

# Stop it! You're Skilling Me!

by Kevin Fox

I listened to Shelley Evenson's talk with great interest. I hadn't really thought of interaction design in terms of 'skilling' versus 'de-skilling' before, and it gave me a lot to think about over the next several days.

Coming out of that tunnel though, I don't think the matter is anywhere so simple as evaluating an interface's technical or social benefit on a sliding 'skilling' scale. The weighing of ease of use and depth of utility has different ideal balancing points depending not only on the purpose of the interaction, but the portion being assessed.

The term 'skilling' implies teaching the user a skill they did not previously possess. In terms of the talk, it seems that what was termed 'skilling' was actually closer to 'giving the user greater utility.' A 'skilling' interface didn't seem to necessarily be about teaching the user how to work something new, but rather it's about giving the user a new *capability* they didn't have before. Not that the two are mutually exclusive, but if the sole metric of whether an interface was 'skilling' or 'de-skilling' was whether the user came away from the interactive experience with a greater knowledge on how to use that particular interface, then such terminology has little to do with the effectiveness of the interface for its intended task (unless the task of the interface was simply to teach the user a new interface, which is rarely the case).

Rather, if we define 'skilling' as enabling the user to accomplish things they could not accomplish before, we have a definition which seems better matched to last Tuesday's conversations.

Using this definition, an application like Apple's iMovie is a prime example of a 'skilling' application. In part through market positioning, but mostly through interface, iPhoto enables inexperienced users to import and non-linearly edit digital video to create polished movies, far beyond those capable through on-camera editing. True, the application gives them a skill they did not possess before, but that skill, as Don Normal would say, exists more as 'knowledge in the world' than it does 'knowledge in the head. The user learns a little on how to operate iMovie (and the better designed the application, the less the user has to learn in order to tap its utility), and they learn a little about non-linear editing in general, but they gain a great deal of utility that would be gone if they no longer had access to the tool.

...

I propose that the relevant perspective here isn't one of 'skilling vs de-skilling' but rather one of focusing interfaces around human core competencies.

Often interfaces that could be seen as 'de-skilling' are called such because they take either process or decision-making out of the hands of the user. Some times this is beneficial, such as in Excel, where the computer takes the load of re-computing values in cells, taking that entirely discrete responsibility off of the user's shoulders. Other times they seek to take control away from the user either 'because the system knows better' (in the case of finding optimal airfares from billions of routing possibilities) or because of bad design (when the designers don't provide an option for the sake of interface simplicity or simple oversight that a user might want to perform an unexpected task).

A balanced interface leaves the tedious or completely predictable work to the computer (or other system), while focusing the interface on the task of presenting the user with data and tools which require a human touch, whether for creative expression, decision making, or confirmation.

In short, a balanced interface acts like an assistant, taking care of the small details, while bringing the higher-level decisions to the user. This kind of delegation of tasks to the computer shouldn't be seen as de-skilling, but rather a reapportionment of energies, so that the interface fosters enhancing the skills endogenous to the human condition, rather than those of a particular tool.