

0 Preface

Interface Patterns

Interfaces can be grouped into smaller pieces that are easily recognizable.

- Forms
- Text Editors
- Graphic Editors
- Spreadsheets
- Browsers
- Calendars
- Media Players
- Information Graphics
- Immersive Games
- Web Pages
- Social Spaces
- E-Commerce Sites

Patterns in General

Patterns are essentially structural and behavioral features that improve the *habitability* of something; this could be anything from a website to a building. More specifically, patterns are:

- *Concrete, not general*: Patterns should be concrete enough to fill the space between high-level general principles and low-level "grammar" of UX design (widgets, text, graphic elements, alignment grids, etc).
- *Valid across different platforms and systems*: Ideally, each pattern captures some minor truth about how people work best in a given situation and translate that truth across mediums.
- *Products, not process*: Patterns *are* possible solutions, not ways to *find* possible solutions (heuristics).
- *Suggestions, not requirements*: Patterns are intended to be suggestions, one should adjust as the contextual and user needs change.

- *Relationships among elements, not single elements*: A text field is a text field. A text field with help text is a pattern. Likewise changes in a set of elements over time may be a pattern.
- *Customized to each design context*: Fit the pattern to your particular users and requirements.

Complete sets of patterns make up a "*pattern language*". They're considered as such because they cover the complete set of elements and their relationships in a particular design.

Other Pattern Collections

Pattern collections began with the "gold standard" titled *A Pattern Language* by Christopher Alexander.

Beginning in the 1990s, several books on Software Engineering began to emerge, chief among them *Design Patterns* by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides.

1 What Users Do

Abstract

Good interface design doesn't start with pictures. We must first understand the user, because the more we know and empathize the more we can create an effective interface for them. This chapter covers people, human behavior, and software that supports these human behaviors.

A Means to an End

Page 30