Chapter 6

Interaction Design

> *" What should be in the designer's mind at the start of an interface project? That's simple. Nothing. "*
>
> [Don Norman, The Front Desk, BBC Video, 1996. [BBC 1996, 00:21:20]]

Many of the ideas in this chapter are adapted from Alan Cooper's groundbreaking books "The Inmates are Running the Asylum" [Cooper 1999] and four editions of "About Face" [Cooper et al. 2014].

References

++ Alan Cooper, et al; *About Face: The Essentials of Interaction Design*; 4th Edition, Wiley, 02 Sept 2014. ISBN 1118766571 (com, uk) [Cooper et al. 2014]

++ Alan Cooper; *The Inmates are Running the Asylum*; Sams, 1999. ISBN 0672316498 (com, uk) [Cooper 1999]

++ Tamara Adlin and John Pruitt; *The Essential Persona Lifecycle*; Morgan Kaufmann, 2010. ISBN 0123814189 (com, uk) [Adlin and Pruitt 2010]

+ John Pruitt and Tamara Adlin; *The Persona Lifecycle*; Morgan Kaufmann, 2006. ISBN 0125662513 (com, uk) [Pruitt and Adlin 2006]

+ Kim Goodwin; *Designing for the Digital Age*; Wiley, 2009. ISBN 0470229101 (com, uk) [Goodwin 2009]

+ Steve Mulder and Ziv Yaar; *The User Is Always Right: A Practical Guide to Creating and Using Personas for the Web*; New Riders, 2006. ISBN 0321434536 (com, uk) [Mulder and Yaar 2006]

+ Ellen Isaacs and Alan Walendowski; *Designing from Both Sides of the Screen*; New Riders, Dec. 2001. ISBN 0672321513 (com, uk) [Isaacs and Walendowski 2001]

∘ Joel Spolsky; *User Interface Design For Programmers*; APress, June 2001. ISBN 1893115941 (com, uk) [Spolsky 2001]

+ Henrik Gedenryd; *How Designers Work: Making Sense of Authentic Cognitive Activities*; PhD Thesis, Lund University, 1998. [Gedenryd 1998]

Online Resources

++ Patrick Faller; *Putting Personas to Work in UX Design: What They Are and Why They're Important*; 17 Dec 2019. https://xd.adobe.com/ideas/process/user-research/putting-personas-to-work-in-ux-design/

+ Tony Ho Tran; *5 Essentials for Your User Persona Template (with Examples)*; 30 Sept 2019. https://invisionapp.com/inside-design/user-persona-template/

+ Tamara Adlin; *Personas Need a Purpose: Why You Shouldn't Try to Make Personas that Cover the Whole Company*; UX Collective, 24 Oct 2017. https://uxdesign.cc/personas-need-a-purpose-why-you-shouldnt-try-to-make-personas-that-cover-the-whole-company-8d696d0548

+ Shlomo Goltz; *A Closer Look At Personas: What They Are And How They Work*; 06 Aug 2014. https://smashingmagazine.com/2014/08/a-closer-look-at-personas-part-1/

◦ Kim Goodwin; *Getting from Research to Personas: Harnessing the Power of Data*; Nov 2002. https://cooper.com/journal/2002/11/getting_from_research_to_perso

◦ Kim Goodwin; *Perfecting Your Personas*; Jul 2001. https://cooper.com/journal/2001/08/perfecting_your_personas

+ Kim Salazar; *Why Personas Fail*; Nielsen Norman Group, 28 Jan 2018. https://nngroup.com/articles/why-personas-fail/

◦ Dan Saffer; *Persona Non Grata*; Adaptive Path, 17 Aug 2005. https://modus.medium.com/persona-non-grata-376f4757d615

Stock Photography and Images

++ Unsplash; unsplash.com

+ pixabay; pixabay.com

+ Pexels; pexels.com

◦ morgueFile; morguefile.com

◦ FreeImages; freeimages.com

◦ OpenPhoto; openphoto.net

◦ FreeImages.co.uk; freeimages.co.uk

◦ Wikipedia free image resources; https://wikipedia.org/wiki/Public_domain_image_resources

Always first assume that images are covered by copyright! Then check the exact licence terms, to see if you can use them.

Computers Versus Humans

Computers do not work like humans (see Table 6.1).

- One part of software, the inside, must clearly be written in harmony with the demands of silicon.

Computers	*Humans*
Incredibly fast	Incredibly slow
Error free	Error prone
Deterministic	Irrational
Apathetic	Emotional
Literal	Inferential
Sequential	Random
Predictable	Unpredictable
Amoral	Ethical
Stupid	Intelligent

Table 6.1: There are incredible differences between computers and the humans who have to use them. [From Cooper [1999, page 87].]

