Testing Interfaces That Do Not Exist

In an ideal world, Developmental Psychologists working in the Interactive industry would be able to study children's use of actual interfaces when asked to evaluate new designs. In the real world, as we all know, the exact opposite holds true. We are typically asked to evaluate and help create interactions that have never been seen before (say, using speech recognition so a child can talk back to her doll), interactions for which no published research is directly relevant and for which no working prototype will exist -- until it is far too late to make any serious changes to the design.

A great many interface design blunders can be avoided with early formative research, but the problem is how to go about it when there is no product to actually test. My goal in these 15 minutes is to suggest some creative techniques for quickly mocking up interfaces without prototypes, and to make the results of that research matter in the design of the product.

Why the emphasis on research?

Dev psych is an empirical field. It emerged specifically by challenging the conventional wisdom about what children knew and how they learned, and by marshalling research data to prove it. When I was in Grad School, we used to say: In God We Trust; Everyone Else Must Have Data.

New interfaces raise a blizzard of never-before asked questions that again challenge conventional wisdom, and which need actual testing to answer.

Now I want to be clear about something: You CAN glean some answers from what I call 'parallel literatures' in many cases. Studies of social interactions and discourse patterns can be a great resource for social interface design. For littler kids, competence in a lot of basic tasks can be judged from any of the many infant and toddler tests out there – find an item on the test that's similar to your task, check the norms, and you have a data point. I've worked on products where a

basic question like 'Can seven year olds distinguish left from right?' is important. That's information you can find in available sources.

But I have found that many of the most important questions that come up cannot be answered by relying on parallel literatures because no precedent exists. You need to actually test the interaction. And this is where the challenge comes. You can be incredibly lucky, like Kathleen at FP and like I was at MS, and have research facilities at your disposal, kids ready to go, video cameras and one way glass, the whole kit and caboodle. But you still don't have anything to test. What to do? I say, borrow a trick from the Interface Research folks: Fake it. **Wizard of Oz studies description: human follows script to mimic the technology interaction as if they are the computer/toy/etc**

Now obviously, if you are going to mock up an interaction and interface, you can't re-create the whole experience. You need to create stimuli that embody the specific features you're concerned about, and that's it. This means you have to start with **your** research questions before anything else. How you define what you are testing will shape the stimuli you create. You're going to use a rough simulation of the interactions themselves to answer your questions. **So the key here is to focus on the developmental issue, not the technology.** It's not important whether a talking, interactive doll is running from a program or is only a guy in the other room with a microphone and a script. What's important is that the talking interaction you emulate in your study contains all the elements from the final interface you're concerned about, such as timing, the vocabulary you expect to be used, and so on. Here are some principles for Developmentalists doing research that I have distilled from working in industry:

1. Understand your schedule and the development process of your client.

This is the single biggest challenge you will face: How much time do you have, and where in the process are you able to make changes to the design? Usually, you have to be in motion on research right at the very beginning of the product development process.

The nature of technology manufacturing is such that the "intelligence," the interactions themselves, has to be completely finished long before the actual toys start to roll off the assembly line. This is because the interactivity is built into just one component of the toy. It is manufactured earlier and separately from the rest of the product in most cases, because it has to already be produced and ready to be built into the final product when it is assembled. This means that it has its own development path, separate from the rest of the product. Make sure you understand when your deadlines for changing the interactions are, and that the entire team understands them and agrees with them. Building a consensus about when you can make changes to the design saves everyone a lot of headaches.

2. Define your research questions in very specific terms, and avoid 'mission creep.'

I have found nobody wants to pay for research or fit it into the schedule, but everybody wants a piece of it if you actually manage to do any, sort of like the old story of the Little Red Hen. The marketers will want to find out about appeal. Your boss will have questions, the engineers will have questions. To the extent that you can, you should try to answer these divergent questions. But you have to be careful. The broader the set of questions, the more features of the final product need to be captured in the testing mock-up (or in longer testing protocols).

If you are working without technology in a Wizard of Oz situation, there is a real limit on what you can simulate and test. The man behind the curtain can only do so much.   So keep focused on YOUR questions. That's what you're paid for.

3. When it comes to actually creating your stimuli, I have two recommendations: First, think pretend.

I say think pretend because remember your focus is NOT on the technology. It's on the interaction. You're trying to fake it, to test the interaction as if it was coming from a computer or toy. Pretend play is all about "as if," about maintaining two levels of representation: knowing a prop is a prop yet treating it AS IF it was something else at the same time. Pretend this doll is your baby, pretend your sister is a lion. A puppet reading from a character's script can be a simple way to test planned dialogue because kids are willing to 'play along.' Interface mock-ups can get away with quite a lot by building on the simple duality of pretend, and you can test a lot of interactions that way.

Second, Suck Up to your engineers. When it comes to creating your actual mock-ups, your engineers are your best allies. Often they have pieces of games, art or code that can be used in your or in simple prototypes. And much of the time, they are as interested in research as you are. If you are working on screen-based content, there are now a variety of quick prototyping tools that can instantiate interface designs easily and use the early art and intended interactive logic.

If you are making a physical object that has interactive properties, then things get even more interesting! Now your mock-up has to embody certain concrete features of the design to be a valid representation of the final experience. And this is where things get truly interesting, which leads to my next point. When it comes to research prototypes…

4. Don't let the perfect be the enemy of the good.

Your prototype will differ from the final product in a variety of ways: The voice may be wrong, it may not even be the character you intend (**Blarney example from Arthur/DW testing), and the interactions may be simpler than they will be in the final product. These things are not necessarily threats to the validity of your testing. Remember, your focus is on the INTERACTION, and the specific features of the interaction you want to test. Your prototypes can vary quite a bit from the final product, as long as they embody the crucial elements you want to test.

5. Document what you did, and your specific conclusions.

Now I know this is a pain. Most people see it as a waste of time, a kind of mopping up. Bureaucratic paperwork. But trust me. Later people will dispute your findings. Or you won't remember them. Your interface expertise depends on accumulated wisdom. Take the time to accumulate your wisdom. Writing down and saving what you learned from your testing is a HUGE advantage as you go forward, both for future products with the same client and for your own professional expertise as well.

So in conclusion, I call upon my fellow Developmentalists in industry to reclaim our intellectual heritage. Become the in-house scientists that quality product development requires. Be creative in your methods, and document your results. Good developmental design is the combination of wisdom and testing. We need to be creative and quick if we are to act as forces for good in a fast-paced industry that, sadly, is all too willing to take our children's money and sell them trash.