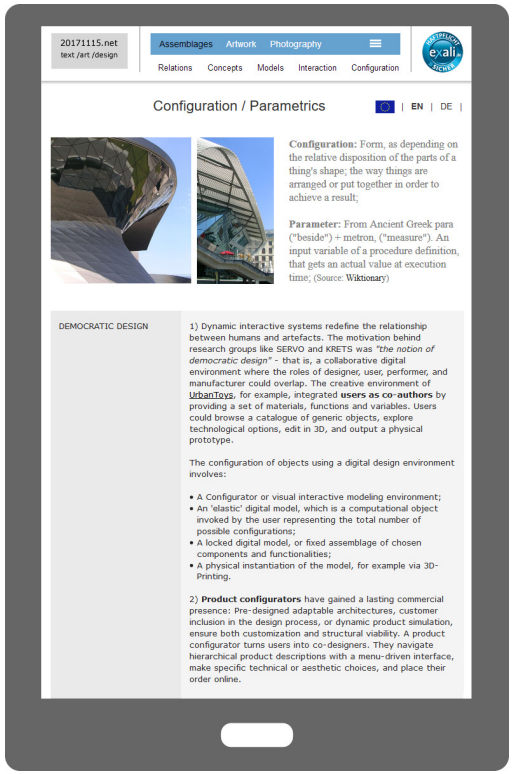
Interaction Design (2017)

Marcel Ritschel

Table of Contents

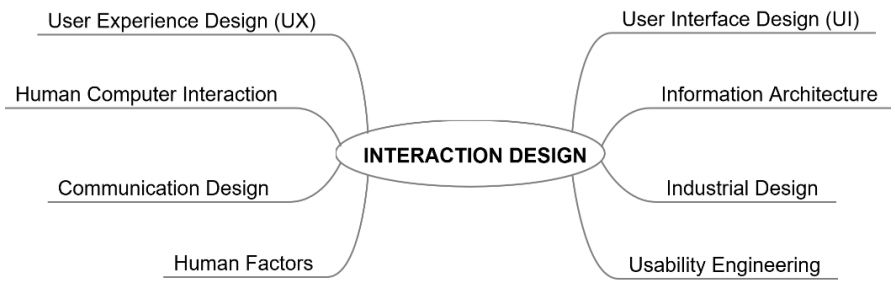


INTERACTION DESIGN

Introduction

Interaction design is a planning discipline in that it imagines what will be. If user experience is prioritised, and computing seen as a natural human activity, then understanding "the lives of the people using [the machinery]" is vital for realising human-focused interactive solutions. Bill Moggridge (2007) insists that *"a system isn't complete without the people who use it. People and their goals are the point of our systems, and we must design for them."*

The [document] examines general aspects of the design process which seem relevant to the development of interactive systems and assembles common methods for: understanding the user, specifying requirements, visualising interactions, and simulating usage through prototypes.



1 Design process

"Problem-oriented models [of designing] are essentially linear: The initial analysis stage involves consideration of the problem and its structuring into a set of objectives. Synthesis involves the generation of a range of solutions, and evaluation involves the critical appraisal of the solutions against the objectives." (Clarkson and Eckert, 2005)

1. Design research; Analysis, constraints;
2. [Re]frame the problem space;
3. Ideation and Visualisation: Create design variants;
4. Select best variant;
5. Build prototype;
6. Test and evaluate.

> The UI-design process can be delineated as a basic continuum: (a) Wireframes indicate layout and visual structure, (b) storyboards allow temporal depictions of a scenario, (c) mockups use colours and textures to instantiate a design language, and (d) interactive prototypes help simulate a user experience. A continuum allows designers not only to apply tier-based methods, but also to progress non-linearly via different versions and iterations (Ritschel, 2014).

Client & User-Centred Design: *"Expert designing involves understanding the situation into which the design will intervene [...] Designers must be as creative in their research as in their idea generation and realisation"* (UTS, 2007). When designers focus on user experience by taking a conceptual or technology-independent approach, the development of interactive systems becomes less implementation-driven and allows more stakeholders to actively participate.

2 Conceptualisation

- There is a continuum that runs from *routine design*, where a problem may be well defined, through *innovative design* where existing components and functions are recombined in novel ways, to *creative design* which occurs when requirements are yet unknown.
- Design communication may appeal to our visual-spatial memory (geometry, geography), semantic memory (general facts), or episodic memory (personal experiences).
- Idea generation and problem-solving require both creative-intuitive and rational-analytical thinking. The transmutation of mental representations into shapes and notions involves generating by communicating, and vice versa.

