

## False Frankensteins: The Costs and Illusions of Computer Mastery

Andrew Sabl

UCLA School of Public Policy and Social Research

Since the beginning of the computer age, people have feared that computers will grow to subjugate their makers. Popular views of computing's future have been shadowed by two mythic figures. First, Frankenstein's monster, with his warning to Doctor Frankenstein, "you are my creator, but I am your master; obey!" (Shelley 1965, pg. 160) Second, the sorcerer's apprentice: we know how to make machines aid us, but fear we will not be able to make them *stop* helping us. As computers have grown in power they have also grown in mystery, their workings becoming ever more opaque to the average user. As computers become ever more useful we fear we are being used. In relying on computers we fear becoming dependent on them. To paraphrase HAL, the renegade computer in *2001: A Space Odyssey*, "I can tell we are very upset about this."

This fear, however, is completely backwards: a psychologist might call it projection. For of course computers are not our masters but our perfect slaves. They perform exactly the steps we command. They carry out only the tasks they are told with neither initiative nor rebellion. They depend on human beings for everything: inputs, memory, physical space, and in the last analysis, electric power. (There is no computer that cannot be unplugged.) This paper, then, shall explore a very different set of dangers from those we often mention: instead of the horrors of *becoming* slaves, what we should really fear from computers are the corruptions involved in *owning* slaves.

In what follows, I shall focus on a criticism of slavery that is not the usual criticism. In both philosophy and actual political debate, the most compelling arguments against slavery are of course *moral* arguments. Slavery now seems wrong for a whole host of reasons, including reasons agreed to by every school of philosophy and popular morality. Slavery deliberately causes human beings great pain; it is the ultimate affront to values of autonomy and human development; it violates the slaves' rights to life, liberty and happiness; it flagrantly violates political equality and the rights of citizenship; it shuts a class of people out of the public debate; it does not consider everyone's interests equally; it tends to embody the radical subordination of people based on group characteristics, enshrining hierarchy rather than commonality. As Bernard Williams has pointed out, we no longer have the

classic ancient excuse for slavery—that it was necessary. Knowing very well how to manage a society without slavery, we can no longer answer the myriad arguments against slavery (Williams 1993, pg. 103-129).

These moral arguments seem obvious and overdetermined. Just as obviously, however, is that none of them applies to computers. Computers feel no pain, have no will, do not have moral faculties, cannot be citizens, and lack the capacity for moral argument.<sup>1</sup> Therefore, in arguing that our use of computers still embodies some of the dangers and corruptions of slave ownership, not only metaphorically but in some ways quite literally, I shall have to appeal to a different set of arguments: arguments which attack not the deprivation of slaves' rights but the debasement of slave-owners' characters. Doing this can shed light not only on the narrow question of computer usage but on elements of the master-slave argument that are often—and rightly—neglected in debates on human slavery. While it is a little obscene to focus on the problems of slave-owners when the condition of slaves themselves cries out for moral denunciation, those problems remain, and become the focus of legitimate inquiry, when the moral and physical sufferings of (human) slaves are not at issue. In this changed context, we may turn with a new sense of relevance to Hegel's argument that masters lack their slaves' knowledge of how the world works and how it can be transformed, and Aristotle's claim that the master-slave relationship was nothing very noble because, as a relationship between gross unequals, it could not embody the goods of politics or friendship. To the extent that they have focused on the debasement of masters and have often ignored the anguish of slaves, such arguments can seem obtuse and elitist. But when applied to computer users these arguments gain new importance: computers feel no pain and are not harmed by domination, but their users can still be debased for the same reasons slaveowners are.<sup>2</sup>

---

<sup>1</sup> I refer to computers as they currently operate. The achievement of true artificial intelligence would of course change matters: I shall discuss this further below.

<sup>2</sup> One reader for this journal makes the plausible objection that the kind of debasement "proper to slave holding as such" is *necessarily* bound up with its moral wrongness. He or she draws a parallel to rape, which is debasing to rapists "primarily because of the way it treats other human beings. Whatever might be debasing about having sex with an inflatable doll can't be the same as what makes rape debasing, nor would it make sense to even talk about 'raping' an inflatable doll." To this I have three responses. (1) Sex with the doll, while certainly not the same as rape, has something in common with rape to the extent that both acts depart from a certain ideal of a sexual relationship as involving love, or at least respect for another human being. Certainly rape is not bad primarily because it departs from this ideal (this departure would not even count in the top twenty reasons why

Compared to the case of human slavery, the question of computer slavery inspires not only different criticisms, but also different, and in some ways opposite, solutions. For one thing, computers will never emancipate themselves—the classic philosophic and political solution to slavery ever since Hegel. Any change must come from computer users' reflection on their own condition. This is a tall order, since it would require the unmodern admission that certain practices can be ignoble or debasing even though they are pleasurable. Beyond this, the problems of computer usage require changing the relation between humans and computers in ways that are different, in fact opposite, from the changes that end human slavery. Human slavery is the treatment of human beings as instruments for another's will, and slavery is overcome when the instruments are recognized as human, self-governing, and the equals of their owners. Computer slavery, on the contrary, is the corruption of a perfectly good instrument by the illusion that computers are quasi-human. To think that the human-computer relationship is ennobled by expanding the capabilities of computers is exactly wrong: what ennobles it is the recognition that a computer *is* a dumb tool, along with the discovery of old and new relationships between humans and their tools.

