
Design von Benutzungsoberflächen als ingenieurmäßiger Prozeß

Axel Viereck
Universität Oldenburg

Zusammenfassung

Bei der Entwicklung von Software muß für deren Benutzungsoberfläche zwischen dem Designprozeß, durch den festgelegt wird, wie der Benutzer das System für seine Zwecke einsetzen kann und handhaben muß, und dem Herstellungsprozeß, der die Umsetzung der zuvor entwickelten Ideen in ein Computer-Programm betrifft, unterschieden werden. Dieser Beitrag konzentriert sich auf den Designprozeß, stellt ihn als einen ingenieurmäßigen Konstruktionsprozeß dar und leitet aus den Erkenntnissen der Konstruktionstheorie Folgerungen für eine methodische und werkzeugartige Unterstützung des Dialog-Designers bei der Konstruktion von Benutzungsoberflächen ab.

1 Motivation

Die Entwicklung von interaktiven Programmen ist ein schöpferischer geistiger Prozeß. Oberquelle vergleicht die Tätigkeiten zum Software-Design mit denen eines Architekten: "Aus meiner Sicht ist die Aufgabe des Software-Ergonomen der eines Architekten ähnlich, der in Kenntnis der Bedürfnisse seiner Kunden und der technischen Möglichkeiten (Spiel-)Räume nach menschlichen Maßen entwirft, bei denen Funktion und Form harmonisieren und sich wechselseitig bedingen, in denen sich die Bewohner ohne Probleme orientieren können, ohne Behinderungen ihre Aufgabe erfüllen, sich wohlfühlen und sich entfalten können" (Oberquelle [7], S. 21).

Keil-Slawik (Keil-Slawik [5]) begründet für eine dem Benutzer unter der Prämisse der Aufgabenerfüllung angemessene Systemgestaltung ausführlich die Design-Kriterien "Orientierung auf den Handlungsabschluß", "Minimalität von Konventionen", "konstruktiver Bildschirmaufbau" und "Individualisierbarkeit". Außer der Zugrundelegung des STEPS-Ansatzes (vgl. Reisin [9]) werden allerdings keine Hinweise gegeben, wie der Entwickler zu einem Ergebnis kommen kann, das die geforderten Eigenschaften aufweist.

Im folgenden will ich genau an dieser Stelle ansetzen. Ausgehend von der Hypothese, daß die Entwicklung von interaktiven Programmen neben schöpferischen geistigen Anteilen auch immer solche mit vorwiegend systematischen oder routi-

nemäßigen Charakter beinhaltet, wird anhand von Erkenntnissen aus der Konstruktionstheorie untersucht, inwieweit solche Prozesse Raum für Unterstützungsmöglichkeiten bieten. Untersuchungen zu diesen Gesichtspunkten werden in Oldenburg seit etwa 1987 durchgeführt und führten zu dem Forschungsprojekt EXPOSE (Expertensystem zur phasen-orientierten Software-Ergonomie-Beratung bei der Benutzerschnittstellen-Entwicklung) (Gorny [3]).

Für den arbeitswissenschaftlich, denk- oder wahrnehmungspsychologisch ausgerichteten Software-Ergonomen mag das im folgenden verwendete, meist technik-zentrierte Vokabular zunächst auf Widerspruch stoßen. Ich bitte ihn um Geduld und um Verständnis für den Versuch, sich dem Software-Design einmal aus einer ganz anderen Richtung zu nähern.

2 Modelle für den Konstruktionsprozeß

Konstruktionstheoretiker beschreiben den Konstruktionsprozeß als einen "heuristischen Algorithmus", der aus Einzeltätigkeiten und Operationen besteht, die voll algorithmisch erfaßt werden können, und aus Lücken im Ablauf der Bearbeitung, die heuristischen Charakter haben (vgl. z.B. Baatz [1], Schlottmann [10]). Hubka unterstellt für eine erfolgreiche Konstruktion ebenfalls eine Kombination aus methodischem Vorgehen und Intuition: "Jeder Konstruktionsprozeß wird ... aus Such-, Verifikations- und Entscheidungsschritten zusammengesetzt, welche intuitiv (in Form von Sprüngen) oder methodisch durchgeführt werden. Der Anteil, den jede dieser beiden Arbeitsweisen hat, differiert in Abhängigkeit von der Aufgabe, der Lösungssituation und besonders vom Konstrukteur selbst. In jedem Falle läßt die Intuition den Konstrukteur systematisch arbeiten, wenn er das methodische Konstruieren studiert und benutzt; daneben rüstet die methodische Analyse das Gedächtnis mit besseren Fachinformationen aus, die als Stoff von den unbewußten Denkvorgängen ausgenutzt werden" (Hubka [4], S. 70).

