formdiskurs

DM 19,80
öS 145,-
sfr 19,80
www.form.de

Zeitschrift für Design und Theorie
Journal of Design and Design Theory

6, I/1999

Neue Technologie – neue Konventionen?

New Technology – New Conventions?

Mit Beiträgen von *With essays by* Volker Albus, Michael Erlhoff, Alfred Hückler, Clemens Lango, Donald A. Norman, Günter Ropohl, Wolfgang Ruppert, Ralf Trachte

formdiskurs – 6, I/1999
Neue Technologie – neue Konventionen?
New Technology – New Conventions?

Inhalt

Contents

Neue Technologie – neue Konventionen?

New Technology – New Conventions?

6 **Vorwort** *Preface*
Der beschwerliche Weg zur digitalen Vollausrüstung
Volker Albus
The Difficult Path to Full Digital Equipment
Volker Albus

16 **Neue Objekte**
Differenzierungen für die universelle Datenkultur
Ralf Trachte
New Objects
Forms of Differentiation for the Universal Data Culture
Ralf Trachte

28 **Der unsichtbare Computer**
Der Trend von technologie- zu nutzerorientierten Produkten
Donald A. Norman
The Invisible Computer
The Move from Technology-Centered to Human-Centered Products
Donald A. Norman

40 **Die Virtualisierung des Buches**
Fortschritt oder Spielerei?
Günter Ropohl
The Virtualization of the Book
Progress or dilettantism?
Günter Ropohl

52 **„Wir erleben eine Leitbildrevolution"**
Ein Gespräch mit Wolfgang Glatzer und Gerd Fleischmann über das Intelligent Home
"We are experiencing a revolution in attitude"
In conversation with Wolfgang Glatzer and Gerd Fleischmann on the Intelligent Home

62 **Interaction Design – zwischen Hard- und Software**
Interaktionsstrukturen als Arbeitsfeld für Designer
Clemens Lango
Interaction Design – between Hardware and Software
Interaction strutures as a challenge to designers
Clemens Lango

Umfrage *Survey*

72 Werden kulturelle Konventionen bei der Entwicklung elektronischer Produkte ausreichend berücksichtigt?
Are cultural conventions being sufficiently considered when developing electronic products?

Auf dem Jahrmarkt
Frieder Nake
On the Fun Fair
Frieder Nake

74 **Interface im Auto und anderswo**
Siegfried Gronert
Interface in cars and elsewhere
Siegfried Gronert

78 **Computermusic – woher, wohin?**
Christian Schröder
Computer Music: Where from, where to?
Christian Schröder

82 **Computer mit Humor?**
Matthias Gommel
Computer with a Sense of Humor?
Matthias Gommel

84 **Eine Brille für Blinde**
Martin Langewitz
Glasses for the Blind
Martin Langewitz

Wiedergelesen *Re-read*

86 *Wolfgang Ruppert* kommentiert das Buch **Die Herrschaft der Mechanisierung** von *Sigfried Giedion*
Wolfgang Ruppert on ***Mechanization takes command*** *by Sigfried Giedion*

Forum *Forum*

96 **Plädoyer für eine grundlagenverbundene Lehre**
Alfred Hückler
A plea for tuition that gets back to basics
Alfred Hückler

Ein Schreiben an Alfred Hückler
Michael Erlhoff
A letter to Alfred Hückler
Michael Erlhoff

Rezensionen *Reviews*

108 *Ralf Trachte* über **Medien, Computer, Realität** von *Sybille Krämer* (Hrsg.)
Ralf Trachte on ***Medien, Computer, Realität*** *by Sybille Krämer (Ed.)*

110 *Uta Brandes* über **Die kleinen Unterschiede** von *Jan Engelmann* (Hrsg.)
Uta Brandes on ***Die kleinen Unterschiede*** *by Jan Engelmann (Ed.)*

112 *Günter Höhne* über **Die ferne Haut** von *Florian Felix Weyh*
Günter Höhne on ***Die ferne Haut*** *by Florian Felix Weyh*

114 **Hinweise für Autoren**
Information for Authors

Interaction Design – zwischen Hard- und Software
Interaction Design – between Hardware and Software

Interaktionsstrukturen als Arbeitsfeld für Designer

Autor / Author:
Clemens Lango

Über mein neues Autoradio freute ich mich. Endlich wieder Musik hören können. Naja, das Einstellen der Sender war nur mit Anleitung möglich, aber das kommt ja nicht so oft vor. Ich wählte genau dieses Modell, weil es eine Eingangsbuchse für meinen tragbaren CD-Player an der Frontseite hat. Viel gibt es bei einem Autoradio ja auch nicht falsch zu machen, was längere Vergleiche notwendig machen würde – dachte ich zumindest. Wie konnte ich nur so gutgläubig sein?

