

ITI0209: User Interfaces

21. UI Behavioral Patterns

Martin Verrev

Spring 2023

Means to an end

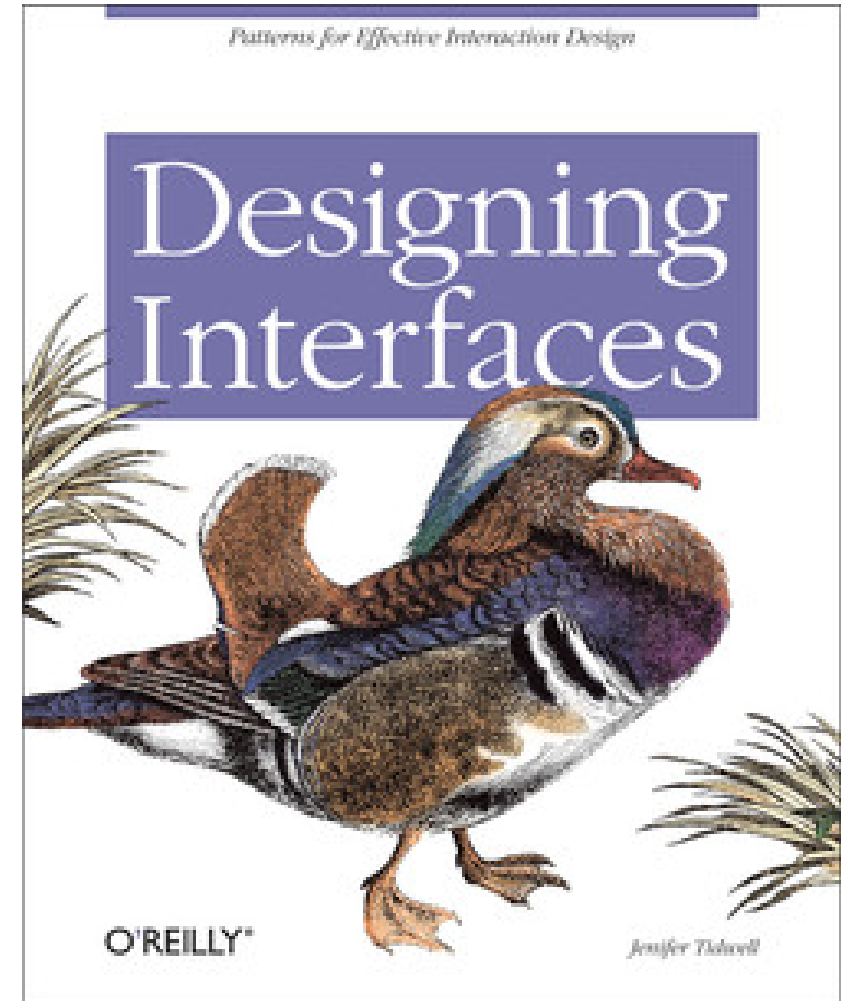
Everyone who uses a tool, software or otherwise, has a reason to use it. For instance:

- Finding some fact or object
- Learning something
- Performing a transaction
- Controlling or monitoring something
- Creating something
- Conversing with other people
- Being entertained

Designing Interfaces

Jenifer Tidwell. 2005

<https://www.oreilly.com/library/view/designing-interfaces/0596008031/>



Well-known idioms, user behaviors, and design patterns can support those abstract goals. Interaction designers have learned, for example, how to help people search through vast amounts of online information for specific facts. They've learned how to present tasks so that it's easy to walk through them. They are learning ways to support the building of documents, illustrations, and code]

Problem

- How to manage the increased complexity of the system being constructed.

Solution

- Use the patterns.

Behavioral Patterns

- Define the behaviors that stand true for most of the user interfaces.
- Are applicable to any user interface.

An interface that supports these patterns well will help users achieve their goals far more effectively than interfaces that don't support them. Those the patterns are not just about the interface, either. Sometimes the entire package, underlying architecture, feature choice, and documentation needs to be considered in light of these behaviors.

1. Safe Exploration (*Turvaline uurimine*)

"Let me explore without getting lost or getting into trouble."

Problem

How to explore the system without breaking it.

Solution

Give users option to undo their actions. When someone feels like she can explore an interface and not suffer dire consequences, she's likely to learn more and feel more positive about it than someone who doesn't explore. Good software allows people to try something unfamiliar, back out, and try something else, all without stress.

2. Instant Gratification (*Kohene Rahulolu*)

“I want to accomplish something now, not later.”

Problem

People like to see immediate results from the actions they take - it's human nature.

Solution

If you can predict the first thing a new user is likely to do, then you should design the UI to make that first thing stunningly easy. If the user's goal is to create something, for instance, then show a new canvas and put a palette next to it. If the user's goal is to accomplish some task, point the way toward a typical starting point.

