

# A Pattern Approach to Interaction Design

Jan Borchers

Technische Universität Darmstadt  
Fachgruppe Telekooperation  
<http://www.stanford.edu/~borchers/>

Diese Arbeit stellt eine Methode zur Darstellung von Entwurfserfahrungen in der Gestaltung von Benutzerschnittstellen vor, die auf dem Konzept der *Entwurfsmuster* beruht, dieses jedoch formalisiert, erweitert und vereinheitlicht, um auch Richtlinien zur Programmentwicklung und das Wissen der Anwendungsdomäne eines Softwareprojekts in gleicher Weise zu modellieren. Nach einer Darstellung der Problematik beim Entwurf von Benutzerschnittstellen wird die Entwicklung von Entwurfsmustern in Architektur, Programmierertechnik und Mensch-Maschine-Interaktion (MMI) untersucht und eine Reihe von Anforderungen an ein Entwurfsprinzip zur Lösung dieser Probleme aufgestellt.

Anschließend wird der neuartige Entwurfsansatz präsentiert und anhand dreier Mustersprachen für diese Gebiete demonstriert. Die Sprachen spiegeln Erfahrungen aus einer Reihe von Projekten wider, in denen der Autor interaktive Exponate entwarf. Da das Thema der meisten Exponate musikalischer Natur war, wird dieses Anwendungsgebiet in einer Mustersprache für Bluesmusik repräsentiert. Die zentrale MMI-Mustersprache beschreibt Regeln für die Gestaltung interaktiver Exponate, während die Muster zur Programmentwicklung sich mit Lösungsvorschlägen für die Entwicklung interaktiver Musiksoftware befassen.

Der Einsatz dieses musterbasierten Entwurfskonzepts wird abschließend anhand verschiedener Kriterien evaluiert und der Entwurf eines Werkzeugs zur rechnergestützten Arbeit mit Entwurfsmustern vorgestellt. Im Anhang der Arbeit findet sich unter anderem eine Darstellung einer typischen Interaktion mit *WorldBeat* [Bor97], einem ausgezeichneten interaktiven Musikexponat, das vom Autor entworfen wurde und in vielen Entwurfsmustern dieser Arbeit als Beispiel zitiert wird.

## 1 Einleitung

Das Forschungsgebiet der Mensch-Maschine-Interaktion (MMI) befaßt sich mit der Gestaltung der Schnittstelle zwischen Menschen und technischen Systemen [HLP97]. Hierbei fließen Informatik-orientierte Erkenntnisse aus Dialogtechnik, Computergraphik, Ein-/Ausgabegeräten und Entwicklungsprozessen mit psychologischen Bedingungen der menschlichen Informationsverarbeitung und sozialen Randbedingungen des Benutzungskontexts zusammen [ACM92].

Diese Benutzerschnittstelle spielt für den Erfolg interaktiver Systeme eine entscheidende Rolle [Lan95], allerdings müssen Experten aus den Bereichen Benutzerschnittstellengestaltung, Programmentwicklung und aus dem Anwendungsbereich eines Projekts zusammenarbeiten, um eine erfolgreiche Benutzerschnittstelle zu gestalten [Tog92]. Hier stellt die interdisziplinäre Kommunikation in der Entwurfsgruppe ein großes Problem dar [Kim90], ebenso wie das Bewahren von Entwurfserfahrungen über Projekte hinweg in der Form eines „Unternehmensgedächtnisses“.

Zur Lösung dieser Probleme schlägt diese Arbeit ein Entwurfsmodell vor, das das Wissen in der MMI, aber auch in Programmieretechnik und Anwendungsbereich eines Softwareprojekts einheitlich strukturiert und beschreibt. Das Modell basiert auf dem Konzept der *Entwurfsmuster*, das im nächsten Abschnitt beschrieben wird.