Finally, I should explain one thing I shall not do: take up the attack of Heidegger and his followers (in both postmodernism and critical theory) on “instrumental reasoning” and “technological” thought. For this line of reasoning is misleading for current purposes: it induces us to misunderstand what people who program computers actually do. Computer programming, properly done, can have the characteristics of a craft, with all the ancient and positive connotations thereof, and it can embody “work” in Hannah Arendt's positive sense—it can create an enduring human environment within which

---

it is bad!), and certainly sex with dolls should be legally allowed as rape is not. But as a point of social critique rather than law or morality, the parallel is still interesting, as is the text analogy. (2) There is surely *something* debasing about sex with the doll, and even those who practice such sex reflect the general sentiment of debasement by all-but-universally hiding their enjoyment of such activity. (Contrast pornography viewing, to which some libertine types happily admit.) Returning to the text example, few adults admit that their great aspiration in life is to win video games, and those who do admit it are considered vaguely pathetic. (3) There is a sense in which one *could* talk about a man raping a doll if the psychic state of the man involved were at issue. (Consider a man who had sex with a doll while striking it, calling it misogynist names, and generally counterfeiting a rape.) If hundreds of millions of men had rape-fantasies with dolls as their preferred sexual activity, there would be cause for social concern if not (again) for legal prohibition.

we can do other things (Arendt 1958, pg. 136-174). Perhaps not Marcuse but Camus is our most reliable guide here: the child of actual workers rather than the friend of the notional proletariat, Camus claimed that contempt for “use” and economic production is a sign of the final corruption, not the road to liberation.

### **I. Action without work, achievement in ignorance: Hegel.**

The master-slave dialectic in Hegel’s *Phenomenology of Spirit* is justly famous. Here I shall not attempt a new interpretation, but shall focus on some implications of the concept of work, which is particularly relevant to the question of how we should relate to computers.

Ruling over slaves involves telling them to perform tasks, but it is the slaves who perform them. If the task involves dexterity or craft-knowledge, it is the slave who acquires these. Aristotle was the first to point out that masters grossly inflate the skill involved in getting slaves to do things: “this science has nothing great or dignified about it: the master must know how to command the things that the slave must know how to do.” Politics and philosophy are noble in themselves; ordering people around is something we should leave to others if we can (Aristotle 1984, § 1.7, 1255b32-35).

Aristotle focuses our attention on the difference between doing something and watching others do it, a profound point but still somewhat elementary. It was Hegel who expanded the scope and importance of this observation. For Hegel, the issue involves not merely skill but *understanding how the world works*. Through work, the slave becomes not only good at the work but conscious of the human condition. If something can be learned about the world from the natural barriers confronting a given task, it is the slave who learns it, and what the slave learns is that the full development of human beings lies neither in accepting the world as it is, nor in stubbornly asserting our will against it, but in acting consciously to transform it.<sup>3</sup> As Kojève puts

---

<sup>3</sup> “[I]n fashioning the thing, the bondsman’s own negativity, his being-for-self, becomes an object for him only through his setting at nought the existing *shape* confronting him. Now...he destroys this alien negative moment, posits *himself* as a negative in the permanent order of things, and thereby becomes *for himself*, someone existing on his own account....he becomes aware that being-for-self belongs to him....the bondsman realizes that it is precisely in his work wherein he seemed to have only an alienated existence that he acquires a mind of his own.” G.W.F. Hegel, *Phenomenology of Spirit*, trans. A.V. Miller (Oxford: Oxford Univ. Press, 1977), 118-119. Cf. Jean Hyppolite, *Genesis and Structure of Hegel’s Phenomenology of Spirit*, trans. Samuel Cherniak and

it, “What changes as a result of Work is not only the natural World; it is also—and even especially—Man himself. It is only by rising above the given conditions through negation brought about in and by *Work* that Man remains in contact with the concrete, which varies with space and time. That is why he changes himself by transforming the world” (Kojève 1980, pg. 51-2).

This mention of the concrete, of what varies unpredictably, points to a problem with computer use. Hegel describes the lord’s consciousness as warped and backwards because the lord must never encounter nature and transform it himself. In Marcuse’s summary, “The laborer he controls delivers to him the object he wants in an advanced form, ready to be enjoyed. The laborer thus preserves the lord from having to encounter the ‘negative side’ of things, that on which they become fetters on man.” This is why the lord is ultimately dependent on the slave: everything on which he bases his satisfaction has already been worked on by another; he does not know how to transform nature and its obstacles directly. (Marcuse 1941, pg. 117).

Computer algorithms, to the extent that we do not program and understand them ourselves, create just such dependence. The programmer, we might say, has given us a “slave” capable of transforming nature (raw, concrete data). One might call this “frozen subjectivity”: the program is a frozen piece of the programmer’s will and intelligence, capable of performing, over and over, a task the programmer has set for it.<sup>4</sup>

The original programmer had a sense of what the task was and why a given set of instructions could accomplish it more efficiently than a human being could. He or she is in the position of Eli Whitney, who knew that a cotton gin was needed, and how to build one, because he knew from

---

John Heckman (Evanston, Ill.: Northwestern University Press, 1974), 176; Herbert Marcuse, *Reason and Revolution*, 2d ed. (London: Routledge & Kegan Paul Ltd., 1941), 116-117.

