

Computing in the Image of God

Keith Vander Linden
Calvin College

October 1, 2002

As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

— Frederick P. Brooks, Jr. *The Mythical Man-Month*, 1975

1 Introduction

Fred Brooks (1975) wrote of the curious delight he enjoyed when designing and implementing computer systems. One can wonder at this, given that Brooks had just completed his leadership of the development of the IBM 360 operating system, a large, messy, and, no doubt, frustrating process. Where's the delight in all of that, one might ask, and why is a Christian such as Brooks working in the field of computing at all when his abilities might better serve God's kingdom elsewhere?

This document addresses these questions by appealing to God's creativity and to our role as God's image bearers in the vocation of computing. It also discusses the distortions inherent within the current state of computing, and the challenge that this presents to Christian computing educators.

2 Our Creative Instinct

Humans derive an innate pleasure from building things. This part of our nature is a result of our being created in the image of a creative God. Genesis 1 records that after the creation of the world, "God saw all that he had made, and it was very good." Given that God was pleased with the product of his labor, and also, presumably, with the process of creating it, it is not surprising that we, who are created in the image of God, would also find delight in designing and building things.

One area of study in which we design and build things is the field of computing, a mathematical science concerned with the design and implementation of data structures and algorithms. In this field, we find a delight in building efficient and useful structures and algorithms, as Brooks notes in his book *The Mythical Man-Month* (1975). He draws pleasure from building computing systems that are useful to people and from using a tool as malleable and complex as the digital, electronic computer. He, of all people, is not blind to the woes of computing. He lists them: the picky, boring details; the lack of control of the environment; the frustration of dealing with buggy software tools;

and the short shelf-life of most of what we build. He finds, nevertheless, that the delight outweighs the woe and he credits this to the spark of divine creativity within him.

Not all pleasurable things are worth doing, however, so we must also ask what Godly purpose there is in computing. Without a Godly purpose, computing could not be seen as a vocation. Here, we can see two basic motivations: computing helps us pursue God's command to exercise dominion over the earth, and it helps us appreciate the wonder of God's creation.

God commanded Adam and Eve to "subdue" the earth (Gen. 1:28), which I take to mean control or bring order to the earth, and to "work it and take care of it" (Gen. 2:15). Computing systems are one useful tool for doing these things. They support a myriad of forms of communication and transportation systems, as well as information and control systems. As we will discuss below, they can be abused, but when designed and used properly, they enhance or extend God's creation. In addition, a truly elegant computational solution to a problem has a beauty that is not unlike the beauty of God's physical creation. As such, it can serve as a fitting reflection of the Creator of all things. Computing systems can also take care of details that we are not well-suited for (e.g., mass data storage and retrieval), allowing us to focus on things that are more important (e.g., helping people, creating useful tools).

Computing can also help us appreciate the wonder of God's creation. Though it is not a "natural" science in that it doesn't focus on observing and modeling the natural systems that God created, computing does confront natural systems in at least two ways: in the laws that govern computing and computability, and in the natural phenomena that we frequently model with computers.

There are laws that govern the scope and nature of computing that God has put into place for us to discover and take advantage of in our creative endeavors. At the very root of these laws is the consistent behavior of natural materials, without which there would be no point in trying to build anything at all. We bear witness to God's creative genius each time we build and use a computer. The same is true in software. The nature of software is itself governed by laws of computability, which dictate capabilities of and limits on the theoretical power of computation. These meticulously fashioned systems of law bear witness to God's power and creativity.