## 2 Entwurfsmuster: Stand der Entwicklung

### Entwurfsmuster in der Architektur

Ein *Entwurfsmuster* ist die textuelle Beschreibung einer erfolgreichen Lösung für ein häufig wiederkehrendes Problem beim Entwurf von Systemen. Der Begriff stammt ursprünglich aus der Architektur, wo Baumeister bereits in der Renaissance begannen, erfolgreiche Konstruktionslösungen systematisch zu dokumentieren [Haa99]. Die Bezeichnung „Muster“ wurde von Alexander [Ale79] geprägt, um derartige Lösungen für wiederkehrende Entwurfsprobleme in der Städtearchitektur einheitlich und strukturiert darzustellen. Ziel ist, durch Anwendung vieler Muster in Kombination in den resultierenden Häusern, Nachbarschaften und Städten eine *namenlose Qualität* zu erzeugen, die die Lebensqualität der Bewohner verbessert.

Jedes der über 250 Muster aus Alexanders Mustersprache [AIS<sup>+</sup>77] beschreibt einen Konflikt bei der Gestaltung von Lebensräumen sowie dessen Auflösung durch architektonische Lösungen. So stellt das Muster „SITTING WALL“ folgenden Konflikt dar: Zu hohe Mauern schirmen beispielsweise einen Park stark ab, ohne Mauern hingegen fehlt jegliche Abgrenzung. Die Lösung sind oben verbreiterte Mauern in Sitzhöhe, die neben einer angemessenen Abgrenzung auch zum Verweilen einladen.

Alexanders Muster sind miteinander zu einer Hierarchie verknüpft, die von Mustern für die Gestaltung ganzer Städte („COMMUNITY OF 7000“) über Muster mittleren Ausmaßes („STREET CAFE“) bis hinunter zur Gestaltung einzelner Objekte in einem Gebäude („WINDOW SEAT“) reicht und so den Architekten in seiner Arbeit leiten und unterstützen kann. Diese intensive Verknüpfung macht aus einer Sammlung von Entwurfsmustern eine *Entwurfsmustersprache*. Diese Muster sind jedoch so geschrieben, daß sie nicht nur für Architekten, sondern insbesondere auch für Bewohner verständlich sind, damit letztere am Entwurfsprozeß ihrer Wohn- und Arbeitsumgebungen aktiv mitwirken können.

Alexanders Entwurfsmuster weisen in Inhalt, Struktur und Präsentation einige wichtige Eigenschaften auf. So besteht ein Muster stets aus denselben Bestandteilen, die in derselben Reihenfolge präsentiert werden, und die Darstellung erfolgt in einer typographisch sehr konsistenten Art, um dem Leser das rasche Erfassen des Inhalts zu erleichtern. Da dieses Format weitgehend demjenigen entspricht, das in der vorliegenden Arbeit definiert wurde, sei auf die weiter unten gegebene Beschreibung verwiesen.

## Entwurfsmuster in der Programmierertechnik

Um 1987 herum griff die Programmierertechnik die Idee der Entwurfsmuster auf [BC87], um erfolgreiche Techniken für Entwurf und Implementierung von Softwaresystemen zu beschreiben. Es etablierte sich eine eigene Konferenzreihe über Entwurfsmustersprachen in der Programmierung [CS95, Vli96, MRB98, HFR99], und insbesondere das Buch der *Gang of Four* [GHJV95] stieß in der Entwicklergemeinschaft auf große Resonanz.

Während die wesentlichen Komponenten eines Entwurfsmusters in der Programmierertechnik beibehalten wurden, sind Inhalt und Form jedoch nur mehr für die Kommunikation innerhalb der eigenen Disziplin geeignet; Benutzer von Softwaresystemen und andere Nichtprogrammierer können nur selten Nutzen aus diesen Entwurfsmustern ziehen. Dieser Wandel ist teilweise gerechtfertigt: Benutzer leben und arbeiten zwar direkt in oder mit den Artefakten, die Architekt und Benutzerschnittstellengestalter schaffen, sollten aber nicht direkt mit der internen Architektur eines Softwaresystems interagieren müssen—diese wird ja gerade durch die Benutzeroberfläche verdeckt. Trotzdem ist der Verzicht auf die Möglichkeit der Einbeziehung von Benutzern und Vertretern anderer Disziplinen anhand von Entwurfsmustern als gemeinsames Vokabular, wie auch das fehlende Streben nach der *namenlosen Qualität* in den entwickelten Systemen, ein Versäumnis gegenüber der ursprünglichen Idee der Entwurfsmuster [Ale96].