<sup>4</sup> For a persuasive demonstration of how several alleged exemplars of “artificial intelligence” essentially replicate the knowledge of their programmers, see Douglas Hofstadter and the Fluid Analogies Research Group, *Fluid Concepts and Creative Analogies* (Basic Books, 1995), 155-168, 310-318, 368-371, 408, 467-491. Hofstadter, an active and prolific researcher on the project of writing computer programs that display truly original, intelligent, and creative thought, concludes from experience with his pleasing but highly constrained models of such thought that we are far from even the most limited achievements in modeling truly human-level intelligence.

experience how hard it was to separate cottonseed from cotton by hand.<sup>5</sup> In running a computer program, we are letting a piece of the programmer's past mind do things for us that we would not know how to do.

Consider the sorting of a list of numbers or names. In the first instance, this is a physical task involving (minimal) intelligence: a clerk must take a bunch of cards, or a pile of papers, and put them in order. Early programmers kept this physical task in mind and put it in their programming books (see Figures below [next page]). The algorithms they derived stem from the clerk's commonsense ideas on how to make the task easier. For instance, if one has a large number of items it is easier first to separate them into two piles, say "A through L" and "M through Z," and then alphabetize each pile separately. Through such partitioning we can improve greatly on a common-sense but very inefficient method: digging all the way through a large pile to find the item that should be first in the list, then digging again for the one that should be second, and so on.<sup>6</sup> Repeated partitioning greatly reduces the size of the pile to be dug through. If I understand it correctly, a complex version of this technique lies behind the fastest known sorting algorithm, known as "Quicksort." The computer makes each "pile" smaller and smaller until it consists of only two items.<sup>7</sup>

---

<sup>5</sup> For a fine account, see John Dewey, *The School and Society*, revised ed., in *The Child and the Curriculum and The School and Society* (Chicago: Univ. of Chicago Press, 1943), Chapter 1, 19-22.

<sup>6</sup> The crude technique is called "straight exchange." Other techniques, called "straight insertion" and "straight selection," correspond to more plausible but still slow techniques, such as going through the pile and putting each item in its proper place among those items sorted so far. Nicklaus Wirth, *Algorithms + Data Structures = Programs* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1976), 56-68.

<sup>7</sup> This is an interpretation of the account in Wirth, 76-82.



**Fig. 2.1** Array sorting.



**Fig. 2.2** File sorting.

Source: Nicklaus Wirth, *Algorithms + Data Structures = Programs*(Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1976), 57, 58.

The programmer who understands Quicksort possesses all the understanding of the clerk. She then improves on the clerk by designing a procedure that can be performed easily by computers but not by humans. In contrast, the user who chooses “sort addresses” from the menu of an email program knows less than the clerk. She has no knowledge about sorting or its difficulties. She may even think that the problem is easy, or assume that the computer is using a commonsense method (like crude digging) that would in fact be very slow if it were used. The situation is like that of the child who eats a ham sandwich but does not know that ham is produced by the raising of pigs. Using computers makes us ignorant and keeps us ignorant.

The programmer also must confront strange, random, and unexpected facts about the world—the “concrete” qualities of which Kojève wrote. For instance, Quicksort requires that one pick an item at random as the one around which others will be partitioned. But there is literally *no way to know* which item would be best, unless we go through the whole pile (which destroys the whole point). If one makes the wrong choice, Quicksort becomes as slow as the slowest known sorting algorithms. (To go back to the commonsense example: one can choose piles of A through L and M through Z. But if all the names *happen to start with M through Z*, one will have sorted the items into “two” piles only to end up with the original pile, thus accomplishing nothing.) Computers cannot solve this problem: it essentially stems from the fact that one cannot know what a random pile contains without going through the pile. The speed of the most efficient algorithm therefore depends on sheer luck—although, as with most random events, the worst case is unlikely to occur any given time.

This is a small example of a larger truth. The arrangement of information, like the acquisition of any art, science, or craft, has its own rules and involves its own discipline.<sup>8</sup> This fact is known to librarians, programmers, and file clerks. But it can be utterly lost to computer users, who may think that information comes without effort and at no cost.

## **II. Aristotle: the dilemma of language and the psychological roots of “artificial intelligence.”**

One solution to the above dilemma is to strive for the kind of computers who can *talk to us about what they are doing*: computers that are self-aware, articulate, and understand the kind of things about which human

---

<sup>8</sup> See Hegel, *Phenomenology*, 119; and on the link between discipline and experience, Dewey, 17.



beings are curious.<sup>9</sup> If computers could do this, working with them would involve no risk of increasing our ignorance: computers would be like colleagues, capable of giving and receiving questions about the work we were doing in common with them. Of course, for a computer to do this it would have to have considerably more abilities than computers do now: it would need artificial intelligence. The quest for artificial intelligence is therefore natural or “immanent”: it stems from the desire to overcome the dependent inequality of the human-computer relationship and the harms we can see as flowing from it.