Programmer	*User*
Wants control, will accept some complexity.	Wants simplicity, will accept less control.
Wants to understand, will accept some failure.	Wants success, will accept less understanding.
Concerned with all possible cases, will accept advance preparation.	Concerned with probable cases, will accept occasional setbacks.

Table 6.2: Programmers ("homo logicus") think and behave differently from normal humans (homo sapiens). They are more in tune with the needs of computers. [From Cooper [1999, pages 97–100].]

- Equally, the other side of software, the outside, must be written in harmony with the demands of human nature.

Programmers are Different

Programmers ("homo logicus") think and behave differently from normal humans (homo sapiens) and most users. See Table 6.2.

Programmers are good at designing the inside of software, interaction designers should design the outside.

Goal-Oriented Interaction Design

Designing software based on an understanding of human goals.

What is a goal?

- A *goal* is a final purpose or aim, an objective.
- *Tasks* are particular ways of accomplishing a goal.

There may be multiple ways of achieving a goal.

Personal Goals	*Corporate Goals*
Do not look stupid.	Increase profit.
Not make mistakes.	Increase market share.
Get adequate amount of work done.	Defeat competition.
Have fun.	Hire more people.
	Go public.

Table 6.3: Personal and corporate goals are different. Both must be taken into account for software to succeed, but personal goals will always dominate. [From Cooper [1999, pages 157–158].]

Tasks are not Goals

- *Goal*: Get something to eat.
- *Task*: Go to the restaurant around the corner. Or
- *Task*: Call the pizza delivery service. Or
- *Task*: Go to the supermarket, buy ingredients, and cook for myself.

Too often, software designers focus on simplifying a task, rather than accomplishing a goal. Tasks are a means to an end, not an end in themselves.

Tasks Change with Technology

Tasks change with technology, goals do not:

- Year 2000
  - *Goal*: Get to work.
  - *Task*: Take the tram.
  - *Task*: Take a taxi.
  - *Task*: Drive in traffic.
- Year 3000
  - *Goal*: Get to work.
  - *Task*: Press the teleport button.
  - *Task*: Fly with jet pack.

Personal and Corporate Goals

Personal and corporate goals are different (See Table 6.3).

- Both are important for their respective owners (so both must be taken into account).
- However, people are using the interface, and their personal goals will always take precedence.

The Interaction Design Process

1. Interview users.
2. Create personas.
3. Define their goals.
4. Create concrete scenarios.
5. Move to a design solution.

The Design Team

Two designers in core team:

- Designer: generates ideas, leads the process.
- Design Communicator: articulates half-formed ideas, writes design spec.

6.1 Creating Personas

From the insight you gained in your interviews, you now invent user archetypes to represent the main user groups of your product.

In other words, you make up pretend users and design for them.

The Elastic User

All too often, the term "user" is bent and stretched by the programmer to adapt to the needs of the moment (see Figure 6.1):

- When a developer finds it convenient to open a file browser, the user is presumed to be an experienced, computer-literate user.
- When a developer finds it convenient to use a step-by-step wizard, the user is redefined to be an obliging first-time novice.

Never refer to "the user", instead refer to a very specific individual, a persona.

Do Not Design for the Average User

- Designing for the "average" user produces a design to please no-one, like the jumble car in Figure 6.2.
- Differentiate primary kinds of user and design for them, like the cars in Figure 6.3.

What is a Persona

A *persona* is a prototypical user:

- An imaginary, but very specific, example of a particular type of user.

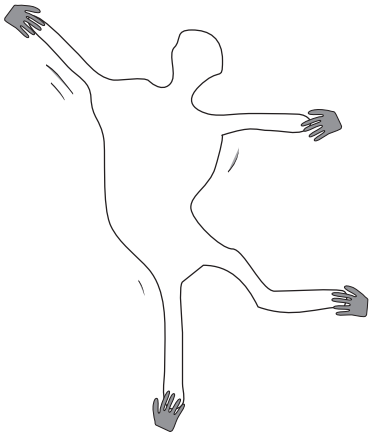


Figure 6.1: The term "user" is elastic and is liable to be bent and stretched by the programmer to the needs of the moment. [Redrawn from Cooper [1999, page 127].]

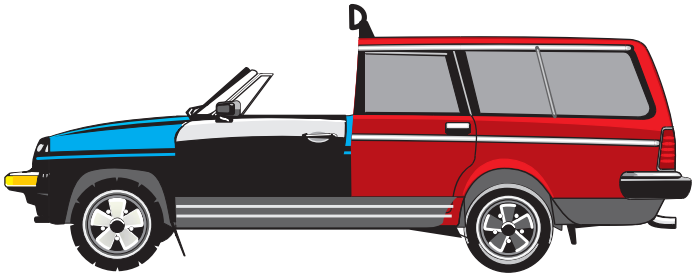


Figure 6.2: The jumble car was designed for the "average" driver.

- Not "real", but hypothetical.

A persona is used to role-play through an interface design and check if the design would meet the needs of such a user.