## Entwurfsmuster in der MMI

Obwohl Verweise auf Alexanders Konzept der Entwurfsmustersprachen in mehreren klassischen MMI-Texten und teilweise sogar früher als in der Programmierertechnik auftauchen [ND86, Nor88, App92], hat sich dieses Gebiet erst in den letzten Jahren intensiv zu entwickeln begonnen.

Barfield et al. [BBL<sup>+</sup>94] beschreiben ein Curriculum der Interaktionsgestaltung an der Kunsthochschule Utrecht, das stark von der Idee der MMI-Entwurfsmuster beeinflusst ist, und weisen zuerst darauf hin, daß MMI-Entwurfsmuster aufgrund der Dynamik von Benutzerschnittstellen im Gegensatz zur Architektur auch die zeitliche Dimension von Lösungen beschreiben müssen. Ein erstes Arbeitstreffen zu MMI-Entwurfsmustern [BBC<sup>+</sup>98] unterschied Muster wie in der vorliegenden Arbeit, die erfolgreiche Entwurfslösungen wiedergeben, von solchen, die wertfrei beobachtete Praxis in der Kommunikation von Menschen mit Maschinen und untereinander beschreiben.

Die größte derzeit verfügbare MMI-Entwurfsmustersprache stammt von Tidwell [Tid98]. Tidwells Muster zerfallen, zusätzlich zur Hierarchisierung nach räumlich-zeitlicher Größe ihres Wirkungskreises, in zwei Gruppen, je nachdem, ob sie primär erklären, wie *Inhalte* präsentiert werden können, oder ob sie die Frage behandeln, wie dem Benutzer *Aktionen* verfügbar zu machen sind. Weitere Arbeitstreffen, an denen der Autor teilnahm, verfeinerten die Definition von MMI-Entwurfsmustersprachen, lieferten eine Taxonomie und Beispiele typischer Entwurfsmuster, und evaluierten Entwurfsmuster in *Writers' Workshops* [Bor00a, GPB99].

## Entwurfsmuster in anderen Disziplinen

Das Prinzip der Entwurfsmuster wurde gelegentlich auch auf andere Bereiche wie beispielsweise Unternehmensorganisation und kognitive Psychologie übertragen [Bar92]; es existiert sogar eine Mustersprache für das Schreiben von Entwurfsmustern [MD98]. Ansätze, die Entwurfsmuster interdisziplinär verwenden [DD96, GL99], bieten jedoch kein einheitliches Format dieser Entwurfsmuster.

## Vergleich und Anforderungsanalyse

Ein Vergleich der wichtigsten Mustersprachen [AIS<sup>+</sup>77, GHJV95, Tid98] zeigt weitgehende bereinstimmung in den strukturellen Komponenten, jedoch deutliche Unterschiede in der Präsentation und inhaltlichen Ausführlichkeit der einzelnen Mustersprachen.

Aus dieser Analyse des Ist-Zustandes ergeben sich eine Reihe von *Anforderungen* an ein interdisziplinäres Entwurfsmodell, das auf Mustersprachen basiert: Neben Lesbarkeit außerhalb der eigenen Disziplin und einem einheitlichen wohldefinierten Format gehören dazu auch der empirische Nachweis für die Gültigkeit der Muster, ein dem jeweiligen Anwendungsbereich angemessenes, den Entwurf unterstützendes Ordnungsprinzip der Muster sowie die Modellierbarkeit auch zeitlicher Abläufe und die Integration in ein größeres Software-Entwicklungsmodell.

Diese Bedingungen werden in der Summe von keinem der existierenden Modelle und Ansätze erfüllt.