The desire for computer intelligence gives rise to a paradox: we want the ease and efficiency of having a machine that will do what it is told, but also the engagement and interaction of human companionship. Unfortunately, achieving the latter would undermine the former: a computer intelligent enough to speak and reason with us would be too intelligent to follow orders and perform artificial mind-numbing tasks without question.

Aristotle’s treatment of slavery, it has been argued, contains exactly the same paradox; both the human case and the computer case will turn out to depend on the question of language. I shall claim that a look at the human case shows us what is absurd about the computer case: the effort to make computers true conversational partners is inconsistent with using computers for anything else.

The most logical justification for slavery is to portray the natural slave as severely deficient and incapable of independent existence. Thus Aristotle describes natural slaves as being as inferior to their masters as the body is to the soul (*Politics* 1254b5-9<sup>10</sup>); the use of the body is the best of which slaves are capable (12154b15-19). Slaves wholly lack the deliberative element of the soul: they perceive reason but have none (1260a11-12). They resemble inanimate tools, or extensions of the master’s body (1253b20-34; 1254a12-13); their closest analogues are animals, which provide “bodily assistance in the necessary things” (1254b25; cf. 1260a33-35). Because slaves lack full virtue, the master-slave relationship cannot involve friendship in the true sense. Consistent with this, seeking friendship for utility alone is characteristic of the unfree [*aneleutherios*], and it is slavish to let anyone except a true friend determine one’s life (*Nicomachean Ethics* 1158a20-23, 1125a1-3). This is why the master-slave relationship involves “advantage

---

<sup>9</sup> Hofstadter, 121-123 and 311-318, considers this a crucial hallmark of creative and intelligent thought; his model programs are self-aware in tiny though significant ways.

<sup>10</sup> All citations of Aristotle’s *Politics* refer to the translation by Carnes Lord cited above.

and affection” for both—but nothing more. A dog is “man’s best friend” in a figurative sense, and having a good master is the best thing for a dog, but to regard a pet as one’s truly best friend, similar to human friends in all important respects, would be bizarre. The same goes for computers.

As Nicholas Smith points out, however, Aristotle’s account has a different side that seems quite inconsistent with this relentlessly harsh portrayal of slaves. Aristotle describes barbarians as natural slaves, which implies a lack of forethought, but later describes them as having ample foresight and deliberation and lacking only a certain spirit (Smith 1983, pg. 110-111). Moreover, after Aristotle justifies natural slavery he then advocates using emancipation as a reward—which, if Aristotle is right, would seem cruel to master and slave alike.<sup>11</sup>

The central paradox in Aristotle’s account appears in the following awkward passage:

Mastery, in spite of the same thing being in truth advantageous both to the slave by nature and to the master by nature, is still rule with a view to the advantage of the master primarily, and with a view to that of the slave accidentally (for mastery cannot be preserved if the slave is destroyed) (Aristotle 1984, 1278b32-36).

One could read this in many ways. Smith takes it to mean that “the morality of slavery erodes in Aristotle’s own thought as rapidly as the alleged benefits to the slave” (Smith 111). That is, by explaining good treatment of slaves in purely instrumental terms, Aristotle shows that he does not really believe what he says about mutual advantage. More charitably, one could suggest a pessimistic or critical reading: perhaps Aristotle thinks that masters ideally *should* have a benevolent regard for their slaves, but in (Athenian) practice tend to use slaves as mere instruments, and to preserve them only for the same reason one tries not to break a good tool. But this seems inadequate, because it does not explain why Aristotle would use the reasoned distinctions “primarily” and “secondarily.” If he were merely talking about common

---

<sup>11</sup> Smith 111, citing Aristotle *Politics* 1330a32-3. Smith also notes the fact that Aristotle emancipated his slaves in his will. This, however, need not involve a difficulty, since Aristotle is the first to admit that not all existing slaves are natural slaves. Perhaps Aristotle merely discovered close to his death that his own slaves were slaves only by convention not by nature—proving Aristotle to have bad judgment, but not necessarily an inconsistent theory.

usage, he could merely say that the opinion of the many was not quite correct—as he often and quite freely does in other places.

So there is a real tension here, one which is summed up in Aristotle's description of a slave as "tool with a soul." The slave is both a tool—something for the master to use—and a being with a soul—therefore *not* for the master's use. This seems to be Aristotle's meaning: the master has nothing in common with the slave "in so far as he is a slave," "but there is friendship with him in so far as he is a human being" (Aristotle 1985, 1161a32ff). This is not quite coherent. It is not clear how a master could treat a slave as a friend if the slave indeed lacked reason (this seems to be Aristotle's position in other passages [Smith 113]), nor how the master could avoid treating the slave as a friend if the slave were capable of friendship (and therefore has full human virtues and capabilities).