Define the Persona Precisely

- Specify a name, age, face, and quirky, believable detail.
- For faces, use stock photos from CD-ROM or the internet, or photographs taken during user interviews.
- It is more important to define the persona in great and specific detail, so that it cannot wiggle under the pressure of development, than that the persona be exactly the right one.

Finding Primary and Secondary Personas

- Start off with a larger set of personas.
- Combine or throw out redundant personas.
- A *primary* persona will not be satisfied with a design for someone else.
  - If there are multiple personas with radically different needs, there are multiple primaries.

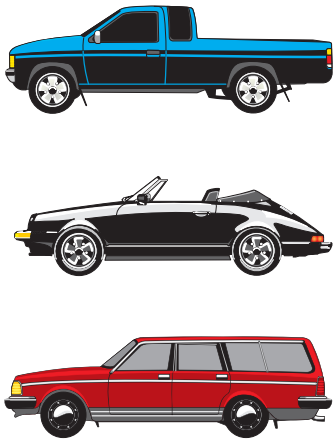


Figure 6.3: Cars are designed to appeal to different kinds of drivers with different needs and goals.

  - Each primary gets their own interface.
- A *secondary* persona is mostly satisfied with a primary's interface, but has a specific additional need.

Case Study: In-Flight Entertainment System

Fictional example based on Cooper [1999], an inflight entertainment system called InFlight for Zoom Airways.

At each seat a touch-screen video console (see Figure 6.4):

- 36 films in five categories, 36 music channels, news, childrens shows, games, shopping.
- computers + large hard disks in front of the plane.
- true video on demand – each passenger can start, pause, and rewind programmes independently.

Case Study: Two Separate Interfaces

- One for the passengers in the seat console. The four main passenger personas are shown in Table 6.4.
- A different one for employees in the attendant's station.

Case Study: Who are the Primary Passenger Personas?

The seat console interface has to satisfy Chuck, Erin, Marie, and Clevis, while at the same time not making any of them unhappy.

- Erin knows wanting to play games is something special, so she does not mind pressing a few extra buttons to get them.
- Chuck knows his vast flying experience has earned him some shortcuts, but he does not mind investing a little effort into remembering those special commands.
- Marie is similar to Chuck, and both would be annoyed by time-consuming training screens for new users.

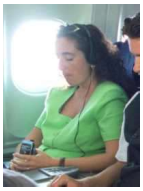
	**Name:** Clevis McQuinn **Age:** 63 **Class:** Economy Clevis was born and still lives in a small town in Texas. He only flys once or twice a year to visit his daughter who lives in Boston. Clevis might be old, but he is still spry. He is slightly embarassed about the touch of arthritis in his hands, but his mind is still very sharp. Clevis does not own a computer and does not know how to use one. He is firmly of the opinion that you can get by without one. Clevis had to start wearing glasses about 5 years ago, because his eyesight was starting to fail him.
	**Name:** Marie Dupart **Age:** 31 **Class:** Business Marie was born in France, but has been living and working in the USA for 6 years. She is bilingual, but English is her second language. Marie travels on business several times times a year. She is a self-confident young woman, who is not afraid of modern gadgetry. She owns a PDA and an iPod. Marie does much of her shopping online. She is also very interested in the latest show business gossip in the entertainment media.
	**Name:** Chuck Burgstein **Age:** 52 **Class:** First Chuck is a resident of New York who flies almost every week. He is a member of the 100,000-mile club. He has an extremely hectic lifestyle and spends more than 100 nights a year in hotel rooms. Chuck expects service here and now and has little tolerance for condescending or time-consuming activities. Chuck has strong opinions, which he is not shy to express. Even if he is usually right, other people do find his in-your-face manner somewhat irritating.
	**Name:** Erin Scott **Age:** 9 **Class:** Economy Erin lives Austin, Texas and is going to stay with her aunt and uncle in upstate New York for two weeks. She is a little bit nervous, but also excited about travelling unacompanied for the first time. Erin likes drinking fizzy orange pop. At home she will often spend hours and hours on her computer playing Sims2.

Table 6.4: The four main passenger personas.

Figure 6.4: The InFlight seat console.

- Clevis is the golden nugget, the primary persona. A menu bar or dialogue box would instantly lose Clevis. With arthritis, any complex manipulation is out of the question. An interface designed for Clevis will be acceptable to all the others, as long as their special needs are accommodated somewhere in the interface.

Case Study: Designing for Clevis

Clevis can not and will not "navigate", so there can be only one screen:

- Horizontal scrolling panoply of film posters and album covers.
- A large rotating knob (a "data wheel") physically below the screen, which can be spun like a radio wheel.
- Clevis views the posters as if strolling past, no need to even think in terms of film categories.
- Navigation bar across bottom of screen, feedback where we are and with jump scrolling for Chuck.

See Figure 6.5.