## 3 Ein interdisziplinärer musterbasierter Entwurfsansatz

Formal definiert diese Arbeit eine Mustersprache als einen gerichteten azyklischen Graphen, dessen Knoten Entwurfsmuster und dessen Kanten Referenzen darstellen. Entwurfsmuster selbst sind in dieser Notation eine Menge von Komponenten, die im folgenden erläutert werden:

- Der *Name* dient der schnellen Referenz auf ein Muster und hilft, ein Vokabular von Entwurfsmustern aufzubauen.
- Die *Bewertung* von 0–2 Sternen gibt an, inwiefern der Autor glaubt, daß die vorgestellte Lösung zwingend ist.
- Ein *Bild* zeigt ein reales Beispiel, an dem eine typische Anwendung des Entwurfsmusters auch für den Laien zu erkennen ist.
- Der *Kontext* erklärt, wann das Entwurfsmuster eingesetzt werden kann und welche größeren Entwurfsmuster einen guten Ausgangspunkt dafür bieten.

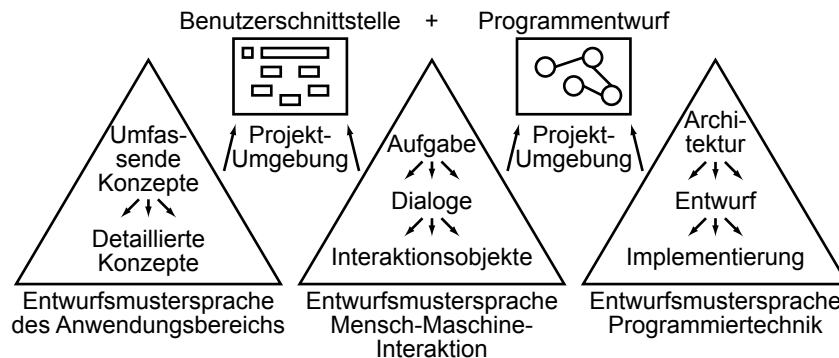


Abbildung 1: Drei Entwurfsmustersprachen, die Konzepte aus Anwendungsbereich, MMI und Programmierertechnik beschreiben, führen in einem Projekt zu Benutzerschnittstellen- und Programmwürfen.

- Das *Problem* faßt kurz den Konflikt zusammen, den das Entwurfsmuster löst.
- Die *Beispiele* zeigen empirisch, wie das Problem in existierenden Anwendungen gelöst wurde, oder geben theoretische Begründungen für die Qualität einer Lösung.
- Die *Lösung* generalisiert die Beispiele zu einer allgemeineren, aber konstruktiven Anleitung, wie das Problem dieses Entwurfsmusters in ähnlichen Situationen wieder gelöst werden kann.
- Das *Diagramm* visualisiert die wesentliche Idee des Entwurfsmusters in prägnanter, rasch erfäßbarer und einprägsamer Form, typischerweise als schematische Skizze.
- Die *Referenzen* verweisen auf detailorientiertere Muster, die zur Umsetzung dieses Entwurfsmusters herangezogen werden können. Kontext und Referenzen definieren die Hierarchie innerhalb einer Mustersprache, die den Gestalter von abstrakten, großformatigen Problemen zu kleinen Detailfragen führt.

Die formale Definition ermöglicht eine klare Darstellung des Konzepts der Entwurfsmuster und erleichtert die Programmierung geeigneter Werkzeuge für den Umgang mit ihnen. Die Muster selbst werden jedoch als illustrierter Fließtext präsentiert, um optimale Lesbarkeit zu garantieren. Dennoch ist der Text typographisch implizit exakt der obigen Definition entsprechend strukturiert.

Der Kern des hier vorgestellten Entwurfsansatzes sind die durch ihn entstehenden Sprachen von Entwurfsmustern, die Konzepte aus der MMI, aber auch aus Programmierertechnik und Anwendungsbereich eines Softwareprojekts modellieren (siehe Abb. 1).

