BLOG @CACM

The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/3297799

<http://cacm.acm.org/blogs/blog-cacm>

Seeking Digital Humanities, IT Tech Support

Herbert Bruderer explains why the opposite of digital is not analog; Robin K. Hill describes how the challenges of user support are aggravated by indeterminate client responsibility.

Herbert Bruderer There Are No Digital Humanities

[https://cacm.acm.org/blogs/blog-cacm/232969-there-](https://cacm.acm.org/blogs/blog-cacm/232969-there-are-no-digital-humanities/fulltext)

[are-no-digital-humanities/fulltext](https://cacm.acm.org/blogs/blog-cacm/232969-there-are-no-digital-humanities/fulltext)
November 26, 2018

Digitization and the digital revolution are quite confusing. Probably most people believe digital is something new. Many think the opposite of digital is analog or mechanical. However, the forerunners of electronic or digital journals and books are printed works. I would not call them analog. Historians sometimes speak of a pre-digital era. Even museum experts are surprised when historical mechanical calculating machines are described as digital. For them, digital and electronic are synonymous. A new field of the humanities is named digital humanities.

However, the equation digital = new, analog = old does not work. Digital is not an achievement of the 21st century. Even the antique Salamis counting board (4th

century BC) was digital. The abacus is regarded as the oldest digital calculating aid. The Romans also used digital bead frames. Similar devices are offered today at flea markets. Digital calculating machines appeared in the 17th century (inventions by Wilhelm Schickard, Blaise Pascal, Gottfried Wilhelm Leibniz). In 1614, the Scotsman John Napier invented digital Napier rods, used for multiplication and division. Since the middle of the 19th century, mechanical calculating machines have been mass-produced in France (Thomas Arithmometer, patent 1820). Charles Babbage's (unfinished) analytical engine (1834) and a similar machine of the Spanish engineer Leonardo Torres Quevedo (1920) were also digital, as were the widely used punch card machines (Herman Hollerith, 1890).

Digitalization is therefore nothing new. The first mathematical instrument was not an analog but a digital device, the abacus. Significant phases of digitalization began in the 1940s and 1950s with the advent of relay and vacuum tube

computers. The shift from mechanics to electronics, which began mainly in the 1970s, replaced analog slide rules and digital mechanical calculators with digital electronic computers. For many years, analog and digital electronic computers competed against each other.

In my opinion, the humanities are neither analog nor digital. They increasingly using digital resources. It would be better to speak of computer-aided or computer-assisted humanities. The pre-digital era must have been before the Greek abacus.

Robin K. Hill Tech User Responsibility

[https://cacm.acm.org/blogs/blog-cacm/231489-tech-](https://cacm.acm.org/blogs/blog-cacm/231489-tech-user-responsibility/fulltext)

[user-responsibility/fulltext](https://cacm.acm.org/blogs/blog-cacm/231489-tech-user-responsibility/fulltext)
September 30, 2018

Some years of experience with faculty assistance has led me to speculate that the well-known frustrations of IT user support hide even deeper problems. Many of us with such experience know the chronic difficulty suffered by both client and consultant in the support scenario. Each day promises, and delivers, repeated problems, trivial issues, and deep misunderstandings attendant on the use of applications and devices. Users ask the same questions, individually and severally, over and over, requesting help when what they really want is someone who will do it for them. In my own experience providing technical support to faculty and also to members of a volunteer civic organization, I

deal with well-educated and competent people. Whereas most clients are cooperative and grateful, some are brusque and demanding, some are apologetic and jocular, many are just not listening.

On the tech support side, malfeasance includes overexplanation, underexplanation, incorrect explanation, and impatience, all transgressions of which I have been guilty from time to time. Why is this all so difficult? As the perceived burdens of technology build up on users, cheerful cooperation gives way to weary resignation and then to foot-dragging resentment. And this against an activity that is for their own good! Users resist reading manuals, or even short instructions, let alone working through a checklist, though learning the fundamentals would help them immensely. I have offered the briefest possible explanations of the client-server environment (“where your programs run”), HTML URLs (“how to reach websites”), and cloud storage (“where your files are stored”), to no avail. Direct orders, such as “Read this” or “Practice this,” even to people who are sincerely motivated (no matter their intelligence, job satisfaction, rank, or personality), have no effect. I have gradually come to the unsettling belief that this is not just exasperating, but revealing. (We acknowledge without comment the obvious possibility that I, and my fellow user support professionals, are just lousy instructors or repellent individuals.)

