Evaluating User Interface Systems Research

Dan R. Olsen Jr. (UIST 2007)

Summary by Brian Hempel

Because of their complexity, new UI systems and toolkits cannot be readily evaluated by traditional usability testing. How might one evaluate a complex software system?

Basics

STU: Situations, Tasks, Users

Each new technology is aimed at a particular set of **users** performing particular **tasks** within particular **situations**. This STU context forms the basis for considering the value of the work and should be stated explicitly.

Importance

The system must address an important problem. Users might be important by their number, social role, or great need. A task might be important by its frequency or its consequences. Learning costs matter: "In most cases at least a 100% improvement is required for someone to change tools."

Possible Claims

Problem Not Previously Solved

A new system may address an STU context not handled by any prior tools—depending on the particular STU context, this can be a compelling claim.

Generality

A tool might be able to solve problems for multiple different kinds of users in varying contexts. Generality can be demonstrated by solving three diverse tasks.

Flexibility

A tool may facilitate rapid changes to a design. This rapid flexibility may be demonstrated by changing a number of example designs using both the new tool and an existing tool.

Expressive Match

A tool may express an interface nearer to the problem being solved—for example, a color picker instead of a hex number. To demonstrate the improvement, users might be asked to find and fix a flaw.

Expressive Leverage

The tool may reduce the number of choices required to reach a desired result, for example by eliminating repetitive actions or by leveraging hardware advances to simplify the problem. Look for some insight that facilitates the leverage—the insight may appear obvious in retrospect but it's on the reviewers to prove its triviality.

Empowering New Participants

A particular population may not usually be involved in a design process. Given that their involvement might be of benefit—designers collaborating with coders, for example—then a new tool might facilitate that involvement. If prior tools addressed this problem, then usability tests are appropriate for demonstrating this claim.

Inductive Combination

The system may present a new, small set of building blocks which can be combined to solve many different tasks. There may also be an escape hatch for tasks which cannot be completed with the provided components.

Simplifying Interconnection

A system may demonstrate an architecture that reduces n^2 interconnections between widgets to only n interconnections.

Ease of Combination

The interconnection interface itself might be simplified (*e.g.*, not SOAP).

Can It Scale Up?

An interaction mechanism might work on toy examples but become fundamentally unusable on practical problems. The system should explore a larger example.

Evaluation Errors

Usability Trap

Usability testing rests on three assumptions often broken by the complexity of systems:

1. Any person can "walk up and use" the system.
2. There exists some task performable in the new and old system that is (a) simple enough to avoid confounding variability and (b) complex enough to exercise the system.
3. It is economical to hire participants (*e.g.*, 1-2 hours per session).

The three assumptions are related. With enough money you could hire participants for long enough to teach them the system thoroughly, and you could test enough different users to compensate for the different ways to perform a complex task. In practice that's rarely possible.

Fatal Flaw Fallacy

For new interaction techniques or other small scale features, it makes sense to look closely for hidden, fatal flaws. Large systems, however, will always have missing pieces and unexplored consequences because of the limitations of researcher time. Evaluation must focus on what the system can do rather than what it cannot.

Legacy Code

Although commercial systems may need to support older artifacts, it is unreasonable to expect that research systems should be backwards compatible. The goal of research is to explore the future unhindered by the past.