Den methodischen Rahmen dafür bietet die Konstruktions- und Entwurfssystematik mit zyklischen Phasenmodellen auf der Grundlage einer hierarchischen Strukturierung von technischen Systemen. Als Konstruktionsphasen werden "Konzipieren - Entwerfen - Ausarbeiten" oder "Funktionsfindung - Prinzipiarbeitung - Gestaltung - Detaillierung" unterschieden. Eine aufgabenbezogene Gliederung von technischen Systemen erfolgt z.B. in "Funktionshauptgruppen - Funktionsgruppen - Funktionskomplexe - Funktionselemente - elementare Funktionsträger". Für die unterste Ebene wurden dazu eine Reihe von elementaren Funktionen herausgearbeitet, auf die unabhängig vom zugrundeliegenden Problembereich letztlich jedes technische System zurückgeführt werden kann (vgl. VDI 2222 [14]). In dieser

Vorgehensweise finden sich viele Parallelen zum Software-Entwicklungsprozeß, für den ebenfalls prozeß- und aufgabenorientierte Zerlegungen zur Komplexitätsreduzierung propagiert werden.

Ansätze für Unterstützungsmöglichkeiten der eigentlichen geistig-intuitiven Tätigkeiten des Konstruierens finden sich in Zusammenhang mit Untersuchungen zu wissensbasierten CAD-Systemen. Den "Suchprozeß" beim Konstruieren in den Vordergrund rückend wurde hier der Ansatz verfolgt, dem Konstrukteur Systeme zur Suche nach der optimalen Lösung seines Problems zur Seite zu stellen. Dazu erfolgten Formalisierungen des Kontextes und des allgemeinen Konstruktionswissen in Form von wissensbasierten Mechanismen, durch die für den Suchprozeß entsprechend eines vorgegebenen Ziels Lösungen abgeleitet wurden.

Kritik an diesem Ansatz bezieht sich einerseits darauf, daß der "Suchprozeß" bekannte Lösungen voraussetzt und deshalb keine Problemlösung, sondern lediglich eine Aufgabenbearbeitung vorliege. Andererseits treten Schwierigkeiten dann auf, wenn Systematiken für die Arbeit nicht vorliegen, wenn Ziele sich während des Prozesses ändern, wenn durch gefundene Teil-Lösungen neue Probleme erkannt und gelöst werden müssen. Für Probleme dieser Art wurden Begriffe wie "Innovatives Design" und "Kreatives Design" eingeführt. Maher formuliert dazu: "One way of dealing with this difficulty is to identify models of the design synthesis process, assuming that formulation has occurred. Another way is to allow synthesis to proceed even with a change of goals. The implication here is that using search as a model for the design process is too general; more specific models that use search in various ways are needed to bridge the gap between a model of design and the eventual representation of design knowledge and experience" (Maher [6], S. 51). Hier seien drei Konzepte vorgestellt, die zumindest teilweise auch zu Modellen weiterentwickelt wurden.

Modell 1:

Für eine exaktere Modellbildung der geistig-intuitiven Tätigkeiten liegen einige grundsätzliche psychologische Arbeiten zum Verständnis der mentalen Vorgänge beim Suchprozeß eines Konstrukteurs vor. Nach empirischen Untersuchungen von Takeda u.a. (Takeda [12]) ist das Problemlösungs-Verhalten geprägt durch eine objektbezogene Ebene, auf der der Konstrukteur über das zu konstruierende Objekt, seine Eigenschaften und sein Verhalten nachdenkt, Bewertungen vornimmt und Entscheidungen trifft, und durch eine Aktions-Ebene (Meta-Ebene), auf der der Designer Überlegungen zu seinem Vorgehen bei der Problemlösung anstellt. Beide Ebenen verlaufen parallel in einem zyklischen Prozeß: Bewußtmachung eines zu lösenden Problems der Aufgabenstellung, Annahmen über Grundkonzepte, die zur Lösung des Problems zu beachten sind, Synthese möglicher Problemlösungen,

basierend auf den Grundkonzepten, Bewertung der erarbeiteten Problemlösungen, Feststellung von neu auftretenden Problemen, Entscheidung für eine geeignete Problemlösung.