On the happy assumption that the average reader thinks the philosophy of computer science deals with lofty issues, this may seem pedestrian. Yet a problem so perplexing and intractable is ripe for a bit of philosophy. We might learn something about education or training from its apparent failure in such cases and thereby something about intelligence. We might learn something about the acceptance of responsibility from its apparent failure in such cases and thereby something about ethical duty.

As we look more closely (at naive users, at technically competent users, and even at us experts when we are faced with new technology), we see a reluctance to learn definitions, commands, good practices, and workflow. The hapless user does not build the cognitive scaffolding necessary to organize the concepts, so does not grasp which feature is relevant to what; that context is then even farther out of reach for the

consultant. Subsequently we see attenuation of commitment, where follow-up tasks are put aside until a better time, the initial momentum fades away, and the skills necessary for effective participation decay. This leads to an adversarial stance, where frustration morphs into resentment. Whose fault is this?

Although there is plenty of research and commentary on the responsibility of the vendor, there appears to be no inquiry into the responsibility of the consumer with respect to technology selection, mastery, and use. Should there be? Let’s interrogate some analogies: We impose a minimal degree of responsibility on someone checking a book out of a library—he or she should return it. The reading of it may be a norm, not an obligation. We impose a high degree of responsibility for driving a car, because it can kill people. We expect some degree of responsibility in the use of natural resources, because the effects are broadly dispersed. In domestic finances and budgeting, we assume the agent eventually will achieve independence, making unaided decisions and taking appropriate actions, out of self-interest. It’s not clear that any of those inform our view of the products of technology. Indeed, the very idea that software and hardware users have any responsibility toward their technology appears to stand in direct conflict to pervasive expectations on their part, as expressed thus:

This is a nuisance.

My duties involve real things, whereas this is just management of those things, not what I signed up for. Record keeping and bean counting should not take time from the job.

This is clerical.

These tools are complex, sure, and they require skill, the kind of skill embodied in a good secretary, who can handle tedium, the quirks, and the exceptions. But I deliberately avoided that career.

This is supposed to be easy.

These products are supposed to magically improve my life—vocational, social, and intellectual—immediately and painlessly. (This attitude, of course, is cultivated by technology vendors and promoters.) Because the product is fabulous, and intended explicitly for me, the trouble must lie with IT.

There’s not much in those expectations that can be corrected by user support staff. So where does responsibility

lie? Garrath Williams’s treatment of the notion of responsibility¹ notes the emergence of that notion only in the last two or three centuries, a brevity consistent with the lack of scholarship on client responsibility (also raising the question whether there really is any such thing). He locates responsibility not in the person, but in the multifarious modern world. “What is central is the moral division of labor created by our institutional fabric. This scheme of cooperation delimits the normative demands upon each of us, by defining particular spheres of responsibility. Given the fluidity, plurality and disagreement associated with normative demands in modern societies, this limitation is crucial.”

If there is a limit on each sphere of responsibility, then there should be a boundary on user support. Right now, no one understands the proper extent of support; no limiting structure is defined for the benefit of user or support staff. To define such a limit is to grant support staff authority to demur. Unthinkable as it may seem, modern technological society needs to consider, define, and sanction a point at which consultants can say “no.” Better yet, they won’t need to, because everyone will understand the limits; everyone will know where user support ends and user responsibility begins. Everyone will know that the manual should be read (and should be written in the first place), and they will know from accepted and ingrained cultural mores rather than from simply being told so by pesky IT people.

But we can’t work that out here and now! In the best case, the tribulation of tech help is a temporary issue, reflecting workplace stress in the face of upheaval, similar to legal and safety compliance demands. The problem will resolve as society grasps tech more firmly; that, however, will take time. We wait for the emergence of norms of responsibility in this and other aspects of technology. ■

Reference

1. Williams, G. Responsibility as a Virtue. *Ethical Theory and Moral Practice*. 11:4, 455–470. DOI: 10.1007/s10677-008-9109-7.

Herbert Bruderer is a retired lecturer in didactics of computer science at ETH Zürich. **Robin K. Hill** is a lecturer in the Department of Computer Science and an affiliate of both the Department of Philosophy and Religious Studies and the Wyoming Institute for Humanities Research at the University of Wyoming.