A Good Persona

- A good persona is not "average", but typical and believable.
- If the set of users interviewed were somehow plotted according to their characteristics as a cloud of points, the best ones to base personas on would be the ones around the edges. See Figures 6.6, 6.7, and 6.8.

Figure 6.5: The InFlight final design.

- If our design satisfies the hard cases around the edges, the ones in the middle should be able to use the interface as well.

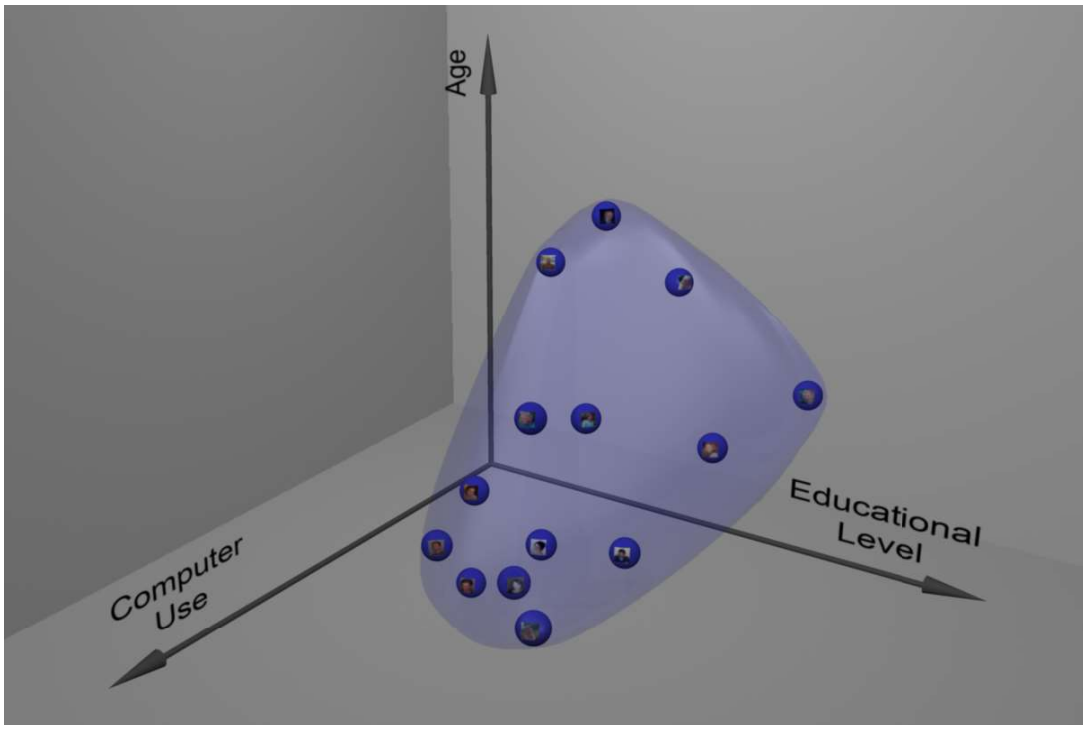


Figure 6.6: Users form a point cloud.

