that children are much more demanding subjects than are adults. Adults have become socialized to be very patient with the computer; this is presumably because the computer is doing something very important (beyond their comprehension, of course) and we are just going to have to wait. Children, however, are different; they are less patient and therefore their actions require faster responses. Further, they are not willing to deal with simple teletype screens and keyboards or stifling user interfaces. They want to manipulate, explore, and create. Finally, children have not been socialized against failure to the same degree as adults, and therefore, are more willing to explore new domains. Kay and others in the LRG reasoned that these qualities made children the true "power users." Children would be a true test of the power, extensibility, and design limits of this new programming paradigm.[28]

Using children as test subjects also allowed Smalltalk to be employed as an experimental testbed for computers in education. Kay and company wanted to use Smalltalk in much the same way as Seymour Papert and LOGO-- to teach children to think deeply by letting them "teach" the computer.[29] Inherent in this approach to education is a strong critique of the contemporary notions of computer assisted instruction. Many CAI systems were directly descended from B.F. Skinner's training technique known as programmed instruction.[30] These systems were criticized by Papert and others because they were not seen as really doing anything but automating already flawed teaching methods (rote memorization, drills). This led to the slogan: Let the student program the computer, not the other way around. Smalltalk was to be used as a medium for exploring new educational possibilities, not a tool for programming the next generation of human beings:

> "The reason, therefore, that many of us want children to understand computing deeply and fluently is that like literature, mathematics, science, music, and art, it carries special ways of thinking about situations that in contrast with other knowledge and other ways of thinking, *critically* boost our ability to understand our world."[31]

The user-oriented focus of Smalltalk is interesting to our analysis in many ways. First, this focus on the user clearly indicates a desire to re-humanize computer technology. The metric of success was not compile time or how fast a Smalltalk program could execute the Towers of Hanoi or the Eight Queens, rather Smalltalk aimed to embody flexibility, beauty, simplicity, and extensibility. Kay compares Smalltalk to a fertilized egg, "that can transform itself into the myriad of specializations needed to make a complex organism [which] has parsimony, generality, enlightenment, and finesse -- in short, beauty."[32]

The only way to evaluate the success of this aim was to try it out on the most demanding users -- children. Second, using Smalltalk as a educational medium exposes a deep desire (as Kay noted above) on the part of the designers, to give children a deep understanding of computers: "my own emotional involvement has always been centered on personal computing as an amplifier for human reach -- rather than programming system design. . ."[33] We see yet another reflection of a prominent contemporary theme -- that children are a society's most valuable resource.

We must note that Smalltalk was never intended to be a language in isolation. It was conceived in conjunction with Alan Kay's dream machine, the Dynabook, as part of a complete design system. It is, therefore, erroneous to look at the history of the Smalltalk as separate from that of the Dynabook. Upon examining the Dynabook, we again discover many of the themes and values inherent in the Smalltalk system. In "Personal Dynamic Media," Goldberg and Kay describe an "interim Dynabook" (a Xerox Alto workstation). The interface to the system has been personalized through the addition of pointing devices, organ-like keyboards, high quality audio, and high resolution displays. They go on to describe some of the systems they have developed and tested using Smalltalk, the new medium for communication. Again, they focus on people-oriented programs, such as animation, music, drawing, painting, and word processing. There is little talk in this paper about processing power, clock speeds, code generation, or compile times. The Dynabook was meant to be the next step in communication media. Not only would it give the user power to access information, it would give them power to create, distribute, and manipulate that information. In this way, the Dynabook/Smalltalk system would fulfill the goals of democratization and distribution of information, and by extension, power.

3.5 Analysis

In analyzing Smalltalk's contribution to the study of Computer Science, Alan Kay states: "Smalltalk's contribution is a new design paradigm -- which I called object-oriented -- for attacking large problems of the professional programmer, and making small ones possible for the novice user."[34] Several important issues are addressed in this single quote. First, it reiterates an idea which we have discussed herein -- that Smalltalk is not simply a programming language in isolation, but part of a larger "design paradigm." Quite simply, the programming language should be be all-encompassing -- a sentiment expressed by Dan Ingalls: "An operating system is a collection of things that don't fit into a language. There shouldn't be one."[35] Second, Kay places equal emphasis on the needs of the professional and the novice, again stressing the theme of democratization of computers. Third, Kay considers Smalltalk's most significant contribution to be its essential object-orientedness.

