Cognitive Engineering: It's Not What You Think

Stephanie Guerlain
Department of Systems and Information Engineering
University of Virginia

What is cognitive engineering? It is not brain science, nor is it cognitive science, although that's closer. It is neither artificial intelligence nor neuro-engineering. It's not robot design or perhaps many of the other things that may come to mind when you hear the term "cognitive engineering". However, it definitely includes aspects of all of the above fields and many others, including psychology, anthropology, computer science, design, and systems engineering. Simply put, it's about the understanding of and the design of systems that require human intellectual work. Since just about any activity that involves humans involves human intellectual work, it follows that cognitive engineering can be applied to just about any human activity. I've even written a paper describing the cognitive engineering aspects of competing a horse in a cross-country jumping competition, had it published in a scientific journal, (Guerlain, 2001). This got mixed reactions. Some people asked me, what next? Golf??

Actually, cognitive engineering methods could be applied to just the activity of WATCHING a horse competition (see White, Odioso, Weaver, et al., 2008 for an example). This would be of course risky for a junior level faculty member to do, since that seems awfully far from engineering for those reviewing tenure packages. Despite these seemingly "non-engineering" applications, in fact, the field has grown up around the much more practical and higher-impact aspects of understanding human work activities in complex domains, such as aviation, medicine and nuclear power control where poorly designed systems can lead to major accidents. The birth of the field is largely in response to accidents happening in large, complex systems that had seemingly nothing to do with the design of those systems. In other words, there was no mechanical failure. The computers and other automation, processes, etc. worked as designed. An example is the aviation accident term "CFIT", which stands for "Controlled Flight into Terrain". In other words, a highly trained cockpit crew flies a perfectly good plane into the side of a mountain. This has happened often enough that it has its own acronym.

So, a first response to the cause of such accidents is often labeled as "human error". However, by studying the causes of those accidents further, or even just by studying the day-to-day activities that people engage in when using the systems designed for them, it turns out that the system just not designed well to begin with. Thus, the computers, automation, and other engineered processes (such as procedures, handoffs during shift change, logbooks, regulatory requirements and other aspects of information passing amongst the people and computers involved) have weak spots, places where a certain set of events can co-occur to cause failure. Ironically, quite often the people working day-to-day in the system see such failure modes (although they may not think of them in that way) and create "workarounds", such as placing sticky notes to remind them what to do or not do, or develop an almost "intuitive" understanding of how to react if and when things start to go wrong. These workers, as important sources of knowledge, are often overlooked by engineers, often due to a lack of training in how to gather such requirements well. This is where cognitive engineers excel. Cognitive engineers do not just focus on interviewing and observing end users, but we also look at the intrinsic system requirements of the task. For example, for air traffic control, it is a fact of the task at hand that multiple planes are moving at varying (but constrained) rates of speed and altitude. Thus, one can not ignore these facts, but can in fact often take advantage of the constraints in this system knowledge when designing representations and other aspects of decision support systems.

Cognitive Engineering is a sub-specialty of the broader field known as "Ergonomics". When most people think of ergonomics, they think of physical ergonomics, such as making a tool or environment more "ergonomic". An early example of this was the design of the Reach toothbrush (Hill and Kreifeldt, 1979), which then spawned a whole new field of "toothbrush design" instead of the straight - handled, rectangular-shaped toothbrushes that were always the norm prior to the introduction of this more "mouth-friendly" technology that was designed to better accomplish the required task of cleaning all aspects of your teeth.

Both physical and cognitive ergonomics are important, and the same environment or system can be analyzed and improved upon from both perspectives. Table 1 gives a few examples of how a cognitive ergonomist or a physical ergonomist might analyze the same system from these two different perspectives. In general, a physical ergonomist is focused on creating an environment that is safe for those who work in that environment, while cognitive ergonomists focus on creating an environment that maintains overall process safety, e.g., by minimizing the chances for "human error". Both analyses are important because improvements in either of these aspects can yield significantly less downtime due to less worker injury, faster overall performance, often due to fewer steps required, and increased worker satisfaction.

Table 1. Physical vs. Cognitive Ergonomics

Human Activity	**Physical Ergonomics**	**Cognitive Ergonomics**
Will sitting for 8 hours…	…cause back pain?	…cause loss of attention?
Will excessive noise….	…cause hearing loss?	…cause operators to miss a request?
Do the operator displays…	…cause eye strain?	…cause a misunderstanding of the situation?
	Worker Safety & Risks	*Process Safety & Risks*

Many people claim that ergonomics is just "common sense" but given the number of engineered systems which do not take into account human capabilities and limitations, and do not actually fit task requirements, I often claim, "It's unfortunately not that common." As a simple example, take the challenge of finding all apartments for rent (that allow pets) within 5 miles of a particular location. This task, while easily specified, is an almost impossible task to achieve with today's search engines, not because the search engines could not accomplish this task, but because they are not set up to run such a query and thus users must endlessly search, type, click, mouse, zoom, scroll, page, phone, bookmark, write notes, etc. when in fact you can imagine a two-step process where a well-designed system would allow such a query, that would then return a map and driving directions, price and everything else you might want to know in one easy result that could be printed out or downloaded directly to your GPS system. Thus, the technology is often available but those who design systems often do not really understand user or task requirements, thus causing all sorts of work-arounds, time, effort, and errors, or worse still, a "giving up" by users.