Concepting interactions: A problem space can be framed by describing intended changes and improvements to the current user experience. The *type of interaction* must be determined (e.g. manipulating objects, filling out a form) as well as the metaphors, analogies or idioms***** that ensure users understand your interface. A conceptualisation should consider the available interaction paradigms (GUI, multimedia, point and click, mobile/wearable, VR/AR) and address business requirements and user experience goals.

***** **(1)** Implementation-centric interfaces are based on understanding; **(2)** Metaphoric interfaces are based on analogy; **(3)** Idiomatic interfaces are based on learning. Most graphic interfaces are idiomatic.

3 Methods and techniques

"How do you optimize the users' interactions with a system, environment, or product, so that they support and extend the users' activities in effective, useful, and usable ways?" (Preece, 2007)

The interaction design process provides a range of methods in order to:

- Understand and model users;
- Understand user needs; Specify system requirements;
- Describe and visualise interactions with the system;
- Test interactive usage of the system.

Data gathering: Design research often begins with a review of existing information such as market research, competitor products, usability studies and best practices, as well as product evaluations and customer feedback. Qualitative research may involve focus groups, digital ethnography, or participatory methods. To optimise users' interactions with a system, it is important to understand both clients and users - what they want and need, what they can afford, their business requirements, corporate identity, etc. This allows designers to perceive stakeholders *"as a source of inspiration for innovation."*

Personas serve to embody significant characteristics of a user group. Though fictitious itself, a persona is based on real people, allowing designers to address real expectations, motivations and goals. Since personas represent, or even simulate, user behaviour they can be used to conceive and validate user experiences.

Scenarios: A scenario describes specific interactions between a persona and the future product or system. Scenarios help explore behaviour and representational state - that is to say, the actions of the user and the reactions of the system. *"Scenarios are prototypes built of words"* (Saffer, 2007). A key path scenario may sketch the task-space [as a responsive system] that must be 'traversed' by the user to reach a goal.

3 Methods and techniques (continued)

Task analysis: Tasks are part of a hierarchy [Activities => Tasks => Actions] and may be seen as intermediate steps on the way to a goal. Hence the analysis of current tasks, as well as the formulation of new ones, should factor in both goals and mental models of the user. The result of a task analysis is a workflow that can be integrated into the design of the new system. Workflows drive the mapping process during the creation of wireframes.

User stories allow brief, informal descriptions of software features. Any stakeholder may write a user story to promote more discussion and kick-start the implementation. The aim is to: (1) work in small steps* with many iterations, (2) be able to react flexibly to changing requirements, (3) promote closer cooperation within the team, and (4) reduce the amount of project documentation. However, because user stories contain few details, and complexity cannot be reduced indefinitely by not writing about it, implementation issues and courses of action must be regularly discussed.

Die endlose Stadt by
Friedensreich Hundertwasser.

* Agile, sprint-driven development tends to result in software architectures that eventually require some restructuring (by means of code refactoring) in order to improve maintainability and extensibility.

3 Methods and techniques (continued)

Wireframes offer a conceptual representation of the user interface, usually as a simple line drawing. A wireframe typically consists of several rectangles to represent UI-containers and basic UI-elements. Wireframes support the mapping process between interaction concepts and the envisaged user experience:
> Selection of suitable UI patterns; [see Appendix]
> i. Mapping of *functional requirements* onto appropriate tools/controls, e.g. input controls that will be needed to carry out actions; ii. Mapping of *data requirements* onto appropriate interface data objects, e.g. display areas that are necessary to perceive a machine state.

Storyboards: Visualisations of the UI by means of wireframes are more effective in conjunction with user scenarios. With storyboards, designers can explain system features and processes by combining images with narratives in a typical context of use. Storyboards also support the mapping process between the interaction concept and user experience:
> Selection of suitable UI patterns; [see Appendix]
> Selection of *navigation types* to enable user movement between (1) windows, views, pages, (2) frame application, panes, panels, or (3) tools, commands, menus.

Prototypes are built to demonstrate or test essential aspects of the new system. A prototype provides clients, developers, and target users with a low-fidelity or high-fidelity approximation of the user experience. Prototypes are not standalone artefacts, they are usually embedded in some socio-technical structure that promotes project-specific discourses and the collection of new information.

User testing: Prototype testing helps verify the feasibility of the concept or utility/usability of the product and incorporates suggestions for improvement. Users may "walk through" key functionalities of the design by means of a basic paper prototype. High-fidelity prototyping involves more advanced interactive [digital] versions. Participants normally accomplish specific tasks by interacting with the prototype and respond to questions in a structured interview.