Though it is not the core of computing, we do, on occasion, model natural phenomena. In this endeavor, we enjoy the same delights common to the natural sciences; we observe a phenomenon and are given a glimpse of the creative genius of God. Artificial Intelligence (AI) is one such area. Alan Perlis once quipped "A year spent in artificial intelligence is enough to make one believe in God." (Perlis, 1982). He may have been speaking in jest, and indeed most people take it as such, but the sentiment is relevant here. In AI, we come face to face with the fearful and wonderful reality of the human mind. The harder we try to model it in computer systems, the more awe we feel in the face of God's creation. The psalmist says:

*For you created my inmost being;
you knit me together in my mother's womb.
I praise you because I am fearfully and wonderfully made;
your works are wonderful, I know that full well.*
Psalm 139:13-14

3 The Current State of Computing

The delight of biblically justified creativity discussed in the previous section springs from the wells of who we are, how we were made, and what our ultimate purpose is. Unfortunately, this ideal is hard to achieve in practice given the number of distortions that have arisen. The existence of these distortions is not surprising; computing professionals are, after all, fallen people working in a fallen world. Christian computing professionals are, however, strategically placed to promote the ideal and to fight against its distortions. This section will review a number of potential distortions and will discuss appropriate responses to them.

Making gods of ourselves. The first, and most egregious of the distortions is the use of computing, or technology in general, to place ourselves above God. Genesis 11 records that the people of the world came together and said “let us build ourselves a city, with a tower that reaches to the heavens, so that we may make a name for ourselves and not be scattered over the face of the whole earth.” There were, undoubtedly, some who drew satisfaction from the process of designing the city and the tower, though the probable use of slave labor must have squelched most of the joy out of the implementation process. The driving force behind this so-called Tower of Babel, unfortunately, was clearly not honoring to God and ultimately ended in ruin.

I know of no projects that are so blatantly arrogant in modern computing, but there are examples of malicious programmers taking advantage of computing systems to work their own will. People who write and distribute viruses are one example. People who exploit the addictions of others, for example through internet pornography or gambling, are another. Our response to these distortions, as educators, is to point them out to our students, and to redouble our efforts at training in the prevention and detection of security breaches. This will help to prevent attacks, minimize their effect, and reduce access to addictive materials.

Another example of this distortion is people who demote God, either explicitly or implicitly, by elevating humanity or the technology it creates. For example, a common assumption in Artificial Intelligence (AI) is that God either doesn’t exist, or isn’t all that relevant. Some futurists, such as Ray Kurzweil, claim that artificially intelligent systems will become more intelligent than humans (Kurzweil, 1999). While parts of these arguments are both practical and justified, careful thought has to be invested in determining the true limits of technology. This would allow us, as educators, to train students to set appropriate goals and to more accurately assess their own accomplishments.

Making a god of computing. Another distortion is that of making computing itself into a god. Some people carelessly use computing technology for its own sake, without regard to the context or the ramifications. For example, the use of Machine Translation (MT) for politically or socially sensitive documents is potentially dangerous. MT technology has simply not advanced enough to be used to mediate general communication, say, between parents and teachers, or between military personnel and civilians.

In each situation, we must train students to carefully decide how computing technology will be employed, or, indeed, if a technological solution is appropriate at all. In those cases where it is warranted, the student must determine the appropriate limits of its use. Making these determinations requires that we train students to know not only what the technology is capable of doing, but also what it is not capable of doing. Much recent work has followed this path by producing tools that are intended to aid humans rather than to replace them. MT tools are good example of this

because they are now commonly seen as aids for translators and their output is commonly treated as draft output.

Another example of the careless use of technology is poorly designed user interfaces. Here, the programmer builds a user interface for a system without regard to its potential users, which can lead to unusable and potentially dehumanizing interfaces. Ben Schneiderman cites a study claiming that 45% of users' time is spent dealing with computer-related problems (Schneiderman, 2002). It is no longer acceptable to build systems that can be made to work, the systems must also be usable and learnable in context.

Producing more useful and usable systems requires attention not only to the requisite data structures and algorithms, but also to the nature of the application domain and the potential users. This requires that students have broader skills in the social and managerial sciences than most computing professionals have. I confess to enjoying this part of computing as much or more than the actual programming. Dilbert may ridicule these concerns by relegating work on them to a fictional "cross-disciplinary SWAT team", but they are as critical to computing as the design of good data structures and algorithms, and can be an equal source of delight. The computing field is slowly awakening to this issue, and computing educators must push themselves and their students to strive not just to get the system to run, but to get it "right" in context.