This is especially true in today's healthcare system, where it requires so much effort to gather together the records and relevant information for a patient (particularly one that's just moved to the area or that's just been admitted to the emergency room) that doctors most often rely on asking the patient for their health history. Even if all records have been sent to a hospital, the data are not easily sorted, digested, or summarized. It's just pages and pages of often repeated information in a very complex and large medical record. Electronic medical records are just now starting to emerge, but they are unfortunately not at all patient-centered. In other words, patients move all over the country and get healthcare from lots of different places even if they live in one state. However, the electronic medical record systems are being implemented piecemeal, and are only integrated within a single healthcare institution.

One mantra of cognitive engineering is to design for data extraction, not just data availability (Hollnagel, Mancini and Woods, 1986). Efficient data extraction by people often means having the data pre-organized and represented in such a way that people can use their pattern recognition skills to directly "pick up" on the answer they are seeking in a very efficient, "parallel" way (e.g., more data displayed does NOT require more time to search). Thus, analog instruments in a car can be read quickly, because one only need to look at the dial and see if it is "in the red"; there is no need to convert a particular number into the "state" of "the car needs more gas soon". Similarly, in control rooms, operators often tend to fill at least one of their multiple monitors with trends of key process parameters, because they are able to interpret patterns in those trends to detect important process events and to then use this overview information to navigate to detail displays as appropriate.

Cognitive engineers consider all the inputs, outputs and decisions that task demands require, and help inform or lead design teams as to what should be "automated" what should be displayed, in what way, and in what order. Cognitive engineers also consider the design of feedback systems to the human operator(s). The auto-pilot system in an aircraft, for example, does not need to pull back on the yoke in order to make the plane go up. However, even in auto-pilot mode, the yoke does "pull back", for the sole purpose of providing feedback to the pilots to the changing state of the airplane. This is something that can be seen in peripheral vision, while performing other tasks, unlike the need to focus attention on a particular dial or instrument panel to see if a number has changed on that display. Thus, not all feedback is the same and these concepts are considered by cognitive engineers when designing systems.

Cognitive engineers also consider the context of use in which a system will be used. If people will be working in a highly noisy (e.g., industrial) environment, then relying on a "beep" is not a very good design to capture attention. Other contexts to consider are the state of the people who will be using the system. How can we design a system that will accommodate all levels of potential users? Can we design the system in such a way that, through its use, people will be able to use it right away and then become better at using the system rather than relying on an extensive training effort?

In general, the design process is an iterative one that focuses on understanding the cognitive requirements and constraints inherent in the system, designing prototypes, testing those prototypes for usability and iterating on the designs until production. This human-centered design process is one that is often skipped, either due to ignorance of the approach or perceived lack of time or funding to do so. Usability experts might be brought in after implementation has been completed, but in fact a usable system is one that meets the actual requirements. Figure 1 shows how far down in the process implementation should start and, conversely, how early in the process task analysis and iterative design and testing should start.

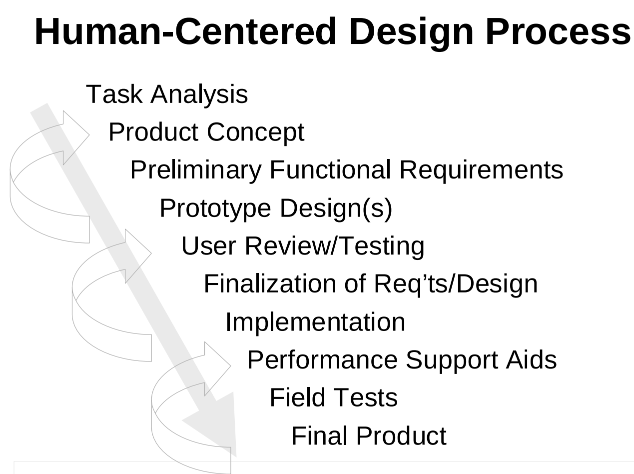

Figure 1. The Human-Centered Design Process

This brief article has not yet touched on the many other aspects of cognitive engineering, such as cognitive modeling, team processes, and designing metrics of performance. The future of the field is quite rich. From a practical perspective, there are boundless applications to current and evolving work practices. From a theoretical perspective, much remains to be understood on creating decision support systems that are flexible enough to support a broad range of applications yet have the automation capabilities to put the data together into a way that directly meets task needs without the need to point, click, zoom, program, etc. extensively. The most direct benefit could be yielded in the healthcare arena, where biological and healthcare data could be combined to create powerful decision support systems, but so much infrastructure, design and likely even legislation needs to take place before that can happen.

References

Guerlain, S. (2001, Summer). Judging pace cross-country. How cognitive task analysis yields insight into an apparently simple, yet complex horse-rider activity. Ergonomics in Design, 9(3), 13-18.

Hill, P. and Kreifeldt, J. (1979). Toothbrush - US Patent D251,038.

Hollnagel, E., Mancini, G. and Woods, D.D. (1986). Intelligent Decision Support in Process Environments. New York, NY: Springer-Verlag.

White, M., Odioso, M., Weaver, M., Purvis, M., Bass, E. J. and Bruce, S. (2008). Horses? There are horses at Foxfield? An analysis of college student hazardous drinking and related decision making behaviors. 2008 IEEE Systems and Information Engineering Design Symposium, April 25, 2008, University of Virginia, Charlottesville, VA, USA.