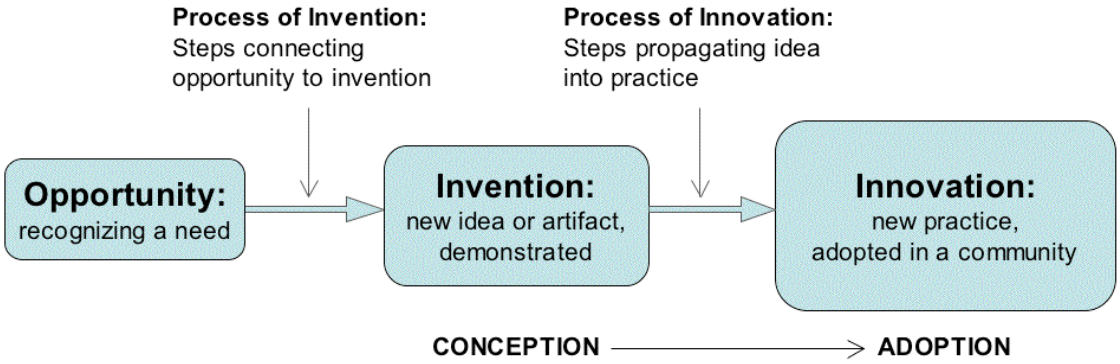
Comments on Diffusion and Generation

P. J. D.
5/23/05

In CS4900, our model of innovation has this superstructure:



In *Diffusion of Innovations*, Everett Rogers offers the same superstructure with different terminology. This table shows the correspondence.

Concept	CS4900	Rogers
Invention	Creation of a new idea, practice, or artifact	Process by which a new idea is discovered or created
Innovation	Adoption of a new practice by a group or community	An idea, practice, or object perceived as new
How inventions are generated	Practices of invention	Scientific research and development
How innovations are generated	Practices of innovation (especially, adoption)	Diffusion

We chose the CS4900 terminology because it is closer to common usage of the terms. But, unfortunately common usage is ambiguous. By deviating from common usage, Rogers avoids ambiguity.

Rogers tells us many aspects of the adoption (diffusion) process that are worth knowing. For example,

- Diffusion takes time. Generally, the more perceived value, the faster the adoption.
- A population divides into groups of different levels of receptivity to the new idea -- inventors, early adopters, majority, late majority, laggards -- with progressively longer adoption times.
- The decision process for adoption depends on numerous factors including peer recommendations, authority mandates, compatibility with existing practice, relative advantage, and trialability.
- After adoption, there are positive and negative consequences of the new practices. An adoption with too many negative consequences will be considered a "bad innovation" and will be discontinued.

These are all useful distinctions when you are doing the cognitive tasks of innovation: building a strategy, planning a marketing campaign, or figuring out how to make allies of laggards. But are they useful when you're in the thick of the action, experiencing mostly chaos and uncertainty? Will you be able to move effectively?

Why We Want a Generative Framework for Innovation

Peter Drucker and Everett Rogers have both offered frameworks for understanding innovation. Drucker is an expert on management and leadership. His framework is theoretical-descriptive, meaning he offers a theory and claims that it describes all the real cases of innovation. He says that the principles of his theory are the basis of a discipline of innovation, which can and should be taught in schools. He says that entrepreneurs are the primary practitioners of the discipline.

Rogers is a social scientist. His framework is empirical-descriptive, meaning that he has detected recurrent patterns from careful statistical analyses of many innovations. He does not claim there is a discipline of innovation. He mostly lays out new research questions that innovation scholars can investigate in order to continue refining the framework. He considers the framework to be an accurate description of natural phenomena occurring in human social systems.

Both Drucker and Rogers exemplify descriptive frameworks. However, the moment we ask what innovators must do to bring forth their results, we find little guidance from their frameworks. We are asking for a generative framework that exposes the ways in which individuals generate actions that produce innovation.

A descriptive framework will focus on explanations of observable outcomes, in some language or notation. A generative framework will focus on the embodied practices by which individuals accomplish the outcomes. A descriptive framework focuses on the "what" ; the generative focuses on the "how".

A descriptive framework is constructed after the fact: it tells you how things will look when the action is done. It will help you to anticipate issues that may arise, but not how to deal with those issues. It may not help you in real time to know where you are in the process that you are trying to deal with.

Descriptive frameworks for innovation tend to focus on the creation and propagation of ideas. Ideas are easy to describe. In fact, an idea *is* a description of some sort. Generative frameworks focus on the creation of actions and the fulfillment of commitments. We like the language-action framework because it tells us how we use language, especially speech acts, to create new actions and commitments.

A few examples will help to illustrate the difference between descriptive and generative frameworks. We have discussed the problem with virtual reality simulators used for training. No matter how good the simulator, it seems that there is a gap between what the trainee learns from the simulator and needs to know in the real situation. A substantial amount of research is being devoted to finding out how to narrow that gap, and what additional training a person needs to become a good performer in the real situation. Professor Rudy Darken gaves an example of a simulator that trained a team on how to clear a building of enemy combatants in urban warfare. When the team had reached the top level of performance in the simulator, it was taken into a real, mock-up building. Their guns shot paint rather than bullets. The team lost their smooth coordination; they shot their own teammates; they stood exposed in doorways where they were felled by enemy fire. The conclusion was that important aspects of the real situation were not (or could not be) captured in the simulator -- for example, the subtle sounds of breathing, the warmth and motion of nearby bodies.