4 Concluding remarks

"The principle of trust and control is characteristic of our interaction with a manmade technology. However, the use of media technologies requires not only a comprehensible view of a direct action (means), but also a view of the extended context (systems environment) and the consequences (diverging goals)." (Buurman, 2005)

Personal beliefs are changing. Ideologies, realities, identities are changing. Our being-in-the-world is permeated with mediating computerized technologies. If human intelligence is to be a role model for artificial intelligence, then the question of trust (or better control) will soon arise. If concepts are only developed for experts, and the highest capability is to drive the design criteria, then user resistance could be the result.
The people who deal with complex situations want help from technologies that are easy to use. They like software that is respectful, shows common sense, puts humans first, can handle responsibility, and that makes them feel cared for and in control.

Author: Marcel Ritschel
Date: Schonach, 10.10.2017

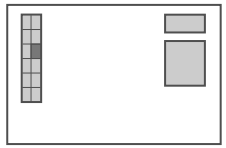
5 Bibliography

Buurman, G (ed.): 2005, Total Interaction: theory and practice of a new paradigm for the design disciplines, Basel.
Clarkson J and Eckert C [eds.]: 2005, Design Process Improvement, Springer-Verlag London Limited.
Goodwin, K: 2009, Designing for the Digital Age: How to create Human-Centered Products and Services, Wiley Publishing, Inc.
Krämer, A: Politik ist keine Mathematik. Retrieved September 2017 from: www.tagesschau.de/inland/bundestagswahl-koalition-jamaika-103.html
Moggridge, B: 2007, Designing Interactions, The MIT Press.
Nelson, E: 2002, Extreme Programming vs. Interaction Design.
Preece, J: 2007, Interaction Design: Beyond human-computer interaction, Second edition, John Wiley & Sons, Ltd.
Ritschel, M: 2014, State of the Practice, Retrieved September 2017 from: www.20171115.net
Saffer, D: 2007, Designing for Interaction: Creating Smart Applications and Clever Devices, New Riders.
Sy, D: 2006, Adapting Usability Investigations for Agile User-centered Design, in: Journal of Usability Studies.
Tidwell, J: 2006, Designing Interfaces, O'Reilly Media, Inc.

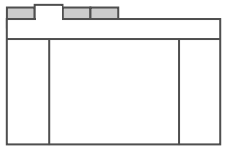
APPENDIX

Design Patterns

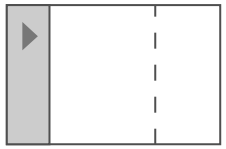
UI Patterns promote user understanding because the essential elements of an interface can be associated with a known experience.



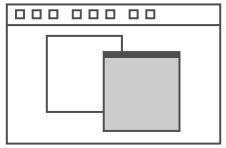
A **canvas-plus-palette** interface enables the user to select tools, colours, materials from a palette in order to create something new, or manipulate an existing object, on the digital canvas.



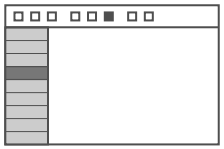
A **tabbed document** interface is able to display several documents or panes inside the one window. The tabs of the open documents remain visible, but only the active document is shown in its entirety.



Information and actions are sequenced in a **progressive disclosure**, allowing users to advance from an overview to details-on-demand, or from simple to more complex.

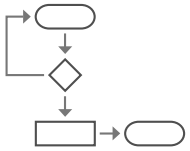


A **multiple document** interface can display several windows at once inside the one parent window. Every active child window will share the same menu-/toolbar.



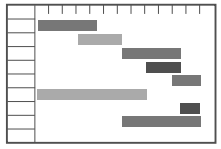
With a **two-panel selector**, list content provides an overview while the details of a selected item will be displayed in the adjacent workspace.

Data charts convert qualitative and quantitative data into visual objects so as to convey relationships or causalities at a glance.

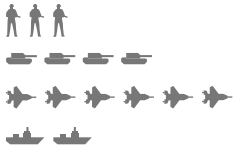


A **flow chart** can be used to visualise processes, computer algorithms, or user navigation structures. This typically involves a number of steps and decisions, shown as boxes, and connecting arrow heads to convey sequence.

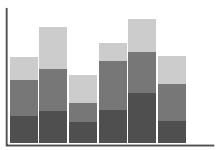
Analytics information resulting from a **conversion funnel** can be used by designers to (a) improve the customer journey, i.e. from clicking a web banner to closing the sale, and (b) to increase the conversion rate.



A **Gantt chart** is used for planning and managing projects. This chart displays tasks, durations, and dependencies against a timescale so as to permit progress tracking and target-performance comparisons.



A **pictogram** is usually a representational symbol, but in a chart it may also stand for a numerical value. Pictographic representations thus require a key so the reader can associate the symbol with physical objects.



A **stacked column chart** serves the comparison of categorical data, where each column represents a category. The various layers help indicate the composition of data within a single category.