Kay acknowledges his debt to Simula for first actualizing an object-oriented programming scheme. However, Kay and his associates clearly saw something in Simula's class/object concept which the rest of the computer science discipline dismissed:

> "The 'official' computer science world started to regard Simula as a possible vehicle for defining abstract data types. . .To put it mildly, we were quite amazed at this, since to us, what Simula had whispered was something much stronger than simply reimplementing a weak and ad hoc idea. What I got from Simula was that you could now replace bindings and assignment with *goals*. . ."[36]

Dan Ingalls, in noting the difficulties presented in explaining Smalltalk those who see programming languages in a restricted manner, states:

> "Smalltalk is not simply a better way of organizing procedures or a different technique for storage management. It is not just an extensible hierarchy of data types, or a graphical user interface. It is all these things and anything else that is needed to support [human-human or human-computer interactions]."[37]

Of course, Smalltalk's breakthrough was the elevation of object-orientedness to all-encompassing programming concept:

> ". . .the objects should be presented as *sites of higher level behaviors more appropriate for use as dynamic components*. . .The object carries with it a lot of significance and intention, its methods suggest the strongest kinds of goals it can carry out, its superclasses can add up to much more code-functionality being invoked than most procedures-on-data-structures."[38]

The question remains, why did Kay and company see the radical implications of Simula's use of objects and classes, when all others (including the Simula designers themselves) dismissed it?

The Parc group made this conceptual leap for a variety of reasons. First, I would point to the personal, political and social beliefs of some of the Parc researchers regarding the potential of computers in society. This becomes clear when we take note of the activities of some of Parc's researchers outside of work hours. In their involvement with communes, free medical clinics, or the anti-war movement, there was always a drive to apply computing technology in a socially beneficent way. Also, we must remember Alan Kay's stated desire to use computers to educate and amplify the human reach. These political or social concerns may seem incidental to establishing the connection between objects and classes as a good programming technique and objects and classes as a radically new paradigm. Indeed, these concerns may well be entirely incidental, but they do indicate the existence of a drive to look beyond the conventional and experiment with new ideas and possibilities. In other words, the social concerns of the researchers created a unique climate of lateral thinking. Second, the Parc group clearly saw themselves as part of a revolutionary historical period, during which time the world around them was in a state of unimagined change and transformation. Large segments of society appeared to be throwing out old ideas, breaking free of a conservative past, and attempting to rebuild from the ground up by adopting radical new viewpoints. It must be remembered that Parc's proximity to San Francisco placed it near one of the epicenters of 60s culture, which clearly influenced the local culture. Finally, inherent in Smalltalk was the idea that the language was bigger than the language itself. This recursive, self-referential ideology inherent in the Smalltalk system encouraged those working on it to look beyond traditional bounds. This manifested itself in many areas -- most notably the desire to unify the hardware and software of a computer system into something larger than the sum of its individual parts. Other extensions beyond traditional notions of programming languages which we have investigated include: the redefinition of a programming language as a communications medium; the re-evaluation of computer aided instruction; the exploration of new interfaces; and the inclusion of the human being as a valid, valuable, and essential component of any computer system.

4. Conclusion

In summarizing our Kuhnian analysis, we can view Smalltalk as the rise of a fully novel paradigm of computer programming. The designers of the Smalltalk system, in response to a crisis in the dominant belief structure, radically restructured the view of what constituted programming. A number of factors - not least of which was Simula's notion of quasi-parallel processes as objects and its use of classes to organize concepts - contributed to this crisis.