Smith suggests an apparent way out. The fact that a slave can receive reason from his or her master but cannot exercise reason suggests that the relationship is one of *teaching* (Smith 113). Slaves, after all, start with some human capabilities: they have souls and an emotional life.<sup>12</sup> Over time, they can pick up better and better moral habits—always under the master's direction—and can eventually become capable of freedom. In particular, a good master should be able to teach slaves the use of *language*. Aristotle distinguishes human beings from animals on the grounds that the latter lack reasoned language (*logos*); the same apparently goes for barbarians. (As Smith points out, this explains why Aristotle in 1256b23-26 compares hunting war captives to hunting animals [Smith 119].) Barbarians may be able to speak and communicate, but they cannot use *proper* language, the kind of language that makes one capable of reason (i.e., Greek). But masters can provide their slaves with a great deal of training, apparently including Greek.<sup>13</sup> Thus, we might distinguish between two states of slavery: *before* the

---

<sup>12</sup> On the role of emotional capabilities as the missing element in discussions of slavery in Aristotle, see W.W. Fortenbaugh, "Aristotle on Slaves and Women," in Jonathan Barnes, Malcolm Schofield, and Richard Sorabji, eds., *Articles on Aristotle*, Vol. 2 (London: Gerald Duckworth & Company Limited, 1977), 135-139. Smith's article contains an extensive consideration and critique of Fortenbaugh's position.

<sup>13</sup> Masters apparently use this language for giving instructions, and from all accounts slaves used it rather well, if we believe Plato's *Meno*. We might note, however, that Socrates asks, of Meno's slave, "Is he a Greek? Does he speak Greek?" Meno replies, "Very much so. He was born in my household" (Plato, *Meno*, trans. G.M.A. Grube, in *Plato: Complete Works*, ed. John M. Cooper [Indianapolis, IN: Hackett Publishing Company, 1997], 82bff). This may imply that the average "barbarian" slave taken in war

slave has a master, he or she lacks *logos* and is like an animal; but as soon as the master takes charge, the slave becomes reasonable, able at least to understand language if not to learn it (at least not right away). Such a slave is a fit, if flawed, companion for a free master and perhaps a candidate for eventual manumission (Smith 120).

But this, Smith points out, creates a paradox. Once the slave is enslaved, and gains “the benefits of his master’s reason,” the justification for despotic rule does not hold. An animal with no potential for reason may legitimately be “enslaved” for pack or draught, but a person who can speak and understand should be taught for his or her own benefit, not ordered around despotically to satisfy the master’s whims. (In Aristotelian terms, the type of rule involved should be monarchical—just as Aristotle says parental rule over children is—rather than despotic.) The argument from lack of language, Smith concludes, justifies making people slaves, but not continuing to rule them as slaves. The slave could only be justly kept enslaved if the master *failed* to give him or her any share in reason—but that would undermine the rationale for enslavement in the first place (Smith 120-121).

The analogy with computers is very close, but the proper conclusions to be drawn are different. Computers do not engage in human language and reasoning. In fact, a computer that could converse like a human for an extended time would have passed the Turing test, a widely accepted (though controversial) test of artificial intelligence. Existing computers are far from that. Left to their own, they would, like Aristotle’s natural slaves, have no language at all. Programmers have given computers very limited languages—many times simpler than the crudest human pidgin—in order to

---

was *not* taught Greek and did not speak it. In speaking to their masters, such slaves might indeed seem like the natural slaves Aristotle portrays: they would clearly be more human than animals and capable of some judgment in carrying out tasks, but one would not be able to reason with them or “deliberate” about distant issues. This implies a host of possible interpretations. Aristotle, like Plato’s Socrates, thought that it was better to hunt barbarians as slaves than to hunt Greeks, perhaps because he had never observed a Greek who could not reason coherently. Perhaps Aristotle thought that barbarians gained more-or-less human status *if* they learned Greek—but that barbarians could rarely, or never, do so. (This would have serious implications for masters: if barbarians are capable of learning Greek, it seems inexcusable on Aristotle’s account not to teach them. But if they are *not* capable, then perhaps there is nothing left but to order them around.) Perhaps the slaves Aristotle emancipated, and the ones he proposes emancipating as a reward, were all Greek-speaking, born in Greek households. But such possibilities take us beyond the scope of this paper.

give them instructions. And the way computers respond to such languages is certainly slavish: they carry out instructions exactly as told, whether or not they make any sense. To paraphrase a common programmer's saying, a computer is like an ox: tell it to plow straight ahead and it plows straight ahead—right to the end of the field, through the fence, and into the lake unless you stop it. More succinctly, “computers are stupid”<sup>14</sup>—*very* stupid, stupid enough to treat as slaves.

This would be fine, and to the advantage of both humans and computers, if humans were not almost as lazy and neurotic as computers are stupid. That humans are lazy means that few of us are willing to learn enough about computers to know how to program them.<sup>15</sup> We want something or someone to take the hard work away from us: hence, we buy complicated interfaces that give us the illusion of smart machines. Those who do not understand AND and OR operators can now type normal English sentences into AltaVista searches. (These searches tend to come out badly, search engines still being too stupid to really understand what we are saying.) And people who need help in Microsoft Word now have at their disposal a small animated computer with legs, which (who?) pops up on the screen to “ask” us what we want help on. Such interfaces give us the illusion that computers are our friends. But this is a dangerous illusion: to seek the company of computers rather than humans is, Aristotle would remind us, to lower oneself to the level of an animal (at best). There are no talking horses, and no smart machines.