Ursprünglich ging ich davon aus, daß es sich bei einem Autoradio um ein minderkomplexes Produkt handelt. Inzwischen wurde ich gründlich eines Besseren belehrt. Bei dem Interface dieses Produktes (wohlgemerkt neuester Baureihe!) wurde so vieles falsch gemacht, daß es immerhin als optimal schlechtes Beispiel dienen kann. Daß ein Drehregler durch eine +/- Wippe ersetzt wird, ist heute häufiger zu sehen. Das ist kostengünstiger und erlaubt, die Lautstärke auf den Mikroschritt genau einzustellen. Leider wird als zusätzliches Funktionselement der Power-Taster notwendig, der sonst – gut nachvollziehbar – mit dem Lautstärkeregler kombiniert war.

Das Produkt hat in erster Linie zwei Aufgaben: Wiedergabe von Kassetten und Radioempfang. Man könnte sich natürlich fragen, warum die Taster zum Aktivieren dieser Funktionen die kleinsten sind und in der rechten oberen Ecke sitzen. Dafür sind einige Sekundärfunktionen, die nur alle paar Monate gebraucht werden, in einer großflächig in Szene gesetzten und visuell imposanten Ovalform eingebettet.

Auf dem Karton stand, daß das Produkt ein extragroßes Display habe. Das stimmt auch. Leider hat der Nutzer davon wenig, da die Leuchtstärke so gering ist, daß man am Tage kaum etwas erkennen kann. Gibt man sich Mühe, etwas zu sehen, wird man erneut überrascht. Man bekommt nicht etwa Sendername (das Radio hat RDS!) oder aktuelle Kassettenseite angezeigt. Statt dessen werden dem Benutzer lustige Lauflicht- und Buchstabenanimationen mit dem Herstellernamen angeboten. Sollte man eine der interessanteren Informationen auf das Display zaubern wollen, müßte man die Tasten und deren Funktion verstehen. Hinzu kommt, daß durch die Größe des Displays die gesamte Gerätefront heruntergeklappt werden muß, um eine andere Kassette einzulegen – perfekt organisierte Konzeptlosigkeit.

Ach ja, für den klassischen Industrial Designer gibt es auch etwas „Feines": Die aus meiner Sicht wichtigsten Taster sind nicht nur klein und in der Ecke plaziert, sondern auch in unmittelbarer Nachbarschaft des Bedienfront-Abklapptasters. Was dies bedeutet ist naheliegend. Wenn ich einmal die Front-Eingangsbuchse für die Anbindung meines portablen CD-Players nutze, komme ich nur mit Fingerakrobatik an die Lautstärketaster. Daß diese Ärgernisse bei einem Autoradio zu einer realen Gefahr

Clemens Lango
ist Diplom-Designer und zur Zeit an der Universität Wuppertal im Fach Computational Design als wissenschaftlicher Mitarbeiter tätig. Er promoviert bei Prof. Dr. Mihai Nadin und entwickelt dabei ein Softwaresystem zur Arbeit in großen Wissensräumen („knowledgespace") wie beispielsweise dem Internet. Darüber hinaus berät er Unternehmen bei der Entwicklung von Produkt- und Software-Interfaces sowie Internet-Konzepten. Seine Arbeiten erhielten zahlreiche Preise und Auszeichnungen.
Kontakt:
lango@interactiondesign.de

Interaction structures as a challenge to designers

Clemens Lango
graduated in design and is currently lecturing in Computational Design at the University of Wuppertal. He is working on his doctoral thesis under Prof. Dr. Mihai Nadin. His thesis involves the development of a software system for work in large "knowledge spaces", for example, the Internet. Furthermore, he advises companies on the development of product and software interfaces and Internet concepts. He has received numerous prizes and commendations for his work.
Contact:
lango@interactiondesign.de

I was really looking forward to my new car radio, to being able to listen to music again, at last. As it turned out, programming the desired stations was only possible with the help of the instruction booklet, but you don't have to do that too often. I had chosen this particular model because it had a socket at the front for my portable CD-player. I hadn't needed to compare different models, as there's not much you can do wrong with a car radio. At least that's what I thought. How could I have been so naive!

Originally, I had assumed a car radio to be a none too complex product. Meantime I am somewhat the wiser. There is so much wrong with the interface of this product (the very latest model, by the way!), that, at most, it can serve ideally as an example of bad design. That the regulator knob has been replaced by a +/- rocker is common these days. It's cheaper and allows you to set the volume exactly as you want it. Unfortunately that makes an additional function element necessary, the power button, which is otherwise, understandably, combined with the volume regulator.

This product has two main tasks: to play cassettes and receive radio stations. So one could legitimately ask why the buttons to activate these functions are the smallest and have been positioned in the upper right corner. By contrast, several secondary functions which you only need every few months are embedded in a prominent and visually striking oval form.