Bei der eigentlichen Synthese verläuft nach Maher (Maher [6]) der hier notwendige schöpferische Prozeß nach unterschiedlichen Mustern, die sie zur Formalisierung für wissensbasierte Systeme in drei Klassen kategorisiert: Zerlegung, Case-based Reasoning, Transformation. Ähnliche Muster ergeben sich - neben anderen - auch aus den empirischen Untersuchungen von Konstruktionstätigkeiten von Ullman (vgl. Ullman [13]).

Modell 2:

Die Zerlegung eines komplexen Problems über mehrere Ebenen in kleinere, überschaubarere Teilprobleme ist in der Konstruktionstheorie ein weit verbreiteter und allgemein anerkannter Weg zur Suche nach Problemlösungen. Dabei kann einerseits der Prozeß in Teilschritte zerlegt (Phasenmodelle) und andererseits das zu konstruierende Objekt, bzw. die Gesamtfunktionalität in Teilobjekte, bzw. in Teilfunktionen aufgeteilt werden (aufgabenbezogene Struktur). Daneben weist Maher auf die Möglichkeit einer Zerlegung nach dem zur Problemlösung benötigten Wissen hin, durch die das komplexe Konstruktionswissen strukturiert und damit einfacher handhabbar wird. Für die Durchführung von Zerlegungen gilt der Grundsatz der größtmöglichen Unabhängigkeit der gebildeten Komponenten; wie man vorzugehen hat, um dies zu erreichen, kann nach Maher nicht allgemein vorgegeben werden.

Modell 3:

Beim Case-based Reasoning benutzt der Konstrukteur sein Erfahrungswissen, indem Lösungen oder Lösungswege einer vorangegangenen Problemstellung auf das neue Problem übertragen werden. Das mentale Vorgehen erfolgt "episodenartig" durch Bewußtmachung zurückliegender Lösungen oder Vorgehensweisen, die ohne Verallgemeinerungen auf ihre Anwendbarkeit für das neue Problem hin eingeschätzt und nach Auswahl einer vielversprechenden für den Problemlösungsprozeß benutzt werden. Durch Test und Bewertung der erhaltenen Lösung wird das Erfahrungswissen des Konstrukteurs erweitert.

Für eine Transformation wendet der Konstrukteur das aus seiner vorangegangenen Tätigkeit oder seiner Ausbildung erlangte, durch Regeln und Algorithmen generalisierte Wissen systematisch an, um von gegebenen Anforderungen zu einer Lösung zu gelangen.

Die drei Modelle zum Vorgehen von Konstrukteuren sind in der Praxis nicht streng voneinander zu trennen. Grundsätzlich wird im Konstruktionsprozeß zunächst die

Zerlegung vorgenommen, um Problemstellungen herauszuarbeiten, für die dann eine der beiden anderen Verfahrensweisen zur Problemlösung anwendbar ist. Maher resümiert: "... design decomposition can provide an overall model for design, where each system-subsystem can be designed using case-based reasoning when a relevant case is available, a set of transformations when a grammar is available, or further decomposition when the generalized decomposition is understood" (Maher [6], S. 57).

3 Ein kognitives Modell für die Benutzungsoberflächen-Gestaltung

Konzipieren, Entwerfen und - mit Abstrichen - auch die detaillierte Ausarbeitung eines technischen Systems sind Prozesse, die wesentlich durch ein geistig-intuitives Vorgehen des Konstrukteurs bestimmt werden. Für das Software-Engineering sind