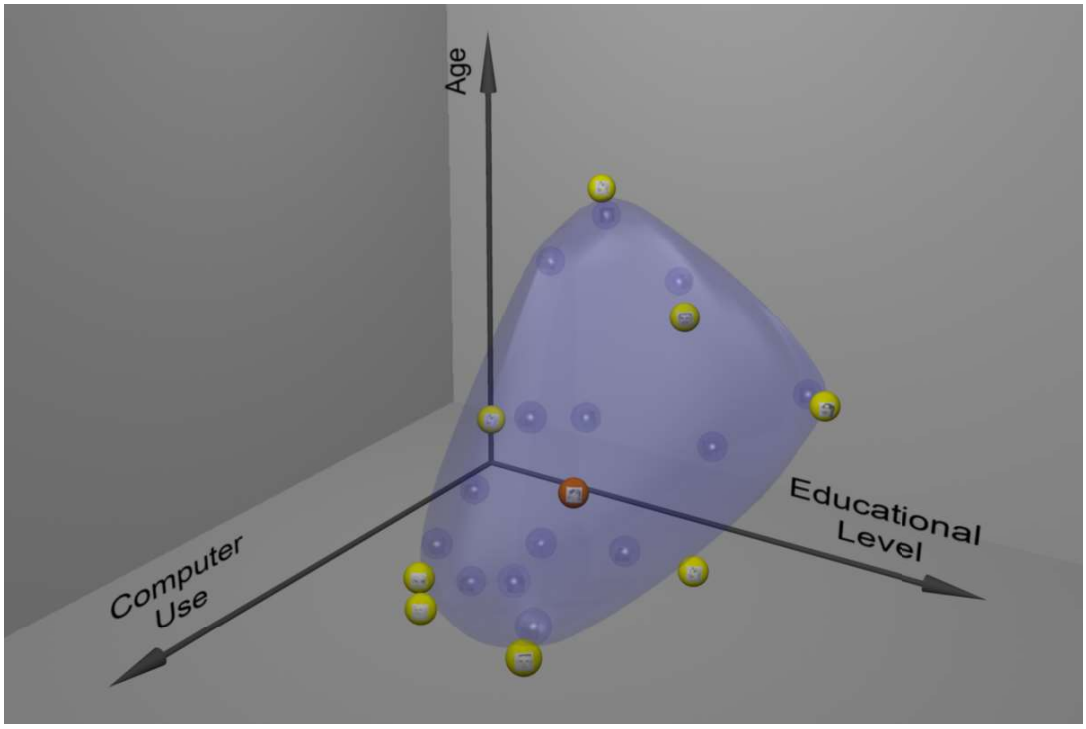


Figure 6.7: The average user and personas.

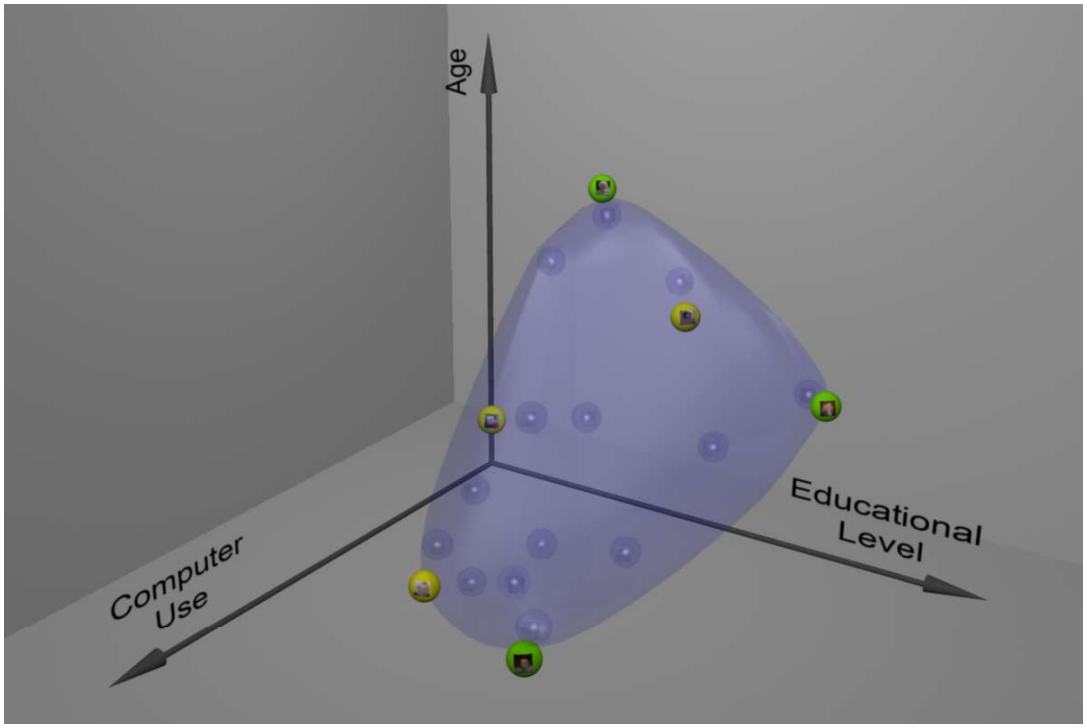


Figure 6.8: Primary personas.

6.2 Video: Illustrating the Persona Design Process

- A 3d animation to illustrate the persona design process.
- Created by Keith Andrews and Ines Legnar in 2012.
- A set of 14 blue spheres represent real users.
- They are placed within three (exemplary) dimensions, according to the characteristics age, educational level, and computer experience.
- The users form a “point cloud”.
- A persona in the middle of the point cloud would represent the "average" user and would not make a good design target.
- Good candidate personas tend to represent the edge cases of user characteristics around the edges of the point cloud.
- Personas are given photographs to make them more believable.
- Finally, personas are divided into primary and secondary personas.

6.3 Defining Goals for each Persona

Goals and personas co-exist:

- A persona exists to achieve his or her goals.