It said on the box that the product had an extra large display, which is true. Unfortunately, the user benefits little from this, as the luminosity is so faint you can scarcely recognise anything in daylight. Should you make an effort to see something, then there's another surprise in store. What you see are not station names (the radio has RDS!) or which side of the cassette is playing. Instead, the user is presented with amusing moving light and letter animations based on the manufacturer's name. Should you want to get one of the more interesting pieces of information onto the display, you have to understand the buttons and their function. What is more, due to the size of the display, you have to fold down the whole front of the radio to insert a cassette. A perfect example of a total lack of concept.

Oh, and there's a "nice" feature for the classical industrial designer: the most important buttons, in my view, are not only small and in the corner, but also directly beside the fold down button on the control panel. Need I go into detail about what this means? If ever I use the front socket for my portable CD-player, I will only be able reach the volume knob with the help of finger acrobatics. It's quite clear that these annoyances in a car radio could constitute a real danger. I wouldn't

werden können, ist klar. Ich möchte nicht wissen, wie viele Unfälle durch dieses Produkt verursacht werden.

Technologische Entwicklung und ihre Konsequenzen für Interfaces

Zu Zeiten der *elektrisch-mechanischen* Produkte hatte das Thema Interface noch nicht seine heutige Bedeutung. Aufgrund begrenzter technischer Möglichkeiten gab es meist wenige, sehr produktspezifische Bedienelemente. Bedienbarkeit erforderte ein Verstehen des mechanischen Systems und seiner Komponenten, die meist in nachvollziehbaren (weil sichtbaren) Bezügen zueinander standen. Die Produkte waren durch strikte Begrenzung von Möglichkeiten „leicht bedienbar", jedoch nicht komfortabel.

Diese Situation verkehrte sich in den 80er Jahren ins Gegenteil. Die *Elektronik* ermöglichte neue Funktionen. Daraus ergab sich bekanntlich die vom Marketing gesteuerte vordergründige Tendenz, Funktionen mit Leistung gleichzusetzen und nach der Formel „mehr Funktionen = mehr Leistung = höherer Preis bzw. Differenzierung von der Konkurrenz" zu handeln. Die Vielzahl der Funktionen wurde plakativ sichtbar gemacht und als „Komfort" angepriesen. Daß derartige Geräte aufgrund ihrer meist schlecht strukturierten Interfaces nicht bedienbar sind, weiß inzwischen jeder aus eigener Erfahrung. Schlechte Interfaces stellen eine erhöhte kognitive Last dar und verhindern ein Bedienen „nebenbei". Prinzipiell steht ein Interface zwar (technisch bedingt) zwischen Mensch und Maschine, doch sollte es die Aufmerksamkeit, die es von der Zielvorstellung abzieht, minimieren.

Die technische Miniaturisierung ist weit fortgeschritten, Hardware-Produkte aus den Feldern der Informationstechnologie und Unterhaltungselektronik haben meist keine technisch bedingte, typische Baustruktur mehr, auf die sich der Designer berufen könnte. Das gilt für Gehäuse und Interfaces. Reine Software-Produkte sind von Grund auf „frei formbar". Displays und Prozessoren führten zu einer Generation von *Software-Interfaces.* Der Interface-Bereich von Produkten konzentrierte sich auf zentrale Displays. Unstrukturierte Mehrfachbelegungen von Softkeys[1] und eine Flut von chaotischen Feedback-Informationen, die oftmals nicht in räumlichem Bezug zu den auslösenden Bedienelementen standen, führten zu neuen Interface-Problemen.

Zukünftige Aufgaben

Die Industriezweige Informationstechnologie, Telekommunikation und Unterhaltungselektronik befinden sich in einer strukturellen Umbruchphase. In den Schnittmengen entstehen neue Märkte und neue Produktklassen. Durch die Entwicklung der *Informationstechnologie* steigen die funktionalen Möglichkeiten von Hard- und Software-Produkten. Diese werden sich jedoch nur durchsetzen können, wenn die kognitiven Fähigkeiten und Erwartungen des Menschen von Anfang an in die Entwicklung einbezogen werden. Bedienungsformen dürfen sich nicht nur an der technischen Machbarkeit orientieren. Der Umweg, der innerhalb der minderkomplexen Produktwelt der Unterhaltungselektronik über rein funktionsorientierte, den Menschen außer acht lassende Interfaces beschritten wurde, würde bei diesen neuen, hochkomplexen Produktklassen zu größeren Problemen führen.

Es wird bei Produkten des täglichen Gebrauches immer mehr im Hintergrund arbeitende Prozessoren und im Vordergrund sichtbare Interfaces geben. Diese Interfaces den Bedürfnissen der Nutzer verschiedener Erfahrungs- und Altersklassen entsprechend zu gestalten, ist ein großes Aufgabenfeld. Die Dimension der Herausforderung und das sich bereits abzeichnende Marktvolumen dürfte ohnehin klar sein, handelt es sich bei den dargestellten Industriezweigen doch um entscheidende Wachstumsmärkte der Zukunft. Erfreulicherweise ist seit einigen Jahren bei manchen Firmen ein Umdenken zu beobachten. Langsam entwickelt sich auch in Deutschland ein Bewußtsein für die Bedeutung von leicht verständlichen Bedienstrukturen. Teilweise wird dies schon jetzt als Differenzierungsmöglichkeit im immer härteren Wettbewerb gesehen.