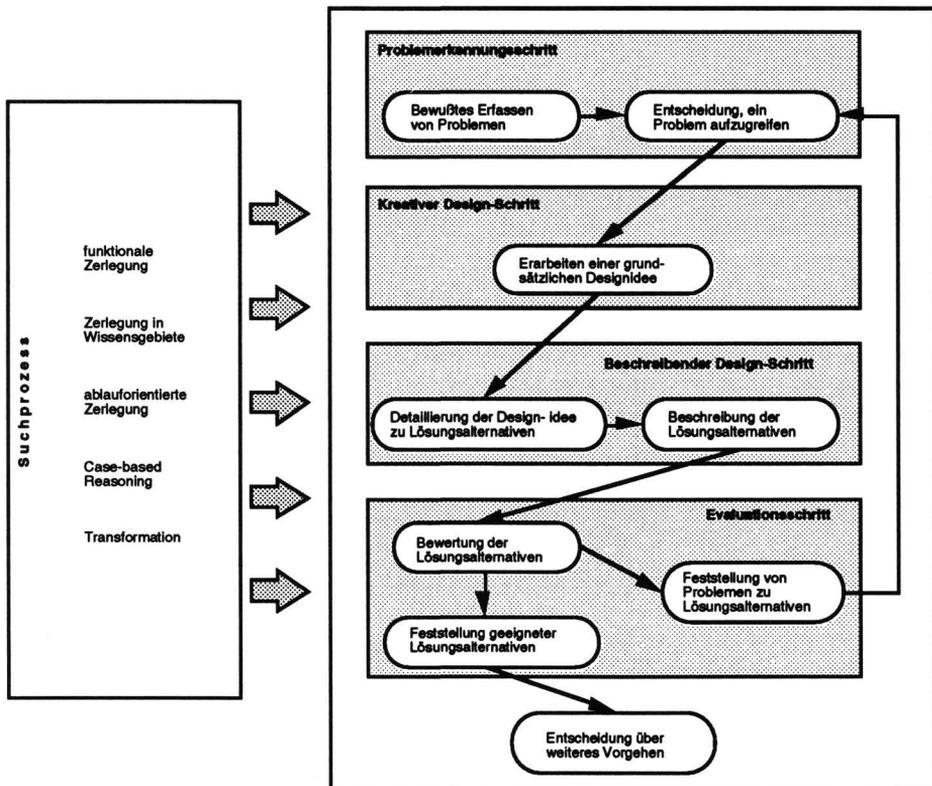


Abb. 1: Kognitives Modell des Prozesses für das Benutzungsoberflächen-Design

dazu eine Vielzahl von komplexitätsreduzierenden Methoden und Verfahren entwickelt worden, insbesondere die Zerlegung in Phasen und die getrennte Bearbeitung von Modulen. Wir gehen von der Hypothese aus, daß dieser Ansatz ebenso für den Dialog-Designer bei der Ausarbeitung der Benutzungsoberfläche eines Computer-Systems angewendet werden kann: Benutzungsoberflächen können systematisch konstruiert werden, ohne dabei auszuschließen, daß der Konstrukteur (der Dialog-Designer) auch intuitiv (und innovativ) das systematische Vorgehen durchbricht, um eine neue Idee weiterzuverfolgen.

In Anlehnung an das kognitive Modell von Takeda [12] läßt sich die kognitive Arbeit des Dialog-Designers wie folgt als zyklischer Prozeß beschreiben (vgl. Abb. 1).

- Im *Problemerkennungsschritt* erfaßt der Dialog-Designer bewußt die aus der Aufgabenstellung resultierenden, unmittelbar für die Gestaltung der Benutzungsoberfläche zu lösenden Probleme und entscheidet, eines dieser Probleme aufzugreifen und zu bearbeiten.
- Im *kreativen Designschritt* erarbeitet der Dialog-Designer eine grundsätzliche Designidee, von der er annimmt, daß sie zur Lösung des bearbeiteten Problems tauglich ist.
- Im *beschreibenden Designschritt* detailliert der Dialog-Designer seinen Lösungsansatz und beschreibt Lösungsalternativen für das bearbeitete Problem.
- Im *Evaluationsschritt* vergleicht und überprüft der Dialog-Designer seine Lösungsalternativen vor dem Hintergrund des gestellten Problems. Er stellt gegebenenfalls mit den Lösungen verbundene neue zu lösende (Teil-)Probleme fest oder er findet eine zufriedenstellende Lösung. Je nach Ausgang der Evaluation durchläuft er die einzelnen Schritte für das Problem bzw. Teilproblem erneut oder er widmet sich einem neuen Problem.

