

“Honey, I Forgot the Users!”

Lori Pajerek
Lockheed Martin

We have all heard some human factors horror stories, and as consumers, most of us have had numerous personal experiences with products that are difficult, dangerous, or annoying to use. Their designers did not seem to have humans in mind. Poor human factors designs are the result of oversight or ignorance. The designers did not do enough work to ensure that the demands placed on the humans are not onerous. To address this shortcoming, all types of design engineers should learn to apply a systems perspective to their part of a system.

Which is better—mouse or keyboard? For example, since the introduction of the computer mouse, numerous interactive computer operations are now “point and click” even though the mouse is the slowest interface for performing many operations. It takes time to move the mouse into position and click it, especially when it must point to a relatively small area on the screen (such as a “soft button”). Although older technology than a mouse, using a function key when possible is not only much quicker, it also puts less wear and tear on your body.

This became apparent to me last winter when I used a new software tool intensively for extended periods. A certain operation required three mouse clicks every time: one to select the item on the screen to be acted upon, one to click the button bar to indicate the desired operation, and one to close a dialog box that asked, “Are you sure?” The three mouse clicks were in widely diverse areas of the screen, each required precise placement of the cursor in a tiny spot on the screen, and the computer took a lot of time to process each click.

Since I was performing this operation on many hundreds of individual items during the course of a day, I could not help but notice how slow the “mouse clicking” process was. Also, my fingers, wrist, arm, and shoulder ached at the end of the day. It was much more time-consuming than performing the same operation on a competing vendor’s tool, which took only one mouse click. Additionally, for most operations, the other tool often

gave me a choice between using the mouse or a function key. Having found that my wrist and arm got more tired and painful from repeatedly using the mouse, I prefer the function key when given the choice. Chalk one up for the competition.

This story illustrates how my personal experience led me to conclude the fewer mouse clicks, the better. But this is nothing more than a subjective opinion. The world loves mice. I may be the only person in the world complaining about too many mouse clicks. However, I recently saw some hard data from an internal IBM human factors study that empirically demonstrates that the mouse is the slowest of several common data entry devices. Yet, most software designers do not know this—they are often not aware that such studies exist. Because the mouse is newer technology than the keyboard, many of today’s designers—who grew up using a mouse—assume it is better. Even though graphical user interfaces (GUIs) invite the use of a mouse, it may not always be the optimal choice.

Human factors is an area where applying the systems perspective to engineering is key. The systems perspective requires looking at all parts of a system throughout all phases of system development and deployment and questioning how a system will be used. For human factors considerations, this means performing operational scenarios early in the system development to determine human interaction with the system, specifying human factors requirements, reviewing designs for human factors impacts, and modeling

the user interface in mock-ups. This activity should continue through system verification to validate assumptions, scenarios, and data. At this point in development, it is usually too late and too expensive to reverse any adverse design decisions—unless they seriously detract from the usability of the system.

The vendors of the “slow” tool described above may have developed operational scenarios, but unless they tested them with users under loaded network conditions, they would not have discovered what the implications were for the human using the system to perform the operation hundreds of times in a row. Nonetheless, it is a highly likely operational scenario for that particular tool.

The Personnel Subsystem

Recently, I learned a term that I had never heard, but one that I immediately liked, the “Personnel Subsystem.” Engineers have become better at remembering the human in the system. The user interface on any system usually has ergonomics requirements applied to it, and conceptual diagrams of systems often have a stick figure or clip art of a person representing the user(s). We know they are there, and we acknowledge their existence to some degree.

Specifications usually contain a section called “Personnel and Training”—it was required by MIL-STD-490. Generally, there are some requirements in the specifications; usually high-level statements that specify education, training, and skills for different users (operators,

maintainers, etc.). But while I have often seen an architectural element called "Training" in system architectures, I have yet to see a personnel subsystem, beyond the stick figure or clip art variety. It is the *forgotten subsystem*.

