Charles N. Yood
Draft: SWEP Conference Presentation
cny1@psu.edu

> High-performance computing has become firmly established as the third mode of scientific research. Specifically, it has led to the development of *computational science* as a new methodology of scientific inquiry that complements and broadens the traditional methodologies of laboratory experimentation and theoretical analysis.[1]

We have looked at issues surrounding:

Effects of funding on scientific and engineering disciplines as well as gender issues within S&E. In contrast (as a historian) I'm interested in the creation of disciplines ***within* science and engineering**.

I argue that computers are a unique scientific technology in that they have spurred the creation of entirely new scientific disciplines and new methodologies for scientific investigation. Cyclotrons did not produce "cyclotron *science*" nor did particle accelerators create "particle accelerator *science;*" computers however produced computer *science* and computational *science*.

To start, I need to explain to you the difference between these two disciplines. Give some wood to a computer scientist and he (or she) will test its strength, check it for knots, and measure its length. Do the same with computational scientists and they will build you a cabinet. Computer and computational scientists work at the cutting edge of technology and their research and products are seen as crucial to the security and economic competitiveness of nations. Computer science began to coalesce in the early 1960s, and still today has difficulties defining itself. In contrast, computational science

---

[1] Alan Schriesheim, to James F. Decker. "A Parallel Supercomputing Facility for Computational Science," March 22, 1990. Meetings with Don Austin, Friday April 13, 1990, box 2, MCS Archives, Argonne National Laboratory. Emphasis original.

emerged almost full-grown -- like Athena from the head of Zeus – twenty years later, in the mid-1980s and has been able to provide cogent arguments not only for its distinct identity but also for the uniqueness of its methods of investigation. Part of my project has traced the different tactics and fortunes of these two disciplines as they strove for existence, recognition, and independence.

I want to spend the majority of my time today talking about the emergence of computational science, but some background on the development of computer science as an independent discipline is necessary in order to compare the different professional trajectories.

Computer Science as a Discipline

In the late 1940s computers were perceived as ultra-fast calculators that could be used by scientists to solve complex problems that were analytically intractable, but that could potentially be solved using numerical approximations. Almost immediately however, the people engaged in the construction and operation of these new electronic digital computers realized that their efficient use required a corresponding effort in mathematical research. This marked the beginnings of the different activities that would eventually become computer science.

Part engineering and part mathematics, computer science did not fit neatly into either of the two traditional categories of theoretical or experimental science. The hybrid nature of computer science left its practitioners in a difficult position when trying to establish an independent disciplinary identity. These tensions are clearly evident in when I looked at the development of computing activities at Argonne National Laboratory.

Here, Argonne National Laboratory was critical for my study.

1. It had a long history of computation, going back to 1948. Argonne was the first national laboratory charged with the primary mission of developing nuclear reactors for civilian use. Computationally intensive.
2. Multidisciplinary, so its computer experts interacted with scientists from a wide variety of disciplines.
3. Spun off their computing services in 1957 into the Applied Mathematics Division. This provides a nice case study to follow the field of computer science as it developed beginning in the 1960s.

Early on, the directors of the AMD envisioned a new role for applied mathematicians scientists and engineers in the development of mathematical models suitable for digital computers. In particular, it was believed that "computational science" required that applied mathematicians be incorporated more directly in all stages of scientific and engineering practices -- from problem formulation to defining what constituted a solution. Arguments in favor of such a collaborative effort drew on Cold War rhetoric, debates within the mathematical profession, and issues surrounding the increasing quantification of the sciences.

Biggest issue, though, was how to integrate computers, mathematicians, and scientists into collaborative projects.

- mathematicians wanted to be seen as equal partners in collaborations
- Users of computers tended to see them as technicians providing a service.
- The latter belief was instantiated in the way computing activities were funded. AMD made money by charging users for services. Built into this charge was

a small overhead that paid for the research activities of mathematicians and budding computer scientists. For a variety of reasons, this funding structure served to hinder the development of computer science as an independent discipline.