Um für Oberflächengestaltung aus diesem Modell umsetzbare Erkenntnisse abzuleiten, müssen einerseits die Detaillierungen von Maher zu den unterschiedlichen Lösungsstrategien innerhalb der einzelnen Schritte in das Modell integriert werden; andererseits ist das Modell mit den allgemeinen Vorgehensmodellen der Konstruktionstheorie in Beziehung zu setzen.

4 Konstruktion von Benutzungsoberflächen durch Zerlegung

Das Zerlegen eines komplexen Problems in einfachere Teilprobleme ist ein wesentliches Prinzip beim Lösen von Problemen. Es wird im Software-Engineering genau wie in der Konstruktionstheorie in zweierlei Hinsicht zum Ausgangspunkt gewählt. Einerseits wird das zu konstruierende Produkt aufgabenorientiert unterteilt. Andererseits wird der Entwicklungsprozeß in Phasen gegliedert.

4.1 Aufgabenorientierte Zerlegung

Aus der aufgabenorientierten Aufteilung entstehen Funktionsgruppen, Funktionskomplexe, Funktionselemente und elementare Funktionsträger in hierarchischer Anordnung zur Gliederung der Gesamtfunktionalität eines technischen Systems, im Software-Engineering mit Programmen, Modulen, Prozeduren und Kontrollstrukturen bezeichnet. Diese Art der Zerlegung im Software-Engineering ist allerdings an der Implementierung und nicht an der Benutzung der Software orientiert. Für eine aufgabenorientierte Zerlegung aus der Sicht des Benutzers, also für eine aufgabenorientierte Zerlegung der Benutzungsoberfläche, ist eine andere Sichtweise nötig. Zwei Herangehensweisen sind hier zu unterscheiden.

Ausgehend von der Funktionalität des Programms werden zum Aufbau der Werkzeugschnittstelle dazu nötige Funktionsbereiche aus den Aufgaben abgeleitet, die wiederum - Teilaufgaben entsprechend - in untergeordnete Funktionen und Unter-Funktionen gegliedert werden. Den so entstehenden Funktionen der Benutzungsoberfläche werden in der Dialogschnittstelle Dialogformen und auf der Ein-/Ausgabeschnittstelle Darstellungsformen zugeordnet. Dies ist das übliche Vorgehen und spiegelt sich in den Benutzerhandbüchern von Software wider.

Ein anderer Weg für eine "aufgabenorientierte Zerlegung" ist der objektorientierte Ansatz. Hier werden - ausgehend von den in einem Anwendungszusammenhang durchzuführenden Aufgaben - die zu manipulierenden Objekte und die für sie nötigen Manipulationsmöglichkeiten betrachtet. Es entstehen Objekthierarchien, Objektklassen und ihnen zugeordnete Funktionen auf der Werkzeugschnittstelle, für die dann Dialog- und Präsentationsformen auf den nachfolgenden Ebenen der Benutzungsoberfläche festgelegt werden.

Beide Wege zur aufgabenorientierten Zerlegung bieten die Möglichkeit, Methoden zur Problemlösung aus der Konstruktionstheorie (Rückführung auf Elementarfunktionen und Morphologischer Kasten) für das Software-Design zu übernehmen. So können für einen zugrundeliegenden Anwendungszusammenhang aus den hier relevanten Objekten und den ihnen zuzuordnenden Manipulationsmöglichkeiten, bzw., bei genügender Zerlegung der Funktionalität, auch für die Benutzungsoberfläche elementare Funktionen abgeleitet werden und für sie software-ergonomische Lösungen vorgegeben werden. Diesen Weg gehen zumindest für die Ein-/Ausgabeschnittstelle die User Interface Style Guides mit der Definition von Interaktionsobjekten und Festlegung ihrer Manipulationsmöglichkeiten und Darstellungsformen (z.B. OSF [8], Sun [11]).