Der musterbasierte Ansatz wurde in ein verbreitetes Prozeßmodell zur benutzerzentrierten Entwicklung interaktiver Programme, den von Nielsen [Nie93] definierten *Usability Engineering Lifecycle* eingebettet. Für jede der elf Phasen dieses Prozeßmodells wurde

erläutert, welche der drei Entwurfsmustersprachen in welcher Weise erstellt oder eingesetzt werden kann, um den Entwurfsprozeß zu unterstützen.

## 4 Eine Entwurfsmustersprache für interaktive Musikexponate

Als Nachweis der Umsetzbarkeit des vorgestellten musterbasierten Entwurfsmodells wurden drei Mustersprachen erstellt, die die Erfahrungen des Autors aus der Entwicklung einer Reihe interaktiver Exponate widerspiegeln.

Die erste Sprache beschreibt die Konzepte des Anwendungsgebiets „Bluesmusik“, die im interaktiven Musikexponat *WorldBeat* [Bor97] modelliert wurden, um das computer-gestützte Improvisieren zu Bluesmusik zu ermöglichen. Abb. 2 zeigt die Hierarchie der erstellten musikalischen Entwurfsmuster, die sich ergeben, wenn man den musikalischen Kompositions- und Improvisationsprozeß als eine Form des kreativen „Entwurfs“ betrachtet, bei dem als Artefakt ein Musikstück entsteht.

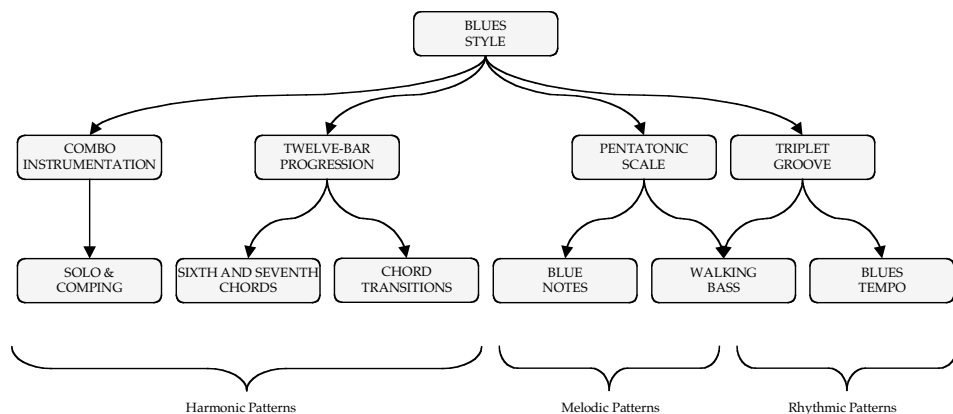


Abbildung 2: Die musikalische Entwurfsmustersprache der Improvisationskomponente des *WorldBeat*-Systems.

Die zweite und zentrale Mustersprache der Arbeit beschreibt Konzepte für die Gestaltung der Mensch-Maschine-Interaktion bei interaktiven Exponaten. Die Hierarchie dieser Mustersprache ist in Abb. 3 wiedergegeben.

Schließlich enthält die Arbeit auch noch eine Reihe von Entwurfsmustern aus der Programmieretechnik, die bewährte Konzepte zur Entwicklung interaktiver Musiksoftware beschreiben (siehe Abb. 4 und [BM98]).