The gap exists because the simulator inherently deals with descriptions of the real situation. Descriptions are of little help in learning practices. To complete their training, the simulator's graduates also need to practice with the real thing.

A similar situation occurs with flight simulators. The pilot learns various cognitive tasks well, such as interpreting instruments and knowing how to adjust levers and controls. But the tasks that require a "feel" of the way the aircraft is moving, say in wake turbulence or engine loss, cannot be learned in the simulator. Helicopter pilots say that it takes practice to get the feel of the subtle stick movements that stabilize against side-to-side oscillations; simulators do not teach this.

We have conducted a group exercise in which each member presented a claim of their own expertise to the group, and then received feedback on whether the group accepted the claim and trusts their expertise. We reviewed the "theory" with the group beforehand, giving precise descriptions of grounded assessments and the various conditions that an audience will look for when deciding whether to accept an assessment. Most group members said that they were familiar with the theory. When they stood before the group, however, many were surprised to discover that they were not able to convince people to accept their claim. Many concluded that they would have to practice giving assessments and adjusting to feedback. The people who do this this exercise discover that there is a big difference between the theory of a grounded assessment and the actual performance of it. You can have a cognitive understanding without having a corresponding embodied practice to make it effective.

Another group did a group simulation of routers passing data through a network. Despite having a clear picture of the routing algorithm, it took them a few minutes of practice to actually carry out that algorithm without dropping the balls representing the data. As the external authorities demanded progressively higher performance from the network, they improvised new practices on the spot, discarding those that did not work and sticking with those that did. Their improvisations were spontaneous and did not involve creating ideas that they then put into practice. If anything, it was the other way around: afterwards they constructed descriptions of what they had done and labeled them as "new ideas".

Many musicians, artists, performers, and athletes train so that they can be great improvisers. They spontaneously create new practices without creating ideas. They often cannot describe how they do their improvisations. Artificial intelligence researchers found the same problem in building expert systems: experts are unable to describe how they generated the actions they are so good at.

Another way to understand the difference between descriptive and generative frameworks is with the analogy of the journalist watching a ball game. The journalist has a detached view of the play and is not involved in it. The journalist can write stories of what the play looked like and can speculate about what the players might have been thinking (often far wide of the mark). The journalist may have many opinions about how the players might have done things differently to win the game, but of course those opinions could not be translated into effective action. Think of the baseball journalist's description of a home run: "Ruth took a hard swing at the 100 mph pitch with a perfect centered fluid twist of his torso and snap of his wrists, launching the ball on a graceful parabolic trajectory into the upper bleachers." After the game, the journalist gets a different perspective from talking to Ruth: "I don't know exactly what I did. I can tell you that that ball came in real slow and just hung there in midair for the longest time. There was no way I cold possibly miss it!" Ruth's coach might say: "It does no good to tell Ruth how to hold the bat or snap his wrists. Instead, I teach him to let time slow down so that ball appears to float in the air when he whacks it."

The following story illustrates how I lost my ability to play good golf after trying to improve my game with the help of a great description. By 1970 I had stagnated at a 10 handicap; but I could make no further progress. One day fellow engineer who heard my laments gave me a new paper from a physics journal. The author had found a perfect model of a golf swing, consisting of a rotating arm with a freely swinging pendulum attached to its end. By getting the torque accelerations of the rotating arm right, the pendulum would whip and hit the ball perfectly at a very high speed, reproducing exactly what pro golfers swings do. The description appealed to my mathematical engineer's mind. From that point on, the image of myself as an arm with a pendulum was lodged in my head. It completely ruined my swing because my body is not a rotating arm connected to a free pendulum It cost me hundreds of dollars of lessons with a pro just to get back to my old rut.

Good coaches know that feeding someone a description will not help the person learn to generate certain actions well. Instead, the coach tries to work with the person's practices. The coach reveals the person's practice to him, and then works with the new perception to move toward better performance. For example, an inexperienced ski instructor might tell the novice to turn by shifting weight to the inside. The experienced instructor will ask the novice to try a turn and tell the instructor what he noticed. The instructor will then suggest something

else to notice and get the novice to try again until he does. Only when the instructor can bring the novice to perceive the world differently, can he help the novice generate actions more effectively.

There's an old story about three umpires discussing how they make calls. The young umpire says, "I calls 'em as they are." The middle-aged umpire says, "I calls 'em as I sees 'em." The old seasoned umpire says, "What I calls, is 'em."

The young guy is working with descriptions of reality. The middle guy knows he is only an observer. The old guy generates reality.

Completing the Circle

Now we return to our original purpose. We are aiming toward helping you understand what is involved in generating a culture of innovation. Your master's thesis is supposed to start an innovation and give you a taste of what it means to participate in a culture of innovators. The process of doing your master's thesis will, as you will discover, contain all the action groups mentioned in the CS4900 innovation map that we gave you. Your faculty advisor will be your coach as you move through this process. You will be responsible for generating the individual moves. If you try to leave an action out, you will encounter a delay until you restore it.