Symptom Management for Schizophrenic Agents

Phoebe Sengers
Department of Computer Science and Program in Literary and Cultural Theory
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213-3891
phoebe@cs.cmu.edu

Behavior-based paradigms are a promising avenue towards creating full-blown integrated autonomous agents. However, until now they have had a major stumbling block: programmers can create robust, subtle, and expressive behaviors, but the agent's overall behavior gradually falls apart as these behaviors are combined. For small numbers of behaviors, this disintegration can be managed by the programmer, but as more behaviors are combined their interactions become so complex that they become at least time-consuming and at worst impossible to manage.

One of the characteristic modes of breakdown that occurs in such an agent is that it engages in stereotyped behavior with abrupt switching between relatively homogeneous modes of behavior. I term this symptom, inspired by cultural theory, *schizophrenia*, and identify its cause in a methodology of *atomization*. Atomization is the reduction of a complex and not necessarily well-defined phenomenon to a set of relatively simple parts with limited interaction; it is what makes behavior switching so abrupt. Unfortunately, the implementation of agents depends on this reduction, since complicated, wholistic agents are nearly impossible to design, build, and debug. While this may mean that we must build atomistic agents, it does not necessarily consign us to the schizophrenic scrapheap—provided we take a fresh perspective.

Cultural theory suggests that when agents appear schizophrenic, it may be because their social and cultural environment is ignored. While alternative AI (including Artificial Life, situated action, and behavior-based AI) insists that agents must be thought of in terms of their environment, this environment is usually thought of only in terms of the physical objects the agent encounters, leaving out the designer and audience of the agent. The problem with ignoring part of the agent's environment is that it obscures the origin of various technical problems, thereby making them harder to solve. I propose an AI methodology that does not ignore the social situation of the agent, but instead uses it to help design the agent.

In particular, "socially situated AI" may help solve schizophrenia. In schizophrenia, behaviors are too atomized, causing agents to switch abruptly between behaviors. From a social perspective, schizophrenia means the user can see the way the designer has broken the agent up. Therefore, schizophrenia may go away *if we make the breaks somewhere the user is unlikely to look.*

To be more specific, behavior-based agents are typically broken up into visible behaviors. But if users are meant to recognize these behaviors, abrupt switching between them will be obvious. Instead, internal switching should occur without changing the visible behavior of the agent. These internal switches may be less obvious, since the user sees the agent doing the same thing before and after the switch.

I am building an architecture in which "atoms" of agents are not just behaviors but also *behavior transitions*, special behaviors that change from an old high-level activity to a new one. Switches among such activities are now implemented smoothly as behaviors, instead of occuring abruptly between them. Behavior transitions need to allow programmers to manage the complexity of combining many activities while making the agent's behavior look more smooth and natural.

Potential technical problems with behavior transitions include quadratic explosion of code; large amounts of state to communicate between behaviors and the transitions taking over for them; and complicated, inter-related, undebugable code. I am developing solutions to these problems based on a socially situated approach, which integrates, not the agent's *behavior*, but the *effect* of the behavior on the user. The designer is given the capability to manage the agent's *effect*, instead of just the agent's behavior. I hope to show that a behavior-transition architecture based on these principles will allow designers to use the benefits of atomization (modularization; clean code; understandability) without the drawback of schizophrenia, thereby creating larger, more coordinated agents.

Acknowledgments

This work was supported by the Office of Naval Research under grant N00014-92-J-1298. Many thanks to my advisors, Joseph Bates and Camilla Griggers.