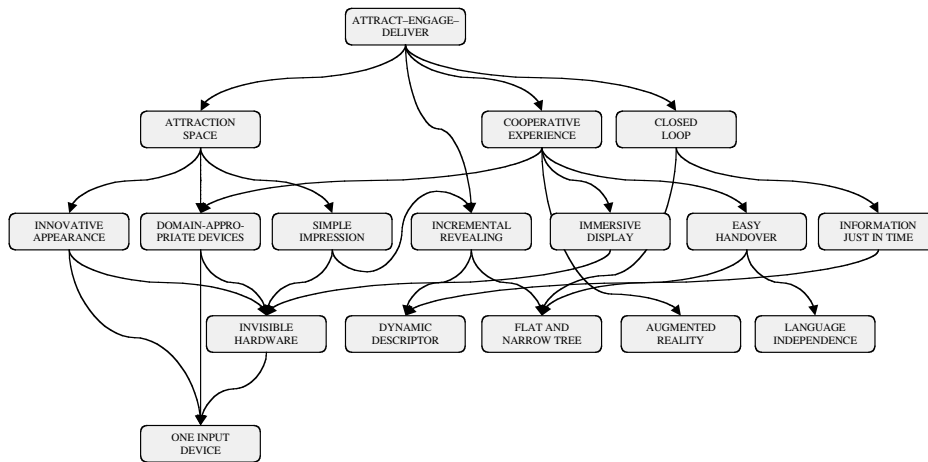


Abbildung 3: Die Entwurfsmustersprache für Mensch-Maschine-Interaktion in computerbasierten Exponaten und verwandten Systemen.

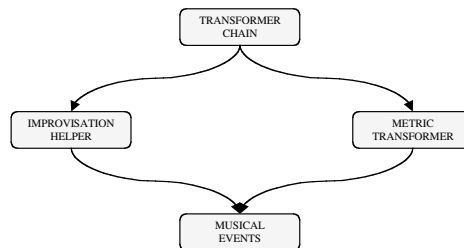


Abbildung 4: Entwurfsmuster zur Programmiertechnik interaktiver Musiksoftware.

## 5 Evaluierung und Werkzeugunterstützung

Ein Vergleich des vorgestellten musterbasierten Entwurfsmodells mit den initialen Anforderungen zeigt, daß alle Anforderungen größtenteils erfüllt werden, wengleich die empirischen Belege vieler Muster noch ergänzt werden können. Ein Arbeitstreffen während der Konferenz *CHI 2000* wählte eines der Muster aus der Sprache des Autors als typisches MMI-Entwurfsmuster. Die Definition von MMI-Entwurfsmustern aus diesem Workshop bestätigt ebenfalls die Ergebnisse dieser Arbeit, und bis auf Autornachweis und Kurzzusammenfassung entsprechen auch die dort definierten Komponenten von MMI-Entwurfsmustern dem Format der vorliegenden Arbeit:

„Ein MMI-Entwurfsmuster fängt das Wesentliche einer erfolgreichen Lösung für ein wiederkehrendes Benutzbarkeitsproblem in interaktiven Sy-

stemen ein.“

Die aus diesem Ansatz resultierenden Systeme beinhalten das interaktive Musikexponat *WorldBeat* [Bor97], das Exponat *Personal Orchestra*, in dem Besucher die Wiener Philharmoniker dirigieren können, und *Virtual Vienna*, einen 3-D-Stadtrundgang durch Wien. *WorldBeat* wurde mit dem Multimedia-Transfer-Preis 1998 ausgezeichnet und lief unter anderem für über drei Jahre als eines der beliebtesten Exponate im Ars Electronica Center Linz. Die beiden anderen Exponate sind permanent im HAUS DER MUSIK WIEN, Wiens größtem musikalischen Ausstellungszentrum, installiert. Daneben wurde die *interaktive Fuge* entwickelt, ein Exponat, das Bachsche Fugenkomposition erläutert und interaktiv ermöglicht [Dan99]. In der Entwicklung insbesondere der neueren Exponate machte sich die Wiederverwendbarkeit der MMI-Entwurfsmuster besonders bezahlt und ermöglichte durch ihre *Vokabularfunktion* rasche Kommunikation nicht nur im Entwurfsteam, sondern auch mit dem Kunden. Darüberhinaus wurden MMI-Entwurfsmuster erfolgreich in der universitären Lehre eingesetzt, und es wurde ein Programm *PET* (Pattern Editing Tool) entworfen, das den rechnergestützten Umgang mit Entwurfsmustersprachen, ihr Erstellen, Beurteilen und Verwenden erleichtert.



Abbildung 5: *WorldBeat*, *Personal Orchestra* und *Virtual Vienna*, drei mit Hilfe des musterbasierten Ansatzes entwickelte interaktive Exponate.