However, when we take a moment to look at what has become of Smalltalk, our Kuhnian argument falls apart, to some degree. The commercial release of a Smalltalk system in 1980, saw a very different language from the one originally intended by the Xerox group. Smalltalk-80 was clearly a language for expert programmers, with a high level of mathematical sophistication. Clearly, at some point, the original, idealistic goals of Kay and company fell by the wayside and Smalltalk became commercialized. By commercialized, I mean that the design focus shifted away from social and political concerns, to an interest in efficiency. By exploiting the ability of class hierarchies to organize knowledge and share code, the designers created a language which was promoted for its ability to facilitate extremely efficient software engineering. Lost was the powerful notion of a programming system which would amplify the human reach and make it possible for novices to express their creative spirit through the medium of the computer. In the spirit of this paper, we must ask ourselves why this happened? What happened to the noble goals of Smalltalk?

While the answers to that question go beyond the scope and intent of this paper, we may find some clues in the following passage, from Max Horkheimer, in 1947:

> Concepts have been reduced to summaries of the characteristics that several specimens have in common. By denoting similarity, concepts eliminate the bother of enumerating qualities and thus serve better to organize the material of knowledge. They are thought of as abbreviations of the items to which they refer. . . Concepts have become 'streamlined,' rationalized, labor-saving devices. . . thinking itself [has] been reduced to the level of industrial processes . . . in short, made part and parcel of production.[39]

Horkheimer is referring to the transformations due to the unleashing of scientific and technical criteria of efficiency upon the fabric of knowledge. As industrial society applies scientific rationalization to all regions of the social sphere, the structures of daily life become increasingly mechanized. The logical and frightening conclusion of this process, Horkheimer maintains, is the mechanization of knowledge and the very processes of thought themselves. It is interesting to note that object-oriented programming languages are most often promoted for their capacity for exactly this act of streamlining and organizing knowledge in a labor-saving manner. This is not to say, however, that Smalltalk (or its object-oriented paradigm), as a technological artifact, is the root of this issue. Rather, I would like to suggest that Smalltalk, and its original noble goals, were merely victims of a greater drive to apply the logic of scientific reason and the capitalistic notions of efficiency to the entire fabric of society.

Notes

[1] We should note that the notion of a Kunian revolution used in this paper really constitutes somewhat of a misreading of Kuhns work. However, I will use his language to provide an interesting, familiar framework for explaining and studying shifts in belief structures within scientific and technical communities.

[2] Knight, "Preface," p. xv.

[3] Guetzkow, "Simulations in the Consolidation," p. 12.

[4] Hammer, "The Future," p. 204.

[5]Ibid., p. 207.

[6] Frosch, "The Sin of Simulation," p. 10.

[7] Fogel, "Keynote Address," p. xix.

[8] Kochenburger and Turcio, *Computers in Modern Society*, p. 196.

[9] Nygaard and Dahl, "Simula," p. 671.

[10] Nygaard and Dahl, "Development," p. 440.

[11] Ibid., p. 440.

[12] Nygaard, "Basic Choices," p.2.

[13] Nygaard and Dahl, "Development," p. 483.

[14] Ibid., p. 485.

[15] A common misconception is that the "60s" ended with the close of that decade. I shall use the term "60s" to denote the historical period which actually extended from the mid- 1960s to the mid-1970s. This is reasonable because the socio-cultural trends we are discussing extended well into the decade of the 1970s.

[16] Valien, "Must We Change?," p. 6.

[17] Tanter, "Comparative Statistics," p. 33.

[18] Some authors who discuss this notion extensively include Marcuse, Horkheimer, Adorno. More recent authors include Lyotard, Baudrillard, Foucault, and Habermas.

[19] Brand, "Fanatic Life," p. 71.

[20] Ibid., p. 75.

[21] Kay, "Early History of Smalltalk," p. 14.

[22] Brand, "Fanatic Life," p. 78.

[23] Kay and Goldberg, "Personal Dynamic Media," p. 31.

[24] Kay, "Early History of Smalltalk."

[25] Ingalls, "Design Principles," p. 288.

[26] Kay, "Early History of Smalltalk," p. 1.

[27] Kuhn, *The Structure of Scientific Revolutions*.