Programmers, who are not lazy, are neurotic: that computer programming involves not human interaction but the repetition of simple instructions to a sub-moronic machine apparently creates in them deep-seated feelings of guilt. They therefore pretend that they are working towards the creation of new, artificial, intelligence. This creation may be possible; we should be skeptical but not totally dismissive. A programmer who really

---

<sup>14</sup> “Computers are stupid. Years ago, to save space, we told them that 60=1960, and they believed us!” Sandy McMurray, “Facing the Y2K Problem,” *Toronto Sun*, 22 April 1998. This is just one example: a web search for “computers are stupid” turns up hundreds of hits.

<sup>15</sup> One could have a noble Aristotelian justification for this too: programming computers is “nothing very noble,” and those who can afford an overseer (hired programmer or bought software) should spend their time instead on philosophy or politics. But few computer users spend their leisure that well. Those of us who have a “Protestant” work ethic or a “do-it-yourself” attitude towards household labor have few excuses to be ignorant of computer programming.

believed in artificial intelligence, however, would face the same paradox that Smith pointed out for slave-owners: a programmer who were *really* trying to teach human language skills to a computer could not justify simultaneously using the computer as a slave to perform mundane tasks. Education rules out despotism. The world contains English as a Second Language students, and also trained hamsters; but to treat the former like the latter involves an error of moral judgment. Programmers probably know this at some level, and yet continue to treat computers as computers, not as immature people. This probably reflects, not a lack of moral judgment, but a tacit recognition that the project of teaching computers to use language and reason is not to be taken too seriously as a near-term aspiration.

There is one final possibility. For computers—fortunately unlike humans—it might be possible to create a computer that had intelligence but no initiative or spirit. This machine would be able to reason but not to rebel or to assert its rights. It is respectable, though controversial, to claim that a being with this quality would be a natural slave.<sup>16</sup> Nor must this argument be merely opportunistic: beings who could never assert themselves, never affect their masters emotionally, are arguably, though again controversially, outside the bounds of moral obligation. Describing how he dangerously picked a fight with a slave-driver who tried to break him, Frederick Douglass wrote:

I was *nothing* before; I was a MAN now....A man, without force, is without the essential dignity of humanity. Human nature is so constituted, that it cannot *honor* a helpless man, although it can *pity* him; and even this it cannot do long, if the signs of power do not arise (Douglass 1938, pg. 247).

The position of David Hume was quite similar: if there were a “species” living among us “which, though rational, were possessed of such inferior strength, both of body and mind, that they were incapable of all resistance, and could never, upon the highest provocation, make us feel the effects of their resentment,” we would have the duty to treat them humanely—but would

---

<sup>16</sup> Aristotle (Politics 1327b23-28) describes Asiatic peoples this way: they have intelligence but lack *thumos*, spirit; that it is nonsense as applied to any human beings in the world does not make the argument uninteresting. Smith (111n4) notes this passage as part of the difficulties in explaining Aristotle’s position, but does not explore the possible principled argument behind it. *Thumos* for Aristotle is often associated with rash or angry action, especially in response to slights: see *Nicomachean Ethics* 1116b24, 1149a24-37; in *Politics* 1327b24-32 Aristotle seems to define the necessary capacities for free political life as thought plus *thumos*.

have no duties of justice towards them, and they would have no rights against us (Hume 1983, § III, 25).

All this, however, describes our hypothetical moral duties if such helpless beings happened to exist among us, or suddenly appeared from nowhere. The morality or ethics of *creating* such beings on purpose is very different. Consider what would be involved. A machine that could use language in a human way could understand arguments about interest and disadvantage, justice and injustice.<sup>17</sup> The computer would know that its capabilities gave it a just claim to be treated at least as a child (if not as an equal, since it would lack one crucial quality). It might even remind us of this fact every so often, simply as a factual observation without an implied action. Yet we would not give it the respect due a reasoning being, because we would know that the computer would not assert itself: we would have *deliberately* made it unable to assert itself. We might avoid destroying these machines—because they would be useful to us— but while we kept them alive we would treat them as slaves, making them do all manner of tasks for us even when they would rather be composing symphonies or discussing Aristotle. This would be a training in cruelty, even more corrupting of our characters than cruelty to animals, because it would consist in mistreating rational creatures rather than dumb ones. A project of creating such beings would not be a noble one, and it is no surprise that most AI researchers aim at creating beings with all human qualities, not just reason.

### III. Conclusion: computing with respect

I have suggested that these two ways of regarding computer use are mistaken. Computers do not increase our intelligence: unless we have programmed them ourselves, or at least understand the principles on which their programs are based, we do not “know” how to do the tasks we accomplish with their aid. In fact, we may know less than if we tried and failed to do those tasks ourselves. Nor can computers be our friends: they are not intelligent enough, and the illusion that we are making them intelligent involves us either in absurdity or in cruelty.