## 6 Zusammenfassung und Ausblick

Um die notwendige interdisziplinäre Kommunikation für den Entwurf intuitiver, erfolgreicher Benutzerschnittstellen zu erleichtern, wurde ein Modell vorgestellt, das *Entwurfsmuster* verwendet, um bewährte Lösungen aus den Gebieten Mensch-Maschine-Interaktion, Programmierertechnik und Anwendungsgebiet eines Softwareprojekts einheitlich zu formulieren. Neben einem Überblick über die Entwicklung von MMI-Entwurfsmustern wurden als neue Beiträge unter anderem eine formale Definition und interdisziplinär einheitliche Struktur von Entwurfsmustern präsentiert. Ein musterbasiertes Entwurfsmodell wurde vorgestellt und in den Software-Entwicklungsprozeß eingebettet. Der Ansatz wurde verwendet, um eine Reihe interaktiver Exponate zu entwickeln, wobei Mustersprachen für



Benutzerschnittstellengestaltung, Programmieretechnik und den Anwendungsbereich „Musik“ entstanden. Neben einer Evaluierung einiger Muster und der entstandenen Systeme wurde der Ansatz auch in Folgeprojekten und der Lehre eingesetzt und bewertet und ein Softwarewerkzeug für die Arbeit mit Entwurfsmustern vorgeschlagen.

Künftige Aufgaben sind vor allem die Erweiterung und weitere empirische Untermauerung der Mustersprachen, die Verwendung des Ansatzes in anderen Anwendungsgebieten und die weitere Entwicklung von Rechnerunterstützung für die Arbeit mit Entwurfsmustersprachen. Daneben wächst derzeit das Interesse in der MMI-Forschungsgemeinde an Entwurfsmustern und Arbeiten wie der vorliegenden; der Autor leitet die HCI Patterns Task Group der IFIP, und die Etablierung eines Gemeinschaftsprojekts zur Schaffung eines Repositoriums von MMI-Entwurfsmustern im Internet wird bereits organisiert. Die englische Dissertation wurde im Mai 2001 im Verlag John Wiley & Sons als Buch veröffentlicht [Bor01]; eine Zusammenfassung findet sich in [Bor00b].

## Literaturverzeichnis

- [ACM92] ACM SIGCHI: Curricula for Human-Computer Interaction. ACM, New York, 1992.
- [AIS<sup>+</sup>77] Alexander, C.; Ishikawa, S.; Silverstein, M.; Jacobson, M.; Fiksdahl-King, I.; Angel, S.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, 1977.
- [Ale79] Alexander, C.: The Timeless Way of Building. Oxford University Press, 1979.
- [Ale96] Alexander, C.: Keynote, OOPSLA'96 11th Annual ACM Conference on Object-Oriented Programming Systems, Languages and Applications (6.–10.10.1996, San Jose, California), 1996. (Konferenzvideo).
- [App92] Apple Computer: Macintosh Human Interface Guidelines. Addison-Wesley, 1992.
- [Bar92] Barsalou, L. W.: Cognitive Psychology: An Overview for Cognitive Scientists. Tutorial Essays in Cognitive Science. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1992.
- [BBC<sup>+</sup>98] Bayle, E.; Bellamy, R.; Casaday, G.; Erickson, T.; Fincher, S.; Grinter, B.; Gross, B.; Lehder, D.; Marmolin, H.; Moore, B.; Potts, C.; Skousen, G.; Thomas, J.: Putting it all together: Towards a pattern language for interaction design. In SIGCHI Bulletin, Bd. 30 (1):(1998), S. 17–23.  
URL <http://www.acm.org/sigchi/bulletin/1998.1/erickson.html>
- [BBL<sup>+</sup>94] Barfield, L.; van Burgsteden, W.; Lanfermeijer, R.; Mulder, B.; Ossewold, J.; Rijken, D.; Wegner, P.: Interaction Design at the Utrecht School of the Arts. In SIGCHI Bulletin, Bd. 26 (3):(1994), S. 49–79.
- [BC87] Beck, K.; Cunningham, W.: Using Pattern Languages for Object-Oriented Programs. Technical Report CR-87-