A final manifestation of this distortion is the tendency of computing professionals to focus on the limited goal of developing and maintaining technical competence in the field. This is not surprising. Technical competence is a highly marketable skill, regardless of whether it is coupled with any delight or Godly justification. Unfortunately, this limited goal will eventually sap a computing professional's energy and enthusiasm because it limits him or her to the unrewarding drudgery of keeping up to date in a fast-paced field. Our best response to this, as educators, is to encourage students to see beyond the process of computing to the Godly delight and purpose of the vocation.

Making a god of technical achievement. Another distortion is that of making achievement into a god. This distortion is, in some sense, better than the one discussed in the previous section in that the computing professional at least sees beyond the act of programming itself. The writer of Ecclesiastes records the achievement of "great projects", including the design and construction of houses, vineyards, parks, gardens and reservoirs. Ultimately, however, he concluded:

*My heart took delight in all my work,
and this was the reward for all my labor.
Yet when I surveyed all that my hands had done
and what I had toiled to achieve,
everything was meaningless, a chasing after the wind;
nothing was gained under the sun.*

Ecclesiastes 2:10-11

The designer experienced the satisfaction of achievement here, but it ultimately proved unsatisfying because neither the products nor the achievement of building them could ever replace God. Focusing our computing curriculum on achievement alone, therefore, sells this Godly vocation short. As with the previous distortion, our best response to this, as educators, is to encourage students to see beyond the achievements of computing to the Godly delight and purpose of the vocation.

Making a god of money. A final distortion is making a god of the money that can be earned in computing. It is quite common for students to come to this field because of the salaries that they expect to earn upon graduation, and there is some financial satisfaction in store for the students who have the appropriate skills. Matthew 6, however, points out how little delight money delivers in the long term. This motivation can also lead to poor software because it encourages professionals to focus on marketability rather than quality.

Being clear in the educational process about the underlying motivations for computing goes some way here to combating this distortion. We will, almost certainly, be required to encourage some students to reassess their motivations for going into computing. There is certainly nothing wrong with making a good living in computing, but we should never let the students, or ourselves for that matter, forget the spiritual motivations for the work. We will also have to prepare students for the unrelenting push in industry to reduce the time-to-market for computing solutions. Our students may, one day, be called on to stand before their technical and financial managers and say that a product is simply not ready to ship. This will take a measure of courage and may cost both the student and their company money, but it may be the most Godly thing to do.

4 Conclusion

This document has reflected on the Godly calling to work in the field of computing and the delight that can be drawn from this. It also discussed a number of potential distortions of this calling along with some appropriate responses to them.

We should note here that the computing techniques mentioned above are not uniquely Christian, nor are they constrained by any worldview. The effects of worldview will only be seen when we consider how to use the techniques, and what we conclude about our work when we are done. Similarly, many of the responses to the distortions discussed above are not uniquely Christian either. Non-Christians certainly enjoy creative work, and everybody believes in the value of elegant, secure, useful, and usable computing systems. Non-Christians, however, are likely to miss the spiritual motivations for computing, or, if they think about motivations at all, are likely to appeal to the timeless nobility of the human spirit and thus to see nothing of God's character in it. Christians, alone I think, can see through the process and the products of computing to the creative God that set it all in motion, and can appreciate the potential distortions and work to counteract them.

Computing educators should strive to identify the legitimate Christian roots of computing and to encourage students to be God's image bearers in the field. This will lead them into a fulfilling vocation in God's kingdom.

References

- Brooks, F. P. (1975). *The Mythical Man-Month*. Addison Wesley.
- Perlis, A. (1982). Epigrams on programming. *SIGPLAN Notices*, 17(9):7–13.
- Kurzweil, R. (1999). *The Age of Spiritual Machines*. Penguin, New York.
- Schneiderman, B. (2002). The new computing. <http://www.acm.org/ubiquity> 3(31), September 24–30.