Nick Knouf

Week 2

As it is with many academic debates that are couched in polar opposities, the solution to the debate of "direct manipulation versus interface agents" lies in a combination of both approaches. As Pattie Maes describes in her remarks, the agents she speaks of are not designed to be anthropomorphic, are not designed to be seen by the user. Rather, only their outputs are designed to be visible. This requires, then, a direct manipulation interface to these results.

There was a point made by Ben Shneiderman that I believe might have been lost; a point that quite well describes the different usage scenarios for direct manipulation and interface agents. Ben's examples of direct manipulation come from highly structured domains--domains where there is a tabular or hierarchical structure to the data--and not from more unstructured domains, such as a user's e-mail or the collection of documents on the Web. I believe that the number of sliders and buttons needed to capture all of the complexity of unstructured data argues against the use of direct manipulation for all knowledge domains.

Ben speaks of the problems of using auto-focus cameras as a means to discredit the usage of software agents. He claims that software agents might do things that are unexpected and that it requires knowledge from the user to user them effectively. While this is true, the same could be said for interactions with humans; they require knowledge on the part of the user; as well, humans sometimes act in ways we cannot predict. Yet we can all argue that human interaction is worthwhile and is worth the effort required to understand it. This can be a slippery slope, allowing us the excuse of creating poor interfaces; however, when not taken to an extreme, this argument allows us the flexibility of creating complicated (but useful) software agents.

There were some items that I don't think were addressed by either Pattie or Ben:

Pattie speaks of software agents working through a user's preferences. This begs the question: how do you enter your preferences? What preferences are necessary for the agent to function properly? What sort of interface do you use to enter the preferences? How can one set of preferences give enough information to the agent for it to solve complicated tasks, such as giving you recommendations?

Ben continually speaks of needing predictability in the interface. On the surface, this seems like a natural request. But it is also true that humans desire serendipity, the chance encounters and discoveries that make life interesting. How can a regimented direct manipulated interface manage this? Is it possible, with sliders and such, to provide random (but pertinent) information to the user?