Lacking a strong disciplinary identity of their own, AMD researchers at Argonne initially relied on physicists, chemists, biologists, and metallurgists to define the areas where theoretical work in machine computation needed to be done. In essence, the fundamental research priorities for the Applied Math Division emerged from its work in helping other disciplines conduct research. On the bright side, if mathematicians and scientists weren't equals, at least the worked closely together. During the 1960s, however, applied mathematicians and nascent computer scientists began to stake out a new, shared research agenda that balanced their service to scientists and engineers with conducting research in the foundations of mathematics and the theory of computing. In the process, disciplinarity erected barriers of its own that impeded the creation of the kinds of collaborative research projects that had been envisioned earlier in the decade.

Problems within the CS discipline, 1970s

Discipline-building remained a work in progress. Throughout most of the 1970s and early 1980s, computer science was, according to its practitioners, in a perpetual state of "crisis." Certainly the persistent inability of the discipline in the early 1970s to define itself and its agenda limited its ability to attract substantial federal funding.

a. NSF funds obligated to research in computing in fiscal year 1972-73 were $9.9 million, down from $12.5 million the year before and were projected to decrease further. At the same time, funding for the other scientific research programs at the NSF increased by 7.2%.[2] In fact, the NSF, which provided most of the funds for computer science research outside of mission-oriented agencies (like DOD and AEC), did not even recognize CS as a distinct discipline with its own funding category.

b. Within the Atomic Energy Commission, the situation was similar. In the case of Argonne, funding for the AMD was cut from $1.16 million in 1972 to $0.9 million in 1973, much of it from the research component of the division.[3]

In 1979, the Association for Computing Machinery produced a report entitled "Rejuvenating Experimental Computer Science."[4] Although the Feldman Report, as it became known, seems to have faded into history, in many ways it can be seen as a landmark in the history of CS because it laid out a blueprint for how to repackage computer science to make it more appealing to funding agencies.

In particular, the report located the weakness of computer science in its inability to articulate clearly how its research was applicable. It therefore called for the revitalization of the **experimental** branch of computer science and a deemphasizing of theoretical pursuits.

---

[2] Ibid.

[3] R.V. Laney, to Edward H. Levi. "Response to the Report of the Applied Mathematics Division Review Committee," March 30, 1973. folder 2, Box 53, Records of the AUA, University of Illinois Urbana-Champaign.

[4] Jerome A. Feldman and William R. Sutherland, "Rejuvenating Experimental Computer Science: A Report to the National Science Foundation and Others," Communications of the ACM 22, no. 9 (1979).

Emphasis on experiments is important for several reasons:

1. Produced tangible products: time-sharing systems are the most obvious. Office automation systems which linked hardware and software from areas like graphics, document preparations, database manipulation, communications technology, and networking each had their origins in university research and especially from the subfield of Artificial Intelligence.[5]

2. Experiments required new and expensive computer systems. Over the preceding two decades, investments in new computers for research purposes had declined precipitously, with potentially dire consequences. More importantly, computer systems for experimentalists by necessity must be different from commercial production machines. While the exploration of novel architectures could lead to more innovation, it was also important that the computer not be engaged in production work, because then it would lose its value as an experimental machine.

3. Finally, the Feldman Report directly linked experimental computer science research to the intellectual, economic, and military strength of the United States. Unfortunately, top people were going to places like Xerox PARC, General Motors, and Bell Laboratories because these companies had state-of-the-art experimental computing facilities. The result was that in 1979 over two hundred faculty positions in computer science went unfilled, and this, in turn, jeopardized the future of computing in the U.S especially in terms of training the next generation of computer scientists.[6]

---

[5] Ibid.: pp. 497-98.

[6] Ibid.: p. 499.

Computational Science as a discipline

In the mid 1980s, a growing number of scientists from across the disciplinary spectrum began to refer to themselves publicly as “computational” scientists. According to these researchers, computer simulations had become so sophisticated that they were no longer just illustrations; they were able to provide new insights into biological and physical phenomena. The use of simulations, these self-named computational scientists claimed, constituted a “third” branch of science that complimented, but was distinct from, theory and experiment.