1 Unter Softkeys versteht man Hardware-Taster, deren Funktionsbezeichnung im angegliederten Display abzulesen ist.

1 Softkeys are hardware buttons whose functions can be read in the display.

like to know how many accidents are caused by this product.

Technological Development and the Consequences for Interfaces

In the era of electro-mechanical *products, interface was not as important a topic as it is today. Given the limited technical possibilities, there were usually fewer operating elements and these were very specific to the product. Usability demanded some grasp of the mechanical system and its components, and these were usually clearly (because visibly) linked. Thanks to the strict limitation of possibilities, these products were "easy to operate", but they were not convenient.*

This situation was fully reversed in the 1980s. Electronics *facilitated new functions. As we know, that gave rise to the superficial trend, promoted by marketing people, to equate functions with performance, and to act according to the formula "more functions = better performance = higher price or differentiation from the competitor". The wealth of functions was made eye-catchingly visible and lauded as "convenient". Meantime, we all know from experience that such appliances are not easy to use because of their often poorly structured interfaces. Bad interfaces represent a significant cognitive load for the user and prevent the appliance being operated "by the way". By definition, the interface stands between man and machine (for technical reasons), but it ought to minimise the amount of attention it diverts from the main objective.*

Great advances have been made as regards technical miniaturisation. Hardware products from the information technology and entertainment electronics sectors no longer have typical, technically-determined structural forms to which the designer might refer. This applies to both casings and interfaces. Pure software products are basically "freely formable". Displays and processors have led to a generation of software interfaces. *Product interface has concentrated on central displays. New interface problems are now being posed by unstructured multi-functional softkeys[1] and a flood of chaotic feedback information which is often in no spatial relationship to the operating elements that gave rise to it.*

Future Tasks

The industrial sectors of information technology, telecommunications and entertainment electronics are in a state of structural upheaval and new markets and product classes are emerging at the intersections. Thanks to developments in information technology, *the functional possibilities of hardware and software products are increasing. Yet these will only be successful if their development takes into account the user's cognitive capacities and expectations from the very beginning. Modes of operation should not be oriented exclusively around what is technically feasible. With these new and highly complex types of products even greater problems will be encountered if the same diversion is taken as has already been taken with the less complex products of the entertainment electronics sector, i.e., by way of purely function-oriented interfaces that ignore the user.*

Everyday utility products will have more and more processors working in the background and interfaces visible in the foreground. One major task will be to design these interfaces in accordance with the needs of users with different experiences and of different ages. The size of the challenge and the anticipated market volume should be evident; the branches of industry in question are decisive future growth markets. Fortunately, there has been some evidence over the past years of companies revising their thinking on this issue. In Germany too, there is a growing awareness of the importance of easy to grasp operating structures. This is even being seen as one possibility of differentiating products in the face of harsher competition.

Who is responsible?

Who is responsible for solutions to these future tasks? Industrial designers, commu-

Wer ist zuständig?
Wer löst diese zukünftigen Aufgaben? Industrial-Designer, Kommunikations-Designer, Software-Ergonomen, Informatiker, Kognitionswissenschaftler? An komplexen oder umfangreichen Projekten arbeiten heute üblicherweise Teams, die aus Personen mit verschiedener fachlicher Ausrichtung bestehen. Es lassen sich folglich nur Aufgabenschwerpunkte umreißen, denen jeweils ungefähre Qualifikationsprofile zugeordnet werden können. Ein Basisverständnis aller oben genannte Disziplinen ist sicherlich Grundvoraussetzung.

Eine typische Frage in meinen Seminaren bezieht sich darauf, ob es sich bei der Entwicklung von visuellen Interaktionsstrukturen nicht um eine Aufgabe für Kommunikations-Designer handelt. Die Benutzeroberfläche eines Anrufbeantworters wird in der Regel ein Industrial-Designer, ein Interaction-Designer oder ein Software-Ergonom konzipieren, auf jeden Fall ein Experte in den Bereichen Funktion und Prozeß. Wenn der Anrufbeantworter aber nicht als gegenständliches Produkt, sondern als Computer-Software für den Betrieb über ein Computer-Modem entwickelt werden soll, wird es dann durch die Tatsache, daß es sich um eine grafische Umgebung handelt, zu einer Aufgabe für Kommunikations-Designer? Natürlich nicht. Entscheidend ist nicht, ob es sich um ein zwei- oder dreidimensionales Produkt handelt, sondern ob Informationen oder Funktionen zu gestalten sind.

Informations- und Funktions-Software
Nach meiner Definition gibt es Informations- und Funktions-Software. *Informations-Software* ist inhaltsorientiert und reaktiv. Sie dient vorwiegend der Vermittlung von Informationen oder der Unterhaltung. Dabei könnte es sich beispielsweise um einfache Informations-Sites im Internet oder Offline-Produktionen auf CD-ROM handeln. Neben der inhaltlichen Konzeption gehört zu diesem Aufgabenfeld die Gliederung und Gestaltung von Informationen und der Aufbau einfacher Navigationsstrukturen.