We know there are requirements that should be allocated to a personnel subsystem, yet we design this subsystem by default. Every time you specify or design a human interface, you are also implicitly specifying or designing the expected characteristics and behavior of the human half of the interface. Humans must perform to this implicit specification, but there is never a requirements review or design review of the personnel subsystem. Unless your system is never touched by human hands (a rarity), you have a personnel subsystem, whether you identify it explicitly or not. Just because you do not have to manufacture it does not mean it does not exist; it is a real part of the system integration picture.

By including a personnel subsystem in your system architecture, you are more likely to achieve the objective of looking at all parts of the system through all phases, including the people part of the system. Though we sometimes assign a few requirements to users, we often do not follow through. How do we know if we have "specified" a user with inappropriate or inconsistent requirements? You treat the personnel subsystem like other subsystems—review the requirements and verify the design.

It also is important to remember that consideration for the personnel subsystem is not limited to physical ergonomics. There are other factors involved that relate to the users' mental processes. These range from straightforward things such as consistent menu design to more subtle intangibles such as impact on job satisfaction.

Conclusion

Human factors is following a path that has been experienced in a number of other traditional engineering disciplines. Concepts like quality control, configuration management, and plug-and-play interfaces were all institutionalized in hardware engineering long before they were applied to software. Ergonomics has become a serious consideration for all types of hardware items. We have furniture, CRTs, keyboards, mice, wrist rests, and all kinds of other things that feature ergonomic design. In software, we now have windows and GUIs, but these features merely scratch the surface of what is possible. However, the systems perspective is what motivates someone to investigate the effects of the combined interaction of the ergonomic hardware and the user-friendly software with the personnel subsystem. What happens when a human sits for hours in the ergonomic chair, using the ergonomic mouse to click thousands of times on the software-generated GUI? All the pieces were designed to be kind to humans, but the integrated system still causes the user to go home in the evening with eye-strain and a sore neck. When designers think like users, the result is a system that is truly user friendly. ♦

About the Author

Lori Pajerek is an advisory systems engineer in the Systems Engineering Technology department at Lockheed Martin Federal Systems in Owego, N. Y. With over 15 years experience in systems and software engineering for defense-related industries, her specific area of interest and expertise is requirements engineering and requirements management. Prior to joining Lockheed Martin, she was employed at Link Flight Simulation (now part of Raytheon), the world's leading manufacturer of flight simulation devices. She has a bachelor's degree in mathematical sciences from Binghamton University and is a member of the International Council on Systems Engineering.

Lockheed Martin Federal Systems
1801 State Route 17C
Mail Drop 0902
Owego, NY 13827
Voice: 607-751-6226
Fax: 607-751-2008
E-mail: lori.pajerek@lmco.com

Systems Engineering at the STSC

The newest systems engineering technologies can be confusing—SECM, SECAM, SE-CMM, CMMI, 632, 731, 1220, 15288. The Software Technology Support Center can help you sort through the jumble. We partner with you to identify, evaluate, and adopt effective technologies that improve product quality and process efficiency and predictability. Our consultants work directly with you to tailor proven processes, methods, and tools to your organization's needs.

- **Process improvement:** assess, plan, implement, and lead improvement efforts that keep you competitive.
- **Technology evaluation:** identify, pare down, evaluate, and select what best meets your needs.
- **Technology adoption.** Understand, apply, and exploit technologies to improve quality, productivity, and customer satisfaction.

Although our information is free, we provide on-site briefings, training, workshops, assessments, planning, and guidance on a cost-recovery basis. The STSC specializes in

- Requirements definition, design, documentation, test, and reengineering.
- Acquisition, project, and configuration management.
- Assessments and metrics.

Call us for your systems engineering needs.



Randall R. Wright
Voice: 801-777-9732
E-mail: wrightr@software.hill.af.mil

Russell Lee
Voice: 801-775-5740
E-mail: leeru@software.hill.af.mil