One of the main spokesmen for computational science was the Nobel Prize winning physicist Kenneth G. Wilson from Cornell University. Wilson immediately provided caché in science funding circles which no computer scientists could match. Wilson felt that the best way to define computational science was by contrasting its core activities with those of experimental and theoretical science. Experimentalists, he argued, “. . . are engaged in measurement and the use (if not the design) of scientific instruments to help make measurements; they are concerned with the design of controlled and reproducible experiments and with the analysis of errors in these experiments.” “Theoretical scientists,” on the other hand, “are concerned with relationships among experimental quantities and the principles (such as Laws of Nature or symmetry principles) which underlie these relationships, and with the mathematical concepts and techniques needed to apply the principles to specific cases.”[7] The core interests of computational scientists, however, were different. They centered on “. . . the algorithms which define computational methods for solving scientific problems, the computer

---

[7] Kenneth G. Wilson, "Grand Challenges to Computational Science," (Cornell Center for Theory and Simulation in Science and Engineering: 1987), p. 2.

software needed to implement these algorithms, the design of computational experiments and the errors in these experiments, the basic laws or models for which the computations are defined, and the mathematical framework underlying the computations."[8] Wilson acknowledged that basic algorithms were a common feature of both theory and experiment, and that computer programs to execute these on standard computers were easily written. But the computational scientists did not use standard computers; instead, they used supercomputers which required "long experience or professional training…making it appropriate to think of computational science as both a separate mode of scientific endeavor and a new discipline."[9]

Wilson also devoted a section of his paper to a defense of the supercomputer as a scientific instrument, arguing that regardless of the expense, funding must be made available in order to advance the frontiers of computer technology. Unlike traditional experimental equipment such as microscopes and telescopes, supercomputer simulations could be used to see *into the future* -- for example in weather forecasting -- something no other instrument could do. Supercomputers could also be used to see into the past, reconstructing events like the Big Bang. Moreover, supercomputers could be used to explore objects on smaller scales and at shorter time intervals than those used in traditional experimental equipment. And finally, supercomputers were indispensable for evaluating models or explanations of experiments, especially if the model were too complex to be solved analytically.[10]

The question I ask, then, is why did computational science emerge suddenly in the mid-1980s? Scientists had been conducting simulations on high-performance computers

---

[8] Ibid., pp. 2-3.
[9] Ibid., p. 3.
[10] Wilson, "Grand Challenges to Computational Science," pp. 5-6.

since the early 1960s, but it would take another twenty-five years before “computational science” exploded onto the scientific scene. Here, I contend that the answer lies in events external to science.

Fifth Generation

In October 1981 Japan’s Ministry of International Trade and Industry (MITI) announced a new $500,000,000 national project to develop, within ten years, revolutionary new computer systems, some of which incorporated artificial intelligence. Dubbed the “Fifth Generation Computer Project,” the initiative was designed to develop innovative computer architectures substantially different from traditional von Neumann machines, and then apply these new machines to “cultivate information itself as a new resource comparable to food and energy.”[11] **Quickly discuss computer “generations”. See footnote.**

For the American computer science community, the Japanese initiative became a rallying point for several reasons. Most importantly, Fifth Generation had all the elements of an experimental project that was intended to develop and apply the new kinds of supercomputers.

