she ji — *The Journal of Design, Economics, and Innovation* — Volume 2 — Number 2 — Summer 2016

Distinguishing between control and collaboration—and communication and conversation

Hugh Dubberly — Dubberly Design Office — hugh@dubberly.com
Paul Pangaro — College for Creative Studies — ppangaro@collegeforcreativestudies.edu

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."[8]

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).[9]

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."[10]

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,"[11] 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.

Variety refers to the capacity of a system to maintain itself in the face of disturbances. All systems have limitations—that is, even the largest and most robust system can be overwhelmed, given a larger and more robust disturbance. Requisite variety refers to the capacity required to overcome disturbances the system is "likely" to encounter. When an automated system is overwhelmed, human operators must come to its aid. We might say that the automated system lacked variety, and the humans increased the variety of the combined system. Deciding how much variety to include is a design decision. Are we designing for a hundred-year flood or a magnitude-8 earthquake? We weigh the likelihood of the disturbance against the cost of including the variety required to resist it.

We may apply the concept of requisite variety to higher-order learning systems—teams, AIs, and sociotechnical systems—while noting that Ashby originally developed it to describe first order systems. Behymer and Flach's essential argument is that neither technology systems alone nor human systems alone have as much variety as both systems might have together, particularly if they are coordinated well. This is the argument for diversity on teams. And while it brings benefits, variety also has costs. As the number of team members grows, so does the complexity of their language, the potential for miscommunication, and thus the need to design conversations.

Endnotes

1
Kyle Behymer and John M. Flach, "From Autonomous Systems to Sociotechnical Systems: Designing Effective Collaborations," *She Ji: The Journal of Design, Economics and Innovation 2,* no. 2 (2016): 113–14.

2
Christopher Alexander, *Notes on the Synthesis of Form* (Cambridge, MA: Harvard University Press, 1964).

3
Douglas C. Englebart, "Augmenting Human Intellect: A Conceptual Framework," Stanford Research Institute Summary Report AFOSR-3223 (Menlo Park: SRI, 1962), accessed October 16, 2016, http://www.dougengelbart.org/pubs/augment-3906.html.

4
Joseph C. R. Licklider, "Man-Computer Symbiosis," I*RE Transactions on Human Factors in Electronics* HFE-1, no. 1 (1960): 4.

5
Nicholas Negroponte, preface to *The Architecture Machine* (Cambridge, MA: MIT Press, 1970).

6
Behymer and Flach, "Autonomous to Sociotechnical," 105–14.

7
See for example John Chris Jones, *Design Methods,* 2nd ed. (New York: John Wiley and Sons, 1992); Horst W. J. Rittel, "The Universe of Design: Faculty Seminar," College of Environmental Design, Institute of Urban and Regional Development, University of California at Berkeley, 1964; and Don Koberg and Jim Bagnall, *The Universal Traveler: A Soft-Systems Guide to Creativity, Problem-Solving, and the Process of Reaching Goals* (Los Altos, CA: W. Kaufmann Inc., 1972).

8
Horst W. J. Rittel and Melvin M. Webber, "Dilemmas in a General Theory of Planning," *Policy Sciences 4,* no. 2 (1973): 160.

9
Peter G. Rowe, *Design Thinking* (Cambridge, MA: MIT Press, 1987).

10
Nicholas Negroponte, author's note to *Soft Architecture Machines* (Cambridge, MA: MIT Press, 1975).

11
Ross W. Ashby, *An Introduction to Cybernetics* (London: Chapman Hall, 1956). A further interesting connection exists between Ashby and Alexander, who relied heavily on Ashby's Introduction to Cybernetics in his Notes on the Synthesis of Form.