Funktions-Software hingegen ist produktiv. Sie hat die Eigenschaften eines offenen Systems. Funktions-Software ermöglicht dem Benutzer, etwas zu verarbeiten, einzugeben oder zu generieren, was vorher nicht festgelegt war. Sie stellt ein Werkzeug dar, das über Funktionen eine Leistung bereitstellt. Als Beispiele für dieses Aufgabenfeld kann man ein Videorecorder-Programmiersystem, ein Anrufbeantworter-Interface, eine Bildbearbeitungs-Software oder Gliederungs- und Navigationsstrukturen für komplexe Informationssyteme oder Internet-Sites nennen.

Die Disziplin, die die Entwicklung von Funktions-Software zur Aufgabe hat, wurde erstmals durch den Begriff *Interaction Design* definiert. Interaction Design setzt auf der strukturellen Ebene von Interfaces an. Es thematisiert die Tätigkeit des Interagierens und umfaßt folglich neben visuellen Interfaces ebenso akustische, kinästhetische und VR-Schnittstellen. *Interface Design* und *Screen Design* beschäftigen sich mit der gestalterisch-visuellen Ausprägung der im Interaction Design entwickelten Strukturen. Dabei stehen teils subjektivere Kriterien wie Ästhetik oder Stil im Vordergrund. Interaction Design muß sich u.a. am objektiv verifizierbaren Kriterium Bedienbarkeit („usability") messen lassen. Die Qualität der entwickelten Strukturen bewirkt entweder Sicherheit und Spaß bei der Bedienung oder – falls sie schlecht gelöst sind – ständige Fehler und Ärgernisse.

Kompetenz
Als Eckpunkte für die Kompetenz eines Interaction-Designers sehe ich die Fähigkeit zu analytischer, abstrakter Denkweise, Verständnis der kognitiven Fähigkeiten und Grenzen des Menschen (Kognitionspsychologie und -wissenschaften), Erfahrung mit der Konzeption komplexer Systeme, Funktionsstrukturen und Schnittstellen (Industrial Design) und Gespür für visuelle Gestaltung, Typographie, Inhalt (Kommunikations-Design). Auf der manuellen Ebene ist die Beherrschung von Grafik-, Sound- und Autoren-Softwares und einer Programmiersprache für den Entwurf von Interfaces und die Erstellung von interaktiven Prototypen notwendig. Ein Grundverständnis für Algo-

nications designers, software ergonomists, computer scientists, cognitive scientists? Today, complex or extensive projects are usually worked on by teams of people who are specialised in different areas. Thus, task priorities can only be named to which approximate qualification profiles are assignable. Of course, one prerequisite is a basic knowledge of all the above mentioned disciplines.

A typical question raised in my seminars is whether the development of visual interaction structures is not the task of communications designers. As a rule, the control panel of an answerphone is developed by an industrial designer, an interaction designer or a software ergonomist, in any case, by an expert in the function and process sectors. If, however, the answerphone is to be developed not as an individual product, but as computer software with operation via a computer modem, does it then become a task for a communications designer, because that it has to do with a graphics environment? Of course not. The decisive point is not whether it is a two or three dimensional product, but whether information or functions have to be designed.

Information and Function Software
I distinguish between information software and function software. Information software *is geared to content and is reactive. Its main purpose is to mediate information or to entertain. For example, it can involve simple information sites in the Internet or offline productions on CD-ROM. Apart from conceiving the content, this area entails the subdivision and shaping of information and the setting up of simple navigation structures.*

Function software, *by contrast, is productive and has all the characteristics of an open system. Function software enables the user to process, enter, or generate something which was not fixed in advance. It constitutes a tool that makes an operation available via functions. Examples of this would be a video recorder programming system, the interface for an answerphone, software for image processing, or grouping and navigation structures for complex information systems or Internet sites. The discipline aimed at developing function software was initially called* interaction design. *Interaction design begins at the structural level of interfaces. It focuses on interaction and thus includes, along with visual interfaces, acoustic, kinaesthetic and VR interfaces.* Interface Design *and* Screen Design *deal with the visual form of the structures developed in interaction design. Sometimes more subjective criteria are to the fore, such as aesthetics or style. Interaction design has to prove itself, among other things, using the objectively verifiable criterion of "usability". The quality of the structures it develops gives rise either to confidence and fun when operating a product or, if the solution is poor, to constant mistakes and irritation.*

Competence
The competence of an interaction designer is based on a capacity for analytical abstract thought, a grasp of man's cognitive powers and limitations (cognitive psychology and sciences), experience with conceiving complex systems, function structures and interfaces (industrial design) and a feeling for visual design, typography, and content (communications design). Also necessary, at the manual level, is a mastery of graphics, sound and author software, plus a programming language for designing interfaces and producing interactive prototypes. Another prerequisite is a basic knowledge of algorithm and data interfaces. An interaction designer should have an overview of the technical possibilities and limitations.