The importance of supercomputing technology to the professional arc of both computer and computational scientists cannot be underestimated. Although supercomputers make up a very small percentage of all scientific computing (both equipment and its usage), their centrality to advanced scientific research made them an especially significant technology. While new architectures and components led to faster

---

[11] David H. Brandin, "The Challenge of the Fifth Generation," Communications of the ACM 25, no. 8 (1982): p. 509. In modern computing parlance, first generation computers were those that used vacuum tubes (1940s-1950s); second generation were those that used transistors (late 1950s-1960s), and third generation computers used integrated circuits (1970s- present). Thus, the decision to choose the title “Fifth Generation” is a clear indication of the Japanese desire to leapfrog current computer technologies.

machines, at least half of the speed improvements over the years have been due to better programming techniques and tools. What this meant was that while the primary users of supercomputers were scientists and engineers, computer science research was absolutely critical to making these machines efficient. Thus, the timing of the Japanese supercomputer initiative could not have been more fortuitous for computer scientists; they moved quickly to emphasize that supercomputers were precisely the kind of experimental facilities they needed.

As previously noted, supercomputers are also important because they are the primary tool of the computational scientists. Thus, on the heels of the Japanese announcement, both computer and computational scientists were united in calling for access to cutting edge computers. Access was certainly an issue. In 1982, there were about sixty-one supercomputers in the world: forty-two in the U.S. (and only three of these at universities); seven in England; six in Germany; four in France; and two in Japan.[1] The small number of computers meant that access was often very difficult. But an equally significant impediment to their use was the price; even if a researcher could gain access to a supercomputer at a national lab, his or her academic budget bought little time on it, as fees for their use typically ran $2,000 or more per hour.[12] The Japanese effort to foster a domestic supercomputer industry transformed the United States supercomputer industry into a national resource. The rising tide of money that poured into supercomputing raised all ships, although computational scientists would rise faster and higher than their computer science colleagues.

The Federal Response to the Fifth Generation

---

[12] Ibid.: p. 293.

It became clear to federal funding agencies that supporting supercomputing might kill three birds with one stone; it would help protect the domestic supercomputing industry in the face of the Japanese challenge; it would provide increased access to these machines for computer and computational science research; and it would provide the experimental facilities needed to attract the best computer scientists to academic work where they could train the next generation of computer scientists.

By the middle of 1984, money was beginning to flow into supercomputer projects at NSA, NSF, and Department of Energy (DoE)[13] Over the next five years, a series of studies as well as several federal agency-based programs to facilitate computational science research culminated in the creation of the High Performance Computing and Communications Program (HPCC). (1991 and 1993). The four-year, $4.7 billion federal program seemed to be a windfall for both computer and computational science. (remember, 5th Gen was $500 million over ten years)

At the heart of the HPCC were several key assumptions that guided the entire program. First and foremost was the belief that high-performance computing and communications were essential to national security as well as to the future economic strength and competitiveness of the United States. Second, the program was framed in the context of scientific and engineering “Grand Challenges.” Grand Challenges were “fundamental problems in science and engineering, with broad economic and scientific

---

[13] The Department of Energy was the successor agency of the Energy Research and Development Agency (ERDA) which itself was the successor of the Atomic Energy Commission. In 1974, the Energy Reorganization Act divided the responsibilities of the Atomic Energy Commission into ERDA and the Nuclear Regulatory Commission (NRC). ERDA handled research and development; the NRC dealt with regulation. Then in 1977, the Carter Administration signed the Department of Energy Organization Act which transformed ERDA into the modern-day DoE.

impact that could be advanced by applying high performance computing resources."[14] If the nature of the problems to be addressed seemed open-ended, their manner of solution was not: computational science was a solution in search of a problem. The third guiding principle was that research projects using high-performance computers would be *collaborative* ventures. Scientists and engineers would work closely with software and systems engineers and algorithm designers from industry, academia, and government laboratories to solve Grand Challenge class problems. They would be supported by a shared computational and experimental facility including professional software engineering support teams.[15]

It should not be surprising that such a large program meant different things to different people. HPCC supported particular lines of technological development (parallel computers and high-speed networks) and endorsed a particular organization of labor to solve Grand Challenge problems. Collaboration was mandatory; interdisciplinary teams of researchers in academia and government laboratories were expected to work closely with their counterparts in industry to explore new computer technologies and then apply these to problems the state deemed important. (climate modeling, ignition studies, remediation, nuclear stockpile reliability studies, material design) By supporting such work, the federal government absorbed much of the risks involved in developing new computer architectures while also speeding up the process by which experimental machines became available to scientists, engineers, and industrial users.

