Distinguishing between control and collaboration—and communication and conversation

In their paper “from Autonomous Systems to Sociotechnical Systems: Designing Effective Collaborations,” Kyle J. Behymer and John M. Flach remind us “the goal of design is a seamless integration of human and technological capabilities into a well-functioning socialtechnical system.” \(^1\) Recent trends—the sensor revolution, big data, machine learning, and intelligent agents, for example—make their reminder timely.

However, the idea of “seamless integration” has a history in design discourse and discourse about computing. Architect Christopher Alexander made “fit” the organizing concept of his first book.\(^2\) HCI pioneer Douglas C. Engelbart focused his life’s work on “augmenting human intellect,” which he described as “increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems ... in an integrated domain where hunches, cut-and-try, intangibles, and the human ‘feel for a situation’ usefully coexist with powerful concepts, streamlined terminology and notation, sophisticated methods, and high-powered electronic aids.” \(^3\) Computer pioneer J.C.R. Licklider wrote about “man-computer symbiosis,” which he described as “cooperative interaction between men and electronic computers.”\(^4\) And architect Nicholas Negroponte explored the possibility of building machines that could collaborate with designers, stating that “the partnership is not one of master and slave but rather of two associates that have a potential and a desire for self-improvement.”\(^5\)

Negroponte made a critical distinction—a master “controlling” a slave differs substantially from one colleague “collaborating” with another. The first is exploitative—the second is generative. Historically, most discussions about man-machine interfaces have been framed around control loops, treating machines as slaves. What’s fascinating is that more than 50 years ago, Licklider and Engelbart envisioned an alternate, more humane relationship, a collaboration between man and machine.

Behymer and Flach build on the idea of collaboration, proposing a model comprised of actors or “agents,” both human and “automaton.”\(^6\) In this model, each agent is part of a simple control loop with sensor (“Perception”) and actuator (“Control”) flowing through a “Wicked Problem Domain.” This simple control loop is a useful approximation of designing, first introduced in the mid-1960s.\(^7\) Behymer and Flach add multiple actors and communications between them. Behymer and Flach also note that the quality of the communication between the actors determines the quality of perception and control of the entire system—with “rich” communication, the whole can be
more effective than any of the parts. This model is helpful. However, we would like to raise four issues. First, the model incorporates an element labeled “Wicked Problem Domain,” through which the control loops pass. While wicked problems are very important, it may make sense to treat less complex classes of problems first. The second figure in the paper introduces an element labeled “Complex Work Domain,” which might easily be substituted for “Wicked Problem Domain” in the first figure. That way, we might avoid suggesting that the systems given as examples in the article interact with “wicked problem domains.”

A model that treats wicked problems must grapple with their “wickedness.” Horst Rittel taught us that wicked problems “are inherently different from the problems that scientists and perhaps some classes of engineers deal with ... which are definable and separable and may have solutions that are findable.... [Wicked problems] are ill-defined; and they rely upon elusive political judgment for resolution.”

Second, the model does not include a “goal,” a key component of any control system model. Goals might be presumed inherent in the actors (human or automaton); however, the question arises: Where do their goals come from? The goals could be taken as given for simple problems (as in student assignments or entry level jobs). Yet, most professional work involves agreeing on goals. And what makes wicked problems intractable is the great difficulty of agreeing on goals (i.e., the problem framing).

Third, the model does not say much about the nature of communication between the actors, except that it should be “rich” and “effective.” How do we achieve that?

Negroponte points to “conversation,” having included a section by cybernetician Gordon Pask on Conversation Theory in his book *Soft Architecture Machines*, which describes “(1) the computer as a designer, (2) the computer as a partner to the novice with a self-interest, and (3) the computer as a physical environment that knows me.”

Pask's model of “conversation” is worth distinguishing from Claude Shannon's model of “communication.” Shannon described a process of sending signals. Pask describes higher-level processes, whereby learning systems (including people) make distinctions, share and understand them, agree that they understand, and then act on their agreement. His model further distinguishes between conversations about goals and those about means, and it might be expanded to conversations about creating new language and new processes needed to deal with new challenges (or disturbances) that require “innovation.”

Fourth, while the paper elsewhere introduces cybernetician Ross Ashby's very useful concept of “requisite variety,” the model misses an opportunity to connect to the idea. Variety is a property of the system's sensors and actuators—and also a property of the environment.
3 Distinguishing between control and collaboration—and communication and conversation

Endnotes