Teaching Practice
In my seminars on Computational Design, I try to provide students with an understanding of the importance of function software. While dealing with set tasks, they implicitly gain insight into the basic principles. This is complemented by infor-

rithmen und Datenschnittstellen sollte ebenfalls vorausgesetzt werden können. Technische Möglichkeiten und Grenzen müssen überblickt werden können.

Lehre

In meinen Seminaren im Fach Computational Design versuche ich, bei den Studierenden ein Verständnis für die Bedeutung von Funktions-Software zu schaffen. Erkenntnisse zu wesentlichen Prinzipien entstehen implizit durch die Bearbeitung von Aufgaben und werden durch ausgegebene Informationen ergänzt. Der Ablauf der Veranstaltungen ist wie bei einem Real-Projekt strukturiert. Dadurch wird automatisch die Systematik eines Entwicklungsprozesses im Team geübt.

Bei der Analyse wird im Team per Brainstorming der funktionale Zweck des Produktes ermittelt. Dabei wird abstrakt vom Nutzen ausgegangen und nicht von konkreten Bedienelementen oder Detailfunktionen. Dadurch wird ein größeres Spektrum an möglichen Strukturen aufgespannt und verhindert, daß ungewollt von bekannten Ansätzen kopiert wird. Die Funktionen werden grob in Primär-, Sekundär- und Komfortfunktionen aufgeteilt, hierarchisch sortiert und in Gruppen zusammengefaßt. Bezüge zwischen den Gruppen werden – ähnlich wie in einer Mind Map – visualisiert.

Danach beginnen die Studierenden in Einzelarbeit strukturelle Interfaces zu visualisieren. Statt der in der Industrie üblichen Paper Prototypes – das sind aus haftenden Papierelementen schnell zusammenklebbare und modifizierbare Collagen – arbeiten sie inzwischen mit dem *low-fidelity prototyper*[2] (siehe Abb. 1), einer kleinen von mir entwickelten Software. Alle Teilnehmer nutzen in dieser Umgebung einen Baukasten mit Standardelementen, wodurch die subjektive Varianz eingegrenzt und Vergleichbarkeit gewährleistet wird. Die Software eignet sich besonders zum Erstellen der in dieser Entwicklungsphase üblichen low-fidelity prototypes, weil damit ohne spezielle Software-Kenntnisse visuelle Strukturen und Bedienungssequenzen sehr einfach und schnell visualisiert werden können. Die erarbeiteten Layouts und Sequenzen werden gespeichert.

Nach Abschluß der ersten Entwurfphase folgt die Besprechung. Jeder Studierende vertritt den eigenen Entwurf, ein Kommilitone führt mehrere vorher festgelegte Handlungsschritte durch, die übrige Gruppe beobachtet ihn dabei. Anschließend wird die Sequenz von allen besprochen und kritisch reflektiert. In der Regel werden durch

2 Der low-fidelity prototyper kann im Internet unter http://interaction design.de kostenlos heruntergeladen werden.

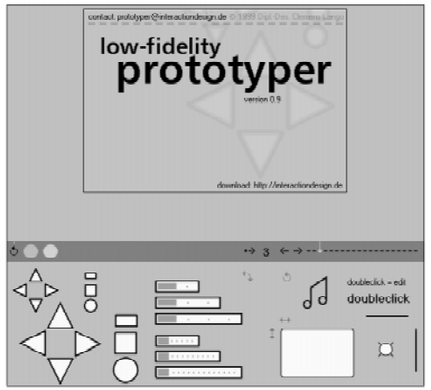
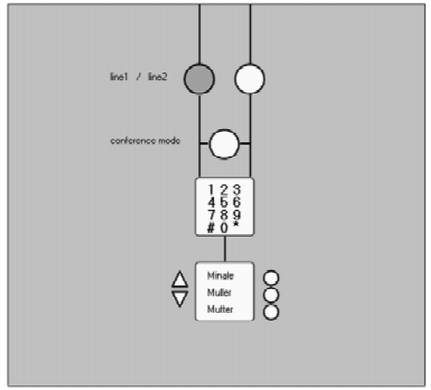


Abb. 1: Die Interface-Layout-Software „low-fidelity prototyper", Entwicklung: Clemens Lango.
Ill. 1: The Interface-Layout Software "low-fidelity prototyper", developed by Clemens Lango.



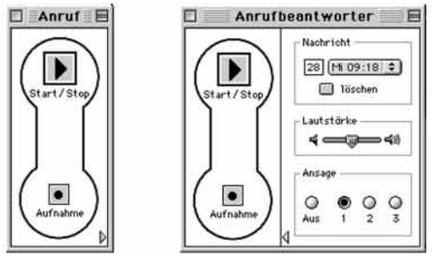
Beispiele für Interaktionsstrukturen, die mit dem „low-fidelity prototyper" entwickelt wurden (Abb. 2–4):
Abb. 2: Komfort-Telefon, Konzeption: Mario Minale.
Examples of interactive structures produced with the "low-fidelity prototyper" (Ill. 2–4):
Ill. 2: Luxury telefon, conception: Mario Minale.