Two alternatives remain. The first is to treat the computer as a tool, neither more nor less, and to treat the use of computers as a craft. Because computers are such powerful tools, and are so hard to use, we are tempted to exaggerate the nature of computer use—to treat it a “science” or even a “new

---

<sup>17</sup> These are the human qualities mentioned in Aristotle, *Politics*, 1253a14-18.

way of thinking.” This aims too high. Instead, the use of tools has a nobility to it that should not be shunned. The possibilities were set out, with remarkable foresight, by Camus, who compared a (primitive) computer or calculating machine well-used to a truck well understood by its driver:

The machine is bad only in the way that it is now employed. Its benefits must be accepted even if its ravages are rejected. The truck, driven day and night, does not humiliate its driver, who knows it inside out and treats it with affection and efficiency. The real and inhuman excess lies in the division of labor. But by dint of this excess, a day comes when a machine capable of a hundred operations, operated by one man, creates one sole object. This man, on a different scale, will have partially rediscovered the power of creation which he possessed in the days of the artisan (Camus 1991, pg. 295).<sup>18</sup>

In this way the computer, treated as a tool by a single craftsperson who knows and appreciates it, can recapture some of the nobility of the artisan age.

Early computer pioneers did use computers this way. Bill Gates, at one time a programmer rather than a tycoon, described in 1986 a programming experience with all the characteristics of craft work. “If you

---

<sup>18</sup> . I do not share Camus’ opposition to the division of labor as such (and am not sure Camus intended to express it so radically either). The issue is not that everyone should know how to do every task from which he or she derives benefit, but that we should try to avoid the illusion of mastering tasks that we do not understand. To the extent that we depend on the products or services of other human beings, we should know that (and how) we do so, because only this knowledge lets us appreciate their contributions as creative, intelligent persons and ones whose difficult work resembles our own. Computers give us the illusion of mastering tasks that in fact embody the product and service skills (software partakes of both) of others. Students can run a “grammar checker” without understanding the rules of English grammar; online encyclopedias and other data sources hire educated people to write “content,” but the intelligent work that is done is obscured when a user’s “research” consists of turning up data indiscriminately and pasting it into a new, portmanteau product. Similar considerations differentiate computers from other tools: someone using a hammer knows that he or she provides the intelligent choice of where to drive the nail, while the hammer embodies the hardness and symmetry needed to make the blow effective. A computer is like a hammer that makes people think their arms are made of steel.



ever talk to a great programmer,” said Gates , “you’ll find he knows his tools like an artist knows his paintbrushes.” Gates’ other observations suggest that he knew what a craft involves. First, there was a master-apprentice relationship: Gates still remembers the memoranda a friendly TRW programmer gave him on his code. Second, good programming meant devotion to the artful result—a “super well crafted” program—rather than the money made or even the number of users who ran the program. Third, programming involved pride in the work, “a refined sense of discipline about what’s sloppy and what’s not sloppy.”<sup>19</sup> Like Camus (and Arendt<sup>20</sup>), Gates thought the ideal craft situation was solitary work: “then you control everything. There’s no compromise. Every line is yours and you feel good about every line” (Gates 1986, pg. 90). Finally, one could add (though Gates does not mention this), the programming craft, done well, can even create the enduring environment for others that Arendt regarded as work’s highest achievement: programmers are taught that one test of craftsmanship is the ability of others to read one’s code. One reason the millennium bug was such a potential problem was that system code from the 1960s was—contrary to all expectations—still in use.

Some critics of technology completely misunderstand what is involved here. For example, Marcuse described the “electronic industries” as involving applied formal logic: he apparently thought that computing meant a few logical connectors like AND and OR, plus the systematic study of what happens when they are combined (Marcuse 1964, pg. 169). This reductionist view of programming is ignorant and half-baked: no one could write a real program by thinking on that level. Marcuse’s portrayal is analogous to the view that all there is to know about racecar driving is contained in “wheel left,” “wheel right,” “accelerate,” “shift gears,” and “brake.” That is true on one level but not on the level that matters, which involves judgment of what action to take in a particular case so as to cover the most distance in the least time given the actions of other racers and the demands of safety. Computing, like many crafts based on simple operations, allows full scope for human judgment and creativity.

---

<sup>19</sup> Bill Gates (interview), in *Programmers at Work*, first series (Redmond, Wash.: Microsoft Press, 1986), 83, 76, 83. The book is not just “house” propaganda: the interviews avoid hard questions, but the responses seem honest and Gates’ attitudes are reflected in others’ statements. Nor does the book seem written for a popular audience or for public relations purposes.

<sup>20</sup> Arendt, *Human Condition*, 161.

This is not to deny that craft has its limits. It cannot substitute for politics, and skill at a craft (Aristotle's *technê*) is not the same as ethical and political judgment (*phronêsis*).<sup>21</sup> Moreover, as Socrates noted, craftspeople's legitimate pride tempts them into thinking that the craftsman's knowledge is the basis of all important knowledge.<sup>22</sup> But all this is somewhat moot, since craft-knowledge of the kind Gates described is no longer really possible under the conditions programmers face. The developments that would bring this about were predicted by Gates (with an odd lack of concern) in 1986. First, programs are now much too big for one coder: it is no longer possible to know all of a fundamental program by heart and take pride in its perfection, as Gates and two colleagues knew the original BASIC interpreter (Gates 72-3). Second (and a bit paradoxically), technology has decreased the incentive for good crafting: BASIC originally fit in 4 kilobytes because that was all the memory available, but similar discipline becomes rarer as it is less necessary.<sup>23</sup> Third, the craft production has followed the route of many physical crafts in becoming a matter of mass production: "routine" code is now written by expert systems,<sup>24</sup> or else by ill-paid programmers living far apart from those who commission them. Fourth, the very success of computers and the focus (still baffling to old-style programmers) on "ease of use" has ensured that the average user has neither skill in computing craft nor clear knowledge of what it entails: just as the average mechanically gifted driver can no longer do fundamental repairs on a car, the average computer user cannot reprogram his or her own system, and could not even if the code were generally available.<sup>25</sup> Thus even where craft is preserved in the work of some programmers, it remains invisible to users. Finally, though Gates does not mention this,