Eine darüberhinausgehende Vorgabe standardisierter Lösungen erscheint mir insofern möglich, als diverse Funktionen und Objekte aus Programmen

kontextübergreifend in vielen Situationen immer wieder auftreten. Das Suchen oder das Eintragen einer Adresse betrifft die Kunden eines Versandhauses genau wie die einer Versicherung, einer Bank, eines Arztes oder eines Kraftfahrzeughändlers. Kunden- und Lieferantendaten eines Unternehmens haben viele Gemeinsamkeiten. Hier können für gleiche Grundoperationen gleiche software-ergonomische Teil-Lösungen standardisiert vorgegeben werden. Bei Betrachtung von komplexeren Funktionsträgern, könnten dann auch dynamische Aspekte der Werkzeugschnittstelle und der Dialogstruktur mit in die Erarbeitung von standardisierten Teil-Lösungen aufgenommen werden. Insbesondere ist es hier bei Berücksichtigung der unterschiedlichen Dialogarten an der Benutzungsoberfläche (vgl. Viereck [15]) möglich, dieses Verfahren erfolgversprechend einzusetzen. Kontextübergreifend werden für Anwendungsfunktionen baukastenartig eine Reihe von software-ergonomisch notwendigen Adaptier-, Steuer- und Metafunktionen mit Dialog- und Präsentationsformen standardisiert.

4.2 Vorgehensorientierte Zerlegung

Ein Vorgehensmodell zur Zerlegung des Konstruktionsprozesses von Benutzungsoberflächen muß zur Klärung der Aufgabenstellung und zum Aufbau einer aufgabenorientierten Zerlegung den Einsatz von Arbeitsanalysemethoden vorgeben. Des weiteren muß ein Vorgehensmodell eine iterative Problemlösung unterstützen. Die Tatsache, daß Technische Systeme nicht ad hoc, sondern in einem über mehrere Stufen erfolgenden Optimierungsverfahren konstruiert werden, gilt ebenso für das Dialog-Design. Insofern entsprechen die Vorgehensmodelle aus der Konstruktions-theorie, die neueren Ansätze des Software-Engineering (z.B. STEPS) und der Software-Ergonomie (MUSE) einander. Sie befinden sich weiterhin in Einklang mit dem kognitiven Modell eines zyklischen Prozesses beim Konstruieren. Die hier notwendigen Evaluationsschritte können nicht allein vom Dialog-Designer durchgeführt werden. Hierzu ist eine Beteiligung der Benutzer mit einem Einsatz von Prototypen unter möglichst realistischen Bedingungen erforderlich.

4.3 Wissensorientierte Zerlegung

Das Erarbeiten von Lösungen zur Organisations- und Werkzeugschnittstelle erfolgt auf einem anderen Erfahrungshintergrund als Entscheidungen zur Dialog- oder Ein-/Ausgabeschnittstelle. Insofern bietet es sich an, auch eine Zerlegung des Problembereichs entsprechend des für die Problemlösung notwendigen Wissens vorzunehmen. Hierdurch wird es möglich, das gesamte zur Konstruktion von Benutzungsoberflächen benötigte Wissen (Arbeits- und Organisationswissenschaften, Kogni-

tionswissenschaft, Wahrnehmungs- und Gestaltpsychologie und Informatik) zu strukturieren und dadurch für den Dialog-Designer einfacher handhabbar zu machen.

Durch eine Verbindung der Grundprinzipien entsteht ein methodischer Rahmen, genannt MUSE (Methode für ein User Interface Engineering), innerhalb dessen der Dialog-Designer bei der Erarbeitung von Problemlösungen einen erfolgversprechenden Weg entlang geführt wird. Kern der Methode ist eine Zuordnung der Wissensbereiche zu Sichten, die ein Dialog-Designer nacheinander einzunehmen hat, um sämtliche Fragestellungen vollständig zu bearbeiten. Die zeitliche Ordnung der Sichten führt zu einem Phasenmodell, das ausführlich in Viereck [16], beschrieben wurde.

Auf der Grundlage der Methode MUSE wird gegenwärtig ein wissensbasiertes System erarbeitet, das den Designer von Benutzungsoberflächen beratend bei ihrem Gestaltungsprozeß begleiten und ihn dabei in den ihm weniger vertrauten Wissensbereichen weiter qualifizieren soll (vgl. Gorny [3]).