[28] Kay and Goldberg, "Personal Dynamic Media."
[29] Turkle, The Second Self.
[30] Gotlieb and Borodin, Social Issues in Computing, p. 150.
[31] Kay, "Early History of Smalltalk," p. 29.
[32] Ibid., p. 15.
[33] Ibid., p. 3.
[34] Ibid., p. 25.
[35] Ingalls, "Design Principles," p. 298.
[36] Kay, "Early History of Smalltalk," p. 25.
[37] Ingalls, "Design Principles," p. 289.
[38] Kay, "Early History of Smalltalk," p. 25.
[39] Max Horkheimer, in *Weizenbaum, Computer Power*, p. 249.

References

Borodin, A. and Gotlieb, C.C. *Social Issues in Computing*. New York: Academic Press, 1973.

Brand, Stewart. "Fanatic Life and Symbolic Death Among the Computer Bums." In *Two Cybernetic Frontiers*, reprinted from Rolling Stone Magazine. New York: Random House Inc., 1974.

Fogel, Lawrence, J. "Keynote Address." *Cybernetics, Simulation, and Conflict Resolution*. Ed. Knight, Curtis, and Fogel. New York: Spartan Books, 1971.

Frosch, Robert A. "The sin of simulation." *Systems and Simulation in the Service of Society*. Vol. 1, no. 2. Ed. David D. Sworder, Ph.D. La Jolla: SCi, 1972.

Guetzkow, Harold. "Simulations and the Consolidation and Utilization of Knowledge about International Relations." *Cybernetics, Simulation, and Conflict Resolution*. Ed. Knight, Curtis, and Fogel. New York: Spartan Books, 1971.

Habermas, Juergen. *Toward a Rational Society: Student Protest, Science, and Politics*. Boston: Beacon Press, 1970.

Hammer, Carl. "The Future: Interactive Electronic Systems." *Cybernetics, Simulation, and Conflict Resolution*. Ed. Knight, Curtis, and Fogel. New York: Spartan Books, 1971.

Horkheimer, Max and Adorno, Theodor. *The Dialectic of Enlightenment*. New York: Continuum, 1972.

Ingalls, Daniel. "Design Principles Behind Smalltalk." *BYTE Magazine*. Vol. 6, no. 8. Peterborough: McGraw-Hill Publications, August, 1981.

Kay, Alan and Goldberg, Adele. "Personal Dynamic Media." *Computer*. March, 1977.

Kay, Alan. "The Early History of Smalltalk." From the Proceedings of the History of Programming Languages Conference II. In *ACM Sig Plan Notices*. Vol. 28, no. 3. Cambridge: ACM, March 1993.

Kochenburger, Ralph and Turcio, Carolyn. *Computers in Modern Society*. Santa Barbara: Hamilton Publishing Company, 1974.

Knight, Douglas. "Preface." *Cybernetics, Simulation, and Conflict Resolution*. Ed. Knight, Curtis, and Fogel. New York: Spartan Books, 1971.

Kuhn, Thomas S. *The Structure of Scientific Revolutions* 2ed. Chicago: University of Chicago Press, 1970.

Lyotard, Jean-Francois. *The Postmodern Condition: A Report on Knowledge*. Minneapolis: University of Minnesota Press, 1984.

Nygaard, Kristen and Dahl, Ole-Johan. "SIMULA -- an ALGOL-Based Simulation Language." In *Communications of the ACM*. Vol. 9, no. 9. Ed. Donald E. Knuth. September, 1966.

Nygaard, Kristen and Dahl, Ole-Johan. "The Development of the Simula Languages." *History of*

Programming Languages. Ed. Richard L. Wexelblat. New York: Academic Press, 1981.

Nygaard, Kristen. "How many basic choices do we really make? How many are difficult."

Tanter, Raymond. "Comparative Statistics on Campus Turmoil: 1964-1968." *Cybernetics, Simulation, and Conflict Resolution*. Ed. Knight, Curtis, and Fogel. New York: Spartan Books, 1971.

Turkle, Sherry. *The Second Self: Computers and the Human Spirit.* New York: Simon and Schuster, 1984.

Valien, Preston. "Must We Change for Change's Sake?" *Cybernetics, Simulation, and Conflict Resolution*. Ed. Knight, Curtis, and Fogel. New York: Spartan Books, 1971.

Weizenbaum, Joseph. *Computer Power and Human Reason: From Judgement to Calculation*. New York: W.H. Freeman and Company, 1976.