---

[14] Wilson originally coined this term in 1987 in a document he drafted while at Cornell. This article was later published here: Kenneth G. Wilson, "Grand Challenges to Computational Science," Future Generation Computer Systems 5, no. 171 (1989).

[15] Bromley. The Federal High Performance Computing Program.

But beyond the technology, HPCC represented a significant national investment in computational and computer science. For the former, the program officially recognized computational science as a distinct third methodology for doing science, and that it was an inherently collaborative and interdisciplinary practice. That computational science was crucial to issues like national security, economic competitiveness, and the U.S. leadership position in science and technology was now firmly established. Also no longer in question was the symbiotic relationship between computational science and supercomputing technology. Although supercomputers still made up only a small percentage of the overall computer market (sales and usage) it was clear that money needed to be pumped into this cutting-edge technology if advances were to continue.

For computer scientists, HPCC offered a solution to many of the problems that had plagued the discipline for decades. First, it represented by far the largest amount of financial support ever directed towards computer science research, and second, the organization of HPCC proposed interdisciplinary collaboration among equals – something that had great professional appeal.

At the same time, though, HPCC made a significant statement about the relative status of the two disciplines. While computational science was recognized as an independent discipline, the status of computer science remained unclear. Moreover, computational science far outstripped computer science in terms of prestige and power. This is manifest in the fact that by the beginning of the 1990s, computational science had reshaped national priorities and science policy. I contend that this is largely because computational science appealed to politicians, science advisory boards, and program managers at federal funding agencies; it invoked a collaboration between technology and people focused on solving problems that were important to the state. The computational science agenda was oblivious to details like disciplinary boundaries and individual research agendas; what mattered was making sure the different parts – human and computer -- worked together correctly in order to achieve the desired results.

Finally, I suggest that the values and methods of computational science co-opted computer science research for its own ends. Though my assertions are based on evidence from Argonne, I believe my conclusions are applicable to academia as well.

Effects of HPCC on Computer Science Research at Argonne National Laboratory

Despite the rhetoric of equality that characterized HPCC, it was clear that true equality would put too many cooks in the kitchen. This point did not escape Rick Stevens, the new Director of Argonne's computer division and the person responsible for implementing HPCC programs at the Lab. "The fundamental challenge in the 'Grand Challenge,'" he mused, "[was its] interdisciplinary nature."[16] Hierarchy quickly became a subtext in the rhetoric of computational scientists as they strove to articulate the characteristics of their discipline to sponsors. In December, 1990 the Department of Energy's Scientific Computing Staff (SCS) received a White Paper entitled "Grand Challenges, High Performance Computing, and Computational Science" in which the authors sought to make a case for an independent computational science discipline while also drawing sharp distinctions between itself and computer science.

> "Computational science is necessary because, although it incorporates elements from applications disciplines, applied mathematics, and computer science, it does so in a way which results in a unique approach to science and engineering. Furthermore…computational science is not computer science. In computer science, computers are the end objective. In computational science, computers are a means to accomplish other scientific and engineering objectives. Computational science is also not simply applied mathematics. While drawing heavily on applied mathematics for methods

---

[16] Rick L. Stevens. MCS and the Washington Plan. Argonne: Argonne National Laboratory, 1990. MCS Files -- General, box 1, MCS Archives, Argonne National Laboratory.

> and algorithms, computational science adds a strong linkage to observation and an intimate understanding of applications disciplines which is not found in applied mathematics."[17]

Moreover, the composite nature of the computational approach served not only as a bridge between applications and computers, but also between disciplines. Groups of computational scientists, they argued, have at least as much in common with each other as with their colleagues in their "home" disciplines, be it biology, chemistry, or physics.[18]

More telling was the assertion by the authors that computational science "should be developed from the scientist's perspective" and "computing environments should be built outwards from the scientist's desk." The underlying message was that the needs of computational scientists would dictate the kinds of projects that computer scientists would pursue. The computational science machine, while preaching collaboration, was without question hierarchical and computational scientists were at the top.