2 *The low-fidelity prototyper can be downloaded free of charge from the Internet under http://interactiondesign.de*

mation handouts. The classes are structured like a real project, so that the systematics of a development process are practised automatically in a team.

At the analysis stage, the functional objective of a product is analysed by the team in a brain-storming session, focusing abstractly on the use of the product and not on concrete operating elements or functional details. This opens up a larger spectrum of possible structures and avoids any unintentional copying of familiar approaches. The functions are subdivided roughly into primary, secondary and convenience functions, sorted hierarchically and grouped. Links between the groups are visualised, like in a mind map.

Then each student begins to visualise a structural interface. Instead of the paper prototypes used in industry – collages made of adhesive paper elements that can be quickly modified and remodelled – the students work with the low-fidelity prototyper[2] *(see ill. 1), a software I myself developed. In this environment they all use a construction set made up of standard elements, thus keeping subjective variation to a minimum and ensuring that the interface designs can be compared. This software is particularly suited to producing the low fidelity prototypes typical of*



Gestalterische Ausarbeitung der Strukturentwürfe (Abb. 5, 6):
Abb. 5: Anrufbeantworter-Software gemäß dem Apple-Macintosh-Style-Guide, Design: Man Choi.
Shaping structure designs (Ill. 5, 6):
Ill. 5: Answerphone software in keeping with Apple-Macintosh style guide, design: Man Choi.

Abb. 6: Produktentwurf für einen Anrufbeantworter, Design: Thorsten Ulbrich.
Ill. 6: Product design answerphone, design: Thorsten Ulbrich.

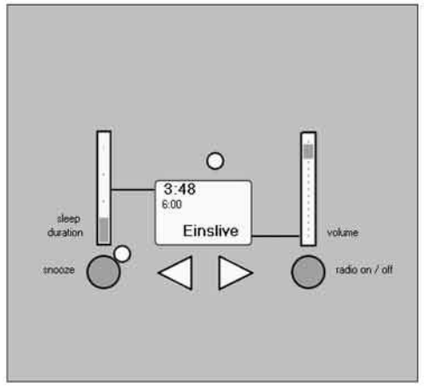


Abb. 3: Radiowecker, Konzeption: Mario Minale.
Ill. 3: Radio alarm-clock, conception: Mario Minale.

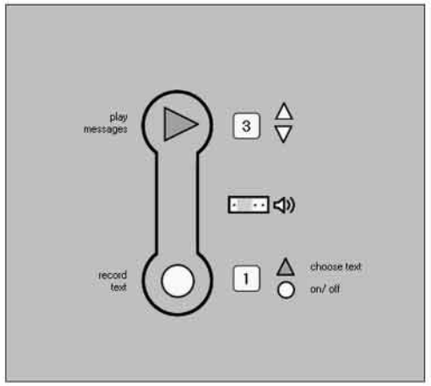


Abb. 4: Anrufbeantworter, Konzeption: Man Choi.
Ill. 4: Answerphone, conception: Man Choi.

diese aufgabenorientierte Herangehensweise typische Fehler bewußt. Außerdem können grundlegende Prinzipien von Interface-Strukturen anhand der konkreten Fälle erläutert werden. Nach der ersten Besprechungsrunde haben alle Studierenden meist mehrere Kritikpunkte und Verbesserungsvorschläge zu ihrem Entwurf erhalten und erstellen eine überarbeitete Version (siehe Abb. 2–4). Dieser Entwurfs- und Besprechungszyklus wird in der Regel zwei- bis dreimal durchlaufen, was zu einer, in der Software-Entwicklung notwendigen, iterativen Optimierung der Entwürfe führt.

Abhängig vom Umfang des Projektes wird der Strukturentwurf nach Abschluß in verschiedenen gestalterischen Varianten ausgearbeitet (siehe Abb. 5, 6). Prinzipiell sind die Strukturen anwendbar auf Hardware- und Software-Interfaces. Bei den Software-Interfaces wird je ein Entwurf dem Apple- bzw. Windows-Style-Guide entsprechend und ein „subjektiverer" Entwurf mit nicht an die Interface-Standards geknüpften Funktionselementen ausgearbeitet.

Prinzipien

Wodurch unterscheiden sich die Interaktionsstrukturen aus dem Seminar von der des eingangs beschriebenen Autoradios? Warum ist ein Interface schlecht, ein anderes gut bedienbar?

Zwar gibt es keine Rezepte, die zu einer guten Interaktionsstruktur führen, doch können einige grundlegende Prinzipien genannt werden, die mit der Bedienbarkeit von Interfaces in Beziehung stehen.