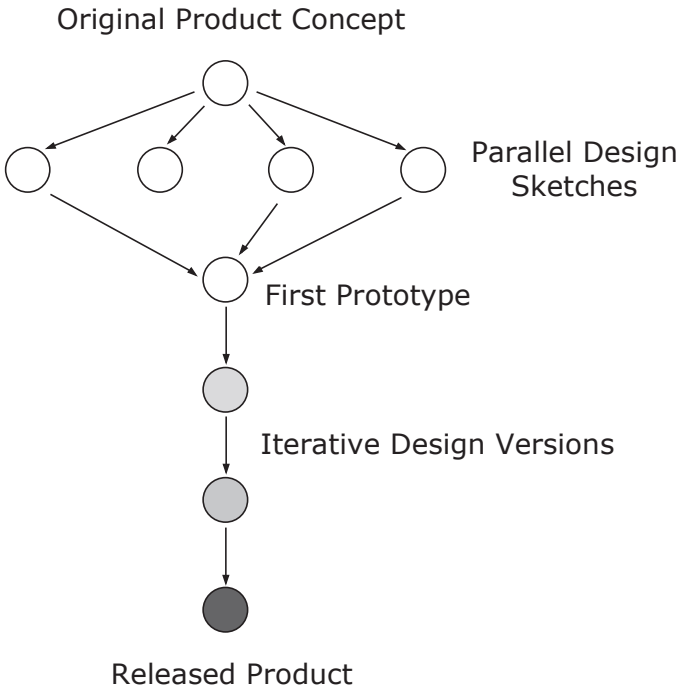


Figure 6.9: The relationship between parallel and iterative design. The first prototype is based on ideas from parallel design sketches. [Redrawn from Figure 8 of Nielsen [1993b, page 86].]

- A goal exists to give meaning to a persona.

  Define the goals of each persona.

6.4 Defining Scenarios for each Persona

A scenario is a precise description of a persona using an interface to achieve a goal:

- *Daily Use Scenarios*: The actions users perform regularly and frequently. These need the most robust design.
- *Necessary Use Scenarios*: More occasional, infrequent actions, but which are necessary from time to time.
- *Edge Case Scenarios*: Loved by programmers, these can usually be ignored during the design process.

As the design progresses, play act the personas through the scenarios to test the validity of the design.

6.5 Moving to a Design Solution

Parallel Design

- If time and resources allow, explore design alternatives.
- Have several design teams work *independently*, then compare draft designs (see Figure 6.9).

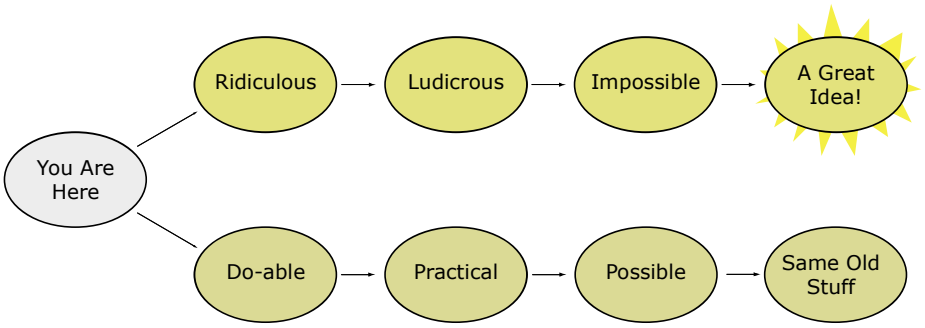


Figure 6.10: Lateral thinking. Build on the crazy to generate new ideas. [Redrawn from Cooper [1999, page 188], based on the ideas of de Bono [1970]] .

Brainstorm

- Meet away from usual workplace (different building, hut in the mountains).
- Brainstorm with mixed team (engineers, graphic designer, writer, marketing types, usability specialist, one or two representative users).
- Use plenty of paper. Cover the walls with it!
- Draw. Scribble. Use lots of coloured pens.
- Be stupid.
- Go crazy, build on the insane, think laterally (see Figure 6.10).
- Three rules during brainstorming:
  1. No one is allowed to criticise another's ideas.
  2. Programmers must not say it can't be implemented.
  3. Graphic designers must not laugh at programmers' drawings.
- Only *after* brainstorming, organise ideas and consider their practicality and viability.

Adapted from Tognazzini [1992, page 67].

Four Techniques for Getting Unstuck

- *Pretend It's Magic*: If it were magic, it would do X. If we can not do X, how close can we get?
- *Pretend It's Human*: What response would the persona expect from a human?
- *Getting Another Brain*: 15-minute rule. If a team is stuck for more than 15 minutes, ask another designer for help.
- *Renaming*: Sometimes, a lack of common vocabulary is the problem. In this case, give elements new names in a silly theme (types of cheese, mountain ranges), define what they mean, and give them real names later.

Under which pull-down menu would you most expect to find each of the word processing tasks below?

File Edit View Insert Format Utilities Macro Window

1. Search for a word: _____
2. Create a header: _____
3. Create an index entry: _____
4. Set up your document preferences: _____
5. Repeat an annotation: _____
6. Paste another file into your current file: _____
7. Use italics: _____
8. Show summary stats about your document: _____
9. Switch to another document: _____
10. Show all annotations in the document: _____
11. Repaginate the document: _____
12. Set up your printer: _____

Figure 6.11: A quick user survey on menu organisation.

Pretend It's Magic

What would these things do if they were magic?

