

## Week Six Response

Aaron Bell

## Week 6 reading response

The relationship between theory and computer programs is aptly stated by Frijda; while a program can represent a theory, a program is not necessarily theory itself. All programs are models of a theory, basing their actions on theorized concepts, but they do not necessarily execute the theory directly. Programs are designed to work efficiently and methodically, and as such, they may often take short cuts to emulate the processes of a theory, but not actually carry them out directly. It is not necessary that a program replicate the structure of a theory in order to arrive at the same basic results. To a large extent, this is a function of computers' highly procedural nature, a quality that Weizenbaum largely chooses to ignore. Because computers are procedural in their processes, they cannot replicate the many non-linear processes of human beings (be they cognitive or otherwise). There are some processes which we understand on a theoretical level, but simply cannot be replicated by a computer; this is a prime example of how Frijda's theory-program relationship conflicts with Weizenbaum's view of computer modeling capabilities. Furthermore, what exactly qualifies as understanding? We may understand a process on the level of Newtonian mechanics, but can we explain it on the biological level, atomic level, or quantum mechanical level? Even when we think we understand something completely, it requires more than a superficial knowledge of the operating principals in order to truly create a comprehensive and program and theory.

A key aspect of computers in psychology which Weizenbaum touches upon is the impact that these "tools" have had on humans. Like any useful tool, humans use computers to change the world around them, and in turn this forces man to view himself through the lens of the new reality he has created. Tools are an essential factor in human adaptation and this facet of their influence cannot be overlooked. The discussion of programs which can become instructive is a step in the right direction, possibly leading down the road to signs of cognition, and eventually true artificial intelligence. If a program can carry out tasks which it was not directly programmed to do, then it is displaying self learning, which begs the question of intelligence.

The dilemma of *problem space*, or basically a measure of the complexity of a problem, is critical to programs. Programs cannot easily estimate the complexity, and the resulting required problem space. If a program cannot judge how much space to allot a problem, there is no way that the program can execute a process in a humanlike manner. Another problem of simulating human thought and logic processes is the question of the metric which these works are measured against. The age old comparison is the Turing test, which is highly biased towards the quirks and idiosyncrasies of humans. Modeling these qualities, while it may make for a program that can beat the Turing test every time, what level of insight does this program give scientists into the true nature of human thought and cognitive processes? The true measure should be based on the replication of true human thought, not the ability to disguise itself as a human. These Turing test driven programs are like mannequins, they look a lot more like humans than robots on an assembly line, but the assembly robots are structurally similar to humans in far more ways than the hollow plastic models.