- Prinzipiell ist das Interface eines Produktes bzw. einer Software so zu strukturieren und zu gestalten, daß es den Denkmodellen und der kognitiven Auffassung des Menschen möglichst nahekommt.
  Um dies zu erreichen, müssen sich die Entwickler mit dem Wesen des Systems auseinandersetzten:
  Was ist der Nutzen?
  Welche Funktionen bieten Zugriff auf diesen Nutzen?
  Wie werden diese Funktionen repräsentiert?
- Weitere wesentliche Parameter, die das Interface mitbestimmen, sind der Benutzerkreis und die Nutzungssituation.
- Als Kriterien, die bei der Repräsentation von Funktionen mit der Benutzbarkeit in Beziehung stehen, sind beispielsweise anzuführen:
  Die Gewichtung von Funktionen und Funktionsgruppen entsprechend der Benutzungshäufigkeit oder -bedeutung für das Gesamtsystem; dies kann u. a. durch Positionierung und Größe von Elementen erreicht werden.
- Alle Elemente sollten in das Gesamtsystem strukturell und semantisch eingebettet werden.
  Räumliches Zuordnen und Darstellen von Bezügen kann die Bedeutung von Elementen besser lesbar machen. Elemente, die räumlich isoliert sind, können nicht in Wirkzusammenhänge mit anderen Elementen gebracht werden und erklären sich dadurch meist nicht selbst.
  Das optimale Interface ist (mindestens in seinen Primärfunktionen) ohne Beschriftung, Erläuterung oder Anleitung direkt bedienbar. Erdachte Systemstrukturen müssen deutlich ausgearbeitet und dargestellt werden, damit diese auf einen Blick erfaßt und verstanden werden können.
- Mehr Funktionen können mehr Komfort bieten. Sie bedingen meist aber auch mehr Funktionselemente und führen zu einem komplexeren – und in der Regel auch komplizierter zu bedienenden – System. Im Idealfall wird die Anzahl von Elementen möglichst gering und ohne undurchsichtige Mehrfachbelegungen von Funktionselementen gehalten.

Übrigens: Im nächsten Semester wird ein Computational-Design-Seminar angeboten, in dem die Studierenden ein komplettes Produkt- und Interface-Konzept erarbeiten und simulieren. Es handelt sich um … genau: ein Autoradio. ▮

this development stage, because it enables students to visualise visual structures and operating sequences very easily and quickly, without special software knowledge. The resulting layouts and sequences are saved.

This first design phase is followed by a discussion. Each student presents his or her own design. Observed by the rest of the group, a fellow student then carries out several previously agreed operations. After that, the sequence is discussed and critically evaluated by the group. As a rule, this task-oriented approach makes the students aware of typical mistakes. Furthermore, the basic principles of interface structures can be elucidated by means of the concrete examples. At the end of the first round of discussions, all the students have received several points of criticism and suggestions for improvements on their design. They then produce a revised version (see ill. 2–4). This design and discussion cycle is usually repeated two or three times, leading to an iterative optimisation of the design, which is so necessary when developing software.

Respective of the size of the project, the final structure design is worked out in different variations (see ill. 5, 6). In principle, the structures are applicable to hardware and software interfaces. In the case of the software interfaces, one design is worked out conform to the Apple or Windows Style Guide, and a "more subjective" one with function elements that are not bound to those interface standards.

Principles

What is the difference between the interaction structures in the seminar and those described at the beginning in the context of the car radio? Why is one interface easy to operate and the other not?

There are no guaranteed recipes for a good interaction structure, however, some basic principles can be mentioned as regards the usability of interfaces:

- *Basically, the interface of a product or software should be structured and designed so as to correspond as closely as possible to the user's cognitive models.*
  To achieve this, designers must come to terms with the essence of the system:
  What is its use?
  What functions provide access to that use?
  How are these functions represented?
- *Other essential parameters influencing the interface are the user group and the situation in which it is to be used.*
- *Some criteria related to the representation of functions and their usability are, for example:*
  Functions and function groups should be weighted according to frequency of use or importance for the whole system; this can be achieved by the position and size of the elements, among other things.
  All the elements should be embedded structurally and semantically in the whole system.
  The spatial position and presentation of links can make it easier to read the meaning of the elements. Spatially isolated elements cannot be easily referred to others, their function cannot be deduced, and they are usually not self-explanatory.
  The optimum interface (at least in its primary functions) is immediately usable without text, explanation, or instruction. Newly devised system structures must be clearly worked out and represented so that they can be understood at a glance.
- *More functions can offer more convenience. However, usually they also mean more function elements and a more complex system, which, as a rule, is also more complicated to operate. Ideally, the number of elements should be kept as small as possible and these should not be given unclear multi-functions.*

By the way: next semester there will be a Computational Design seminar on offer in which students will work on and simulate a complete product and interface concept. The product is, what else, a car radio. ▮

Translated from the German by Pauline Cumbers.