- TV/VCR entertainment system
- Telephone/voicemail
- Calendar software
- Microsoft Windows

6.6 Getting Ideas from Your Users

Quick Surveys for Menu Organisation

A simple user survey can quickly tell you where users would expect to find certain functionality.

Figure 6.11 shows a survey to place word processing functions.

[This is, in fact, a quick and dirty version of a closed card sorting test.]

6.7 Follow Conventions

Following the standard way of doing things is generally good:

Figure 6.12: Jakob Nielsen's web site `useit.com` uses the conventional link sytle of blue underlined for unvisited links and purple underlined for visited links [Nielsen 2005].

- Users can then *transfer knowledge* as they move between applications.
- Only break the convention for good reason.

Link Styles

- Users should not have to guess where they can click (or have to scan the screen with the mouse to see what is clickable).
- For textual links, use coloured and underlined text (and do *not* underline non-link text).
- Distinguish between visited and unvisited links.
- By convention, unvisited links are blue underlined, visited links are purple underlined. This is is a good default choice. See Figure 6.12.
- If you do choose to use a different link style, at least use it *consistently* across your site. See Figure 6.13.

An Asterisk Means a Required Field

- In the context of a web form, a field marked with an asterisk (*) has come to mean a *required* field.
- Do not change this meaning arbitrarily, as Avis did on their web site in 2005, shown in Figure 6.14.
- These three fields were the only three optional fields on their site.
- Rather than place asterisks meaning *required* next to dozens of other fields (a terrible waste of asterisks), Avis' web designers thought it reasonable to redefine the meaning of an asterisk to indicate an *optional*

Figure 6.13: Non-Standard and inconsistent link styles at `orf.at`. There are three different styles for links: white, bold black, and bold blue-grey. Furthermore, bold black is used inconsistently: the first bold black text (Ausland) is not clickable, the second (Weitere Proteste. . .) is, the third (Erster Minister. . .) is not. Users can no longer predict what is a link and what is not.

field.

- Breaking the convention caused unnecessary confusion to their users.
- Avis redesigned the site to follow the conventional meaning where an asterisk indicates a *required* field, as shown in Figure 6.15.
- See Jared Spool's article [Spool 2005] and talk video [Spool 2010, 01:50–03:52]. [Video: `https://youtu.be/ucVeyVgt6Rg?t=01m50s`]

Platform Conventions

Platform conventions are the accepted way of doing things on a particular platform:

- Apple; Macintosh Human Interface Guidelines, 1993. [Apple Computer 1992]
- Apple; iOS Design Themes, 2019. `https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/`
- Microsoft Windows User Experience, Microsoft Press, September 1999. [Microsoft 1999].
- Microsoft; Windows UX Guidelines, 2018. `https://docs.microsoft.com/en-us/windows/desktop/uxguide/guidelines`
- Java Look and Feel Design Guidelines, Sun Microsystems, July 1999. [Sun Microsystems 1999]
- Material Design; 2019. `material.io`

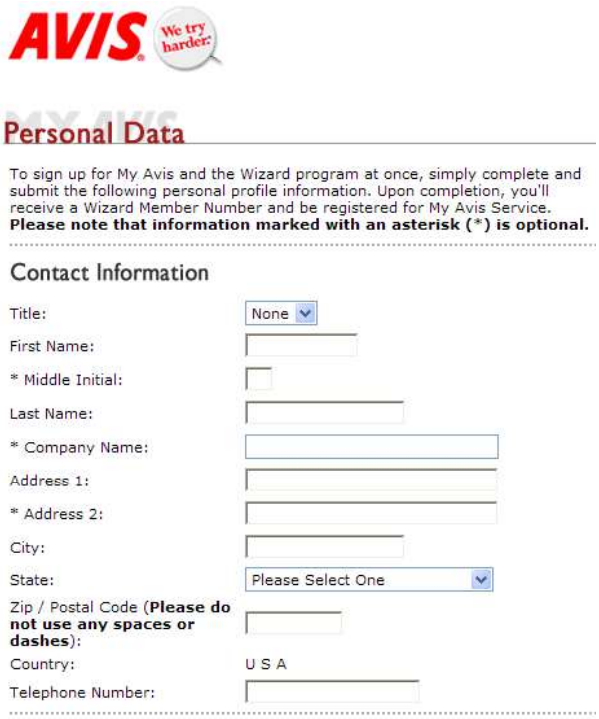


Figure 6.14: In 2005 the Avis web site used an asterisk to indicate an *optional* field, breaking the convention that an asterisk means *required*. This caused confusion to their users. [Image used with kind permission of Jared Spool [Spool 2005]]

