

Finding the Simplicity Embedded in Complexity

My friend, the amazing designer and facilitator, [Gail Taylor](#), often exhorts people to find the simplicity embedded in complexity. I've been puzzling over this advice over the past few weeks, as I've been reflecting on our own practices for working with groups on complex problems. I recently came across two resources that I'd like to share.

The first is from my friend, [Jeff Conklin](#), who [first introduced me](#) to the notion of [Wicked Problems](#) a dozen years ago. This past October, he and [Ian Mitroff](#) gave short presentations on Wicked Problems to the [Delta Conservancy](#). The hour-long video is well worth watching for quick, accessible introductions to Wicked Problems and [Russell Ackoff's](#) notion of messes:

I put off watching this video for a while, because I've had so many conversations with Jeff about Wicked Problems over the years, I didn't expect to hear anything new. But as I listened this time, I picked up on a nuance that I think has gotten away from me over the years.

One common way to address Wicked Problems is to use some kind of visual mapping tool to help get a sense of the system as a whole. My partner, [Kristin Cobble](#), is an avid practitioner of [system mapping](#), I love to use [Dialogue Mapping](#) (which I learned from Jeff), and we regularly partner with other visual practitioners to help with sensemaking.

The mistake that people often make with these approaches is that they can easily sucker you into the very thing you're trying to avoid: an attempt at linear problem-solving. The goal of mapping is not to develop some rigorous model of a system as a whole, because the whole point of Wicked Problems

is that you cannot develop such a model. The goal of mapping is to help develop *shared understanding*.

There are lots of ways to develop shared understanding, but the core of all such processes are dialog and listening. Mapping tools are only useful if they support that dialog and listening process. It's easy to lose sight of this with visual modeling languages, because they can lull you into thinking that your models are objectively correct, which can then shift your focus away from building shared understanding and onto the models themselves.

Developing shared understanding is also often viewed as a first stage in a linear process, which is also a mistake. You develop shared understanding through a continuous process of dialog, listening, interacting with the system, and learning. Dave Snowden describes this beautifully in his blog post, [“The Aggregative Error,”](#) where he makes a key distinction between breaking down a system into “simpler” parts versus attacking the system in multiple ways. The former is a misguided process of linear model building. The latter is about building shared understanding through iteration and interaction.

Last year, we were helping a client understand how it could collaborate more effectively, and we designed a process that was a cycle of parallel, quick experiments followed by group reflection. Our client signed off on the process, but we couldn't get any traction within the organization to implement it. We ended up doing a more traditional case study, although we made our research and learning a participatory process, which was a different and valuable experience for our participants.

On the one hand, it was a reminder of how unintuitive and challenging it can be to truly embrace complexity in our design processes, even for people like us who live and breathe these ideas. On the other hand, it reminded me of the

importance of Gail's advice: Find the simplicity embedded in the complexity. As Dave warns, you don't achieve this by pretending the complexity away. Instead, embrace the complexity. Embrace the fact that the system is too complex and nuanced for one person to grasp, and make it a collective process. Assume that you'll need to play with the system to have a chance at understanding it, and that you'll hit dead ends more often than you'll see the light.

The simplicity is this: At the end of the day, it's about sensing together, playing together, learning together. We *know* how to do this. Start there. To the extent that tools and processes and frameworks help you do this, embrace them, but never lose sight of what you're doing and why.