5 Schlußbemerkungen

Ich bin mir bewußt, daß die Hypothese, die Benutzungsoberflächengestaltung sei systematisch zerlegbar und wesentliche Teile davon routinemäßig und regelhaft bearbeitbar, teilweise auf scharfen Widerstand stößt. Vergleichbare Diskussionen haben jedoch viele Technikbereiche bereits hinter sich: Maschinenbau und Bauingenieurwesen wurden z.B. noch bis in das späte 19. Jahrhundert so betrachtet. Gerade die Diskussion über die Architektentätigkeit hat aber die Freiräume für die seltenen Aufträge zu "großen" Einzelobjekten aufgezeigt, während für die Standardaufgaben eine Vielzahl von Normen, Regeln und baurechtlichen Bedingungen lösungsbestimmend sind.

Einzuordnen ist diese Diskussion außerdem in den Prozeß des fortlaufenden "Reifens" beim Bewältigen neuer technischer Entwicklungsstufen. Eine Meta-Beschreibung für das Durchlaufen von technischen Entwicklungsstufen bietet das BRETAM-Modell (Gaines [2]), das für Technikbereiche von einer Phase des "Breakthrough", der Nachahmung des Breakthrough ("Replication"), des Experimentierens, der Theoriebildung, der theoriebasierten Anwendung und schließlich der Reife ("Maturity") mit regelbasierter Anwendung führt. Mit dem Vorschlag, auch das Design von Benutzungsoberflächen systematisch zu betreiben, soll der Prozeß der Theoriebildung beschleunigt und die regelhafte Gestaltung von Oberflächen eingeleitet werden.

Literatur

- [1] U. Baatz: Bildschirmunterstütztes Konstruieren. Düsseldorf: VDI-Taschenbücher, VDI-Verlag, 1973
- [2] B.R. Gaines: A Methodological Framework for the Design and Evaluation of Software in System Involving Complex Human-Computer-Interaction. In: Schönflug, Wittstock (Hrsg): Software-Ergonomie 87. Stuttgart: Teubner, 1987
- [3] P. Gorny u.a.: EXPOSE - Expertensystem zur phasenorientierten Software-Ergonomie-Beratung bei der Benutzerschnittstellen-Entwicklung. Internes Papier. Oldenburg, Rostock, 1990
- [4] V. Hubka: Theorie der Konstruktionsprozesse. Berlin, Heidelberg, New York: Springer, 1976
- [5] R. Keil-Slawik: Konstruktives Design. TU Berlin, Bericht 90-14, März 1990
- [6] M.L. Maher: Process Models for Design Synthesis. AI Magazine, Winter 1990, pp 49-58
- [7] H. Oberquelle: MCI - Quo Vadis? Perspektiven für die Gestaltung und Entwicklung der Mensch-Computer-Interaktion. In: Ackermann, Ulich (Hrsg): Software-Ergonomie 91. Stuttgart: Teubner, 1991
- [8] Open Software Foundation: OSF/Motif Style Guide. Englewood Cliffs: Prentice Hall, 1991
- [9] F. Reisin, G. Schmidt: Steps - Ein Ansatz zur evolutionären Systementwicklung. In: Jansen, Schwitalla, W (Hrsg): Beteiligungsorientierte Systementwicklung. Opladen: Westdeutscher Verlag, 1989
- [10] D. Schlottmann (Hrsg): Konstruktionslehre. Wien, New York: Springer, 1983
- [11] SUN Microsystems, Inc: Open Look Graphical User Interface Application Style Guide. Reading: Addison-Wesley, 1990
- [12] H. Takeda u.a.: Modeling Design Processes. AI Magazine, 1990 (Winter), pp 37-48
- [13] D. Ullman, T. Dieterich, L. Stauffer: A Model of the Mechanical Design Process Based on Empirical Data. In: AI EDAM, 1988, 2(1), pp 33-52
- [14] VDI-Richtlinie 2222: Konstruktionsmethodik. Konzipieren technischer Produkte. Düsseldorf: VDI-Verlag, 1977
- [15] A. Viereck: Ein Phasenmodell zur Gestaltung von Benutzungsoberflächen. Oldenburg: Projekt EXPOSE, interner Bericht Expose.UO.9101, 1991
- [16] A. Viereck: Computer-unterstützte Benutzungsoberflächen-Gestaltung. In: Konradt, Drisis (Hrsg): Benutzungsoberflächen in teilautonomer Gruppenarbeit. Köln: Leske&Budrich, 1992

Dr. Axel Viereck
Universität Oldenburg
FB Informatik
Postfach 2503
2900 Oldenburg