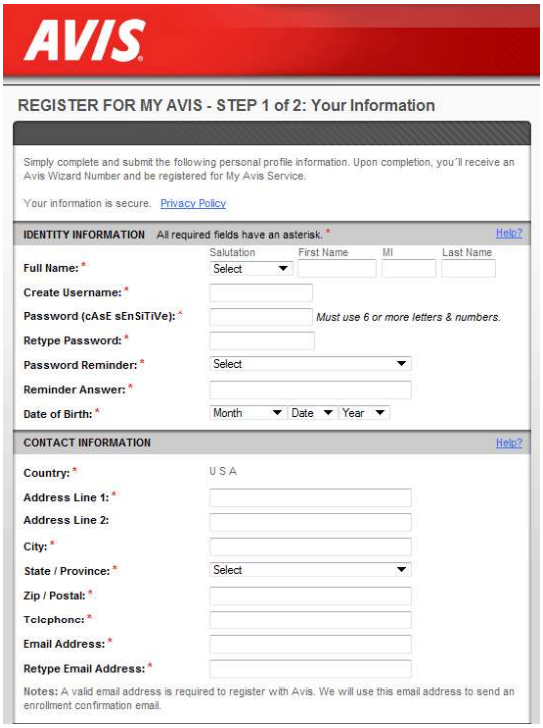


Figure 6.15: Avis redesigned the site to follow the conventional meaning where an asterisk indicates a *required* field. Here, the same form in 2011.

They are often supported by *widget libraries* of shared code.

6.8 User Interface Patterns

- *Design Pattern*: a best-practice solution to a particular design problem.
- *Anti-Pattern*: an example of bad practice.
- *Dark Pattern*: an example of a carefully crafted solution designed to mislead users.

Design Patterns

A design pattern descibes a best-practice solution to a particular design problem.

Look for *design patterns* for your particular situation: platform-independent tried-and-tested solutions to common design problems (for example, the shopping cart design pattern for an e-commerce web site).

References

+ Christopher Alexander et al; *A Pattern Language*; Oxford University Press, 1977. ISBN 0195019199 (com, uk) [Alexander et al. 1977]
+ Jenifer Tidwell; *Designing Interfaces: Patterns for Effective Interaction Design*; 2nd Edition, O'Reilly, 06 Jan 2011. ISBN 1449379702 (com, uk) [Tidwell 2011]
  ◦ van Duyne, Landay, and Hong; *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*; 2nd Edition, Prentice Hall, Dec 2006. ISBN 0131345559 (com, uk) [van Duyne et al. 2006]
  ◦ Bill Scott and Theresa Neil; *Designing Web Interfaces: Principles and Patterns for Rich Interactions*; O'Reilly, 26 Jan 2009. ISBN 0596516258 (com, uk) [Scott and Neil 2009]

Online Resources

+ Jenifer Tidwell; *Designing Interfaces*; `http://designinginterfaces.com/patterns/`
+ *UI Patterns*; `ui-patterns.com`
  ◦ *Interaction Design Pattern Library*; `http://welie.com/patterns/`
  ◦ *Pattern Tap*; `http://patterntap.com/patterntap`
  ◦ Nick Babich; *Basic Patterns For Mobile Navigation: Pros And Cons*; Smashing Magazine, 10 May 2017. `https://smashingmagazine.com/2017/05/basic-patterns-mobile-navigation/`
  ◦ Harry Brignull; *Dark Patterns*; `darkpatterns.org`

Hierarchy of Patterns (from Architecture)

Design patterns are sample design solutions based on good practice (Musterlösungen):

- Knobs
- Doors

- Walls
- Rooms
- Buildings
- Communities
- Regions

Different patterns are available at each level of abstraction.

Hierarchy of Patterns (from Web Design)

A heuristically-derived system of pluggable interface components:

- Radio Buttons
- Forms
- Search Interface
- Page Layout
- Navigation System
- Site Architecture
- Site Genre

Different patterns are available at each level of abstraction.

Video: Dark Patterns

- Harry Brignull; *Dark Patterns*; UX Brighton Conference 2010, 29-minute video [Brignull 2010, 05:22–12:09]. [Video: `https://youtu.be/1KVyFio8gw4?t=05m22s`]

Style Guides

In contrast to design patterns, style guides document the patterns (to be) used on one particular interface.

Video: Inventing on Principle

- Bret Victor; *Inventing on Principle*; 54-minute video [Victor 2012, 01:54–09:30]. [Video: `https://youtu.be/PUv66718DII?t=01m54s`]
- Bret describes a principle he calls *immediate feedback*, but which is more generally known as *direct manipulation* [Shneiderman 1983].
- That kind of editing is now known as *live coding* or *live programming*: `http://liveprogramming.github.io/liveblog/2013/01/a-history-of-live-programming/`
- There are many live coding projects inspired by Bret's talk: `http://stackoverflow.com/questions/9448215/tools-to-support-live-coding-as-in-bret-victors-inventing-on-principle-talk`

- I have not been able to locate Bret's original editor anywhere.
- Bret blogs at worrydream.com.