The Advanced Computing Research Facility

In mid-1982, the research component of the Applied Mathematics Division was reorganized into the Math and Computer Science (MCS) Division. Its first director, Paul Messina, wanted to find a new avenue of research for his mathematicians and computer scientists and was very interested in parallel architectures on which the next-gen supercomputers would be based.[19]

Under his leadership, the MCS established the Advanced Computing Research Facility (ACRF) in 1986 as a laboratory for testing different models of parallel

---

[17] Ibid.

[18] Ibid.

[19] Personal Interview with Paul Messina by author, May 13, 2002 at Argonne National Laboratory; Interview with Jim Pool by Tom Haigh, CalTech, July 14, 2004.

computers. Within two years, the facility had seven experimental parallel machines available to researchers.[20]

Although the ACRF was intended primarily for computer science research, it was purposely organized to allow other scientists to get their hands on these computers. Messina believed that the value of the enterprise hinged to a large extent on turning out real work, so there was a strong effort to encourage scientists to bring their problems to the ACRF, where they could collaborate with members of the MCS on these machines. Casting the facility in this light also allowed him to head off criticism that the facility did not contribute to the computing needs of Argonne scientists.

Signs that computational science promised to alter the activities of the MCS, and especially its Advanced Computing Research Facility, became apparent even in the earliest planning stages of HPCC. Beginning in late 1987, the Scientific Computing Staff in the DOE's Office of Energy Research set aside a third of its annual budget for Grand Challenge projects. This dramatic shift in the SCS budget suggested to members of the Math and Computer Science Division at Argonne that a broader reorientation of computer science funding was imminent.

Increasingly, the ACRF was pressured to use their parallel computers to do more "production" work at the expense of doing research on novel architectures. DoE managers of that program made it clear that they were "unhappy with the ratio of the profile to results of the ACRF" given the level of support provided to the facility.[21] It

---

[20] William J. Cody, to E. Vanberkum. "Major Accomplishments for MCS 1983-88," June 22, 1988. Major Accomplishments 1980-1994, box 1, MCS Archives, Argonne National Laboratory, Messina, to "Advanced Computing Research Seeks Answers to Questions Raised by Multiprocessing,"

[21] Kaper. Notes from Don Austin Visit.

was their opinion that the Argonne unit was not working on the right kinds of problems -- “crack in the airplane versus noble prize” -- and the former made for weak PR.

True, the large sums of money involved in HPCC placed intense pressure on its managers to produce results. But the very nature of the problems to be addressed by the program -- those that were “previously intractable” -- meant that “results” might be very difficult to attain. For the SCS managers, it seemed their best chance for success lay in reorienting the entire computer science and mathematics research program to align with the goals of HPCC. That computer science would be subservient to computational science imperatives can be seen in the revised criteria by which Grand Challenge proposals were evaluated. In order of priority, they were: a project’s fundamental significance in terms of economic, social, and/or scientific impact; its contribution to international competitiveness; its applicability to the DoE mission; its ability to generate needed technologies; and finally, its interdisciplinary approach.

After a meeting at DoE headquarters in April, 1990, Hans Kaper, the MCS director, reported that SCS managers saw themselves as “spending a whole lot of money at ANL for parallel computing and… not winning accolades.” Kaper went on to explain, “ [The] complaint is that the ACRF needs to be more visible (with results) and that one way to do that is by playing the biggest hardware game or the strangest hardware game.”