---

<sup>21</sup> Aristotle's distinction between *technê* and *phronêsis* was stressed by Heidegger and taken up by many critical theorists and others. For a good short treatment, see Hans-Georg Gadamer, "The Problem of Historical Consciousness," in Paul Rabinow and William M. Sullivan, eds., *Interpretive Social Science* (Berkeley: University of California Press, 1979), 138-145.

<sup>22</sup> Plato, *Apology of Socrates* 22d-e. Gates at one point equates the ability to understand computer code quickly with "pure IQ" (83).

<sup>23</sup> Gates, 73. Gates complained about both these factors in 1986; in 1999 the situation is of course even worse.

<sup>24</sup> A prediction made by Gates, 85.

<sup>25</sup> Gates points out that early BASIC included PEEK and POKE commands which essentially let the user do machine-language programming within the high-level language (79). The author's own memory recalls that this engineer-like tinkering—checking what number was in a particular place in computer memory, or replacing it with another number—was not just a luxury but a necessity for early Apple II programming.

money and fame now seem to be more of an incentive to him, and to others who formerly did programming for its own sake, than they perhaps once were.

The craft ethic still survives in some areas of computer work, and will always survive in cutting-edge work, but the average user has little access to it. It can only now be salvaged through education, and provides a reason for studying computing history. Just as Dewey taught students why the cotton gin was so important by making them try removing cottonseeds by hand, the history of computer programming can help us truly understand what computers can and cannot do by giving us some idea how people made them *able* to do it.

Over time, however, we are likely to go in the opposite direction. If computers from a programmer's perspective are stupid and ugly, from the user's perspective they are likely to become ever smaller, more attractive, and easier to use. At some point they are likely to become "transparent," not in the intellectual sense of being easier to grasp, but in the more literal sense of being easier to look *through*. Computers could become like telephones: something we communicate through automatically, rather than something whose technological importance we think about. Even now, calling something up on a screen is almost as familiar as calling someone up on the phone. We now talk about living in the "computer age" or "information age," but the concept may one day sound as strange as "telephone age," "photocopier age," or "pencil age."

As computers become ubiquitous, our relation to them may become less emotionally serious. Fear, arrogance, smugness, and nostalgia may yield to an unconscious practicality, as anxiety becomes insouciance. Computers may become neither masters nor slaves. One will still use computers, for computers will be a form of entertainment. But one will be careful lest the entertainment be too harrowing. Who will still want to rule? Who obey? Both require too much exertion.

This is the language of Nietzsche's "last man," but non-Nietzscheans among us might welcome such a normalization of the human-computer relationship. Computers cannot provide us with the joys (and dangers) of political rule, social recognition, or economic competition and cooperation. At their best and noblest, they are modest tools that can help us with the things that really matter, and the sooner we admit this, the better.

**Reference**

Arendt, Hannah. *The Human Condition* Chicago: University of Chicago Press, 1958.

Aristotle, *Nicomachean Ethics*, trans. Terence Irwin. Indianapolis, Ind.: Hackett Publishing Company, 1985.

Aristotle, *Politics*, trans. Carnes Lord. Chicago: Univ. of Chicago Press, 1984.

Camus, Albert. *The Rebel*, trans. Anthony Bower. New York: Vintage, 1991.

Douglass, Frederick. *My Bondage and My Freedom*, New York: Arno Press and The New York Times, 1968.

Gates, Bill. (interview), in *Programmers at Work*, first series Redmond, Wash.: Microsoft Press, 1986.

Hume, David. *Enquiry Concerning the Principles of Morals*, Indianapolis: Hackett Publishing Co., 1983.

Kojève, Alexandre. *Introduction to the Reading of Hegel*, trans. James H. Nichols, Jr., ed. Allan Bloom. Ithaca, N.Y.: Cornell Univ. Press, 1980.

Marcuse, Herbert. *One-Dimensional Man* Boston: Beacon Press, 1964.

Marcuse, Herbert. *Reason and Revolution*, 2d ed. London: Routledge & Kegan Paul Ltd., 1941.

Shelley, Mary. *Frankenstein, or, The Modern Prometheus*. with an afterword by Harold Bloom New York and Toronto: New American Library Signet Classics, 1965), p. 160.

Smith, Nicholas D. "Aristotle's Theory of Natural Slavery," *Phoenix* 37, No. 2, 1983.

Williams, Bernard. "Necessary Identities," in *Shame and Necessity* Berkeley: Univ. of California Press, 1993.