3. Satisficing (*Piisav rahulolu*)

“This is good enough. don’t want to spend more time learning to do it better.”

Problem

People are willing to accept “good enough” instead of “best” if learning all the alternatives might cost time or effort.

Solution

Make the most important information simple for your user to grasp at a glance. If you’re relying on your users to read large blocks of text to understand how your product can benefit them, then you’re allowing the majority of potential users to fall through the cracks.

4. Changes in midstream (*Jooksvad muutused*)

"I changed my mind about what I was doing."

Problem

People change what they're doing in the middle of doing it. She may visit eshop to read reviews, but ends up buying a book. Maybe she is sidetracked; maybe the change is deliberate. Either way, the user's goal changes while she's using the interface.

Solution

Make choices available. Don't lock users into a choice-poor environment with no global navigation, or no connections to other pages or functionality, unless there's a good reason to do so.

5. Deferred Choices (*Edasilükatud otsused*)

“I don’t want to answer that now; just let me finish!”

Problem

This follows from people’s desire for instant gratification. If you ask a task-focused user unnecessary questions in the process, they might prefer to skip the questions and come back to them later.

Solution

Why not allow them to skip most of the question, answer the bare minimum, and come back later (if ever) to fill in the rest?

6. Incremental Construction (*Samm-sammuline ehitus*)

"Let me change this; let me change it again. That's better."

Problem

When people create things, they don't usually do it all at once. Even an expert doesn't start at the beginning, work through methodically, and come out with something perfect at the end.

Solution

Make it easy to build small pieces one at a time. Keep the interface responsive to quick changes and saves. Feedback is critical: constantly show the user what the whole thing looks and behaves like while the user works.

7. Habituation (*Harjumuslikkus*)

“That gesture works everywhere else; why doesn’t it work here, too?”

Problem

When you use an interface repeatedly, some frequent physical actions become reflexive. The user no longer needs to think consciously about these actions. They’ve become habitual.

Solution

If there is an industry-standard for interaction or UI, then it’s best to follow these conventions to be safe — redesigning existing patterns is generally more confusing than useful. Save your creativity for other aspects of the product.

8. Spatial Memory (*Ruumiline mälu*)

“I swear that button was here a minute ago. Where did it go?”

Problem

When people manipulate objects and documents, they often find them again later by remembering where they are, not what they're named.

Solution

Think about your application as though it's a physical space. People will be arranging things in a way that they see fit, not how your algorithm thinks they should be. Make it easy for your users to re-arrange things and bookmark or save items for later.

9. Prospective Memory (*Tulevikumälu*)

"I want to be reminded of information at a later point in time."

Problem

We engage in prospective memory when we plan to do something in the future, and we arrange some way of reminding ourselves to do it. For example, if you need to bring a book to work the next day, you might put it on a table beside the front door the night before

Solution

Build in support for remembering tasks. But do not overdo!

10. Streamlined Repetition (*Sujuv kordus*)

“I have to repeat this how many times?”

Problem

In many kinds of applications, users sometimes find themselves needing to perform the same operation over and over again. The easier it is for them, the better.

Solution

If your users are continuously repeating the same command or action — make a shortcut or workflow for it to make their life easier.

11. Keyboard only (*Ainult klaviatuur*)

"Please don't make me use the mouse."

Problem

Some people have physical trouble using a mouse. Others prefer not to keep switching between the mouse and keyboard because that takes time and effort. Others can't see the screen, and their assistive technologies.

Solution

For the sakes of these users, some applications are designed to be "driven" entirely via the keyboard. They're usually mouse-driven too, but there is no operation that must be done with only the mousekeyboard-only users aren't shut out of any functionality.

12. Social Proof (*Sotsiaalne kinnitus*)

“What did everyone else say about this?”

Problem

People are social. As strong as our opinions might sometimes be, we tend to be influenced by what our peers say or do.

Solution

Social proof can make your product more compelling. It doesn't necessarily need to include the user's friends, though it can. Displaying items based on popularity will make your users feel less alone in their decision making.

Conclusion

- Interface helps the user to achieve certain goals.
- Patterns help to achieve those goals more effectively.
- Those patterns are not just about the interface but entire architecture, feature choice, and documentation needs to be considered.

Viiteid

- User Interface Patterns. <http://designinginterfaces.com/patterns/>
- User Interfaces Patterns. <https://ui-patterns.com/>
- Laws of Simplicity. <http://lawsofsimplicity.com/>
- 14 Behavioral Patterns of User Interface Design.
<https://medium.com/@mariliaferreira/14-behavioral-patterns-for-user-interface-design-f08c5034ef83>

Tänan :)