The values of computational science, which promised to produce solutions to problems important to the state, permeated the highest levels of federal funding agencies. The reorientation of computer science research followed quickly. By November of the same year, 1990, Kaper announced:

> “The direction of the AMS (Applied Mathematics section) has changed. It is being driven by the HPC Program. Basic research in mathematics and computer science will be supported only if it contributes to the objectives of the high-performance computing

> initiative. The HPC initiative will not lead to an expansion of the core research program. It has its own short-term objective. If the MCS chooses not to align itself with the new direction, its budget will certainly decrease."[22]

In general, after 1990, it became much more difficult for members of MCS to pursue computer science research that was inspired by issues within the discipline. Instead, the structure and goals of HPCC pushed them towards solving problems that originated outside of the discipline. Specifically, computer scientists were pressured to assume duties they thought they had shed long ago, namely doing service work for other scientists. As a result, discontent was rife within the MCS and was reflected in Kaper's final report to the Division's Review Committee before he resigned as director in 1991:

> "Traditionally, the MCS Division's research program has been discipline driven. We concentrate on the development of methods, algorithms, and tools and operate principally within the applied mathematics and computer science community. New program managers in DoE's Applied Mathematical Sciences program, which funds most of our activities, have indicated that our program needs to become more applications oriented, maybe even applications driven. This change is causing strains in our relationship with DoE, which the committee may wish to analyze."[23]

Conclusion

One of my goals in this project has been to explore the process by which disciplinary identities are established in science, and I think that the disparity in the fortunes of computer and computational science can shed light on this issue. In both cases practitioners based their claims to disciplinary independence on the technology of computing. However, computational scientists assumed the leadership positions within collaborative, interdisciplinary research projects and computer science became

---

[22] Hans G. Kaper. Meeting with Scientific Computing Staff, Doe Headquarters, Germantown; Nov 7, 1990. Argonne: Argonne National Laboratory, 1990. Program Presentations to Cavallini and Johnson, Washington 11/7/90, box 2, MCS Archives, Argonne National Laboratory.

[23] Kaper, to "Issues for Discussion by the MCS Division Review Committee,"

subordinate to computational science. What I would like to do now is speculate a bit about why computational scientists succeeded where computer scientists failed.

I suggest that the inability of computer scientists to articulate a coherent agenda has been the main impediment to achieving disciplinary recognition. In part, this failure arises from the unique nature of computer science. Despite efforts to devise the best methodology for programming, to a large extent computer science is still a craft practice. Because it blurs the distinctions between science, engineering and craft, computer scientists have had trouble agreeing on what ought to be done, how important it is, how it should be solved, or even what constitutes a solution.[24]

Seen in this light, computer science might be an example of what historian of science Theodore Porter calls a “weak” scientific community because it lacks “widely shared assumptions and meanings.”[25] In the sciences, weak communities struggle until their practitioners are able to establish those elements that are recognized externally as scientific. The mathematics of computer science seemed to offer an objective basis for the construction of new knowledge, but the strong presence of both craft and engineering practices ultimately militated against its acceptance as a science. This weakness, I argue, persisted until computer science was given a boost by the Fifth Generation. At this point, the expertise of computer scientists was seen as crucial to fulfilling the larger goals of the state and its disciplinary status was put on the back burner. National needs produced increased funding for computer science research and training and this, in turn, amounted to a state-sanctioned endorsement of computer science as an independent discipline. Similar closure, however, has not occurred within the discipline itself.

---

[24] Mahoney, "Computer Science: The Search for Mathematical Theory."
[25] Porter, Trust in Numbers: The Pursuit of Objectivity in Science and Public Life, p. 228.

In contrast, computational scientists came from established disciplines like chemistry, physics, and meteorology. Consequently, they did not have to face issues of scientific credibility because their authority was already established. Ken Wilson brought to the table more than a Nobel Prize; he was also a physicist and thus possessed a cultural and scientific authority which eluded computer scientists. Thus, the arguments of computational scientists found a receptive audience within funding agencies in part because these agencies had long histories of dealing with the physical and natural sciences.

More importantly, while computer science could produce only a fuzzy agenda, computational science aligned its agenda with the needs of its sponsors. I suggest that the rapidity with which computational science became established as a discipline had much to do with its clear focus on applications that the state valued. Computer science promised to improve the efficiency of computers; computational science and its methods promised to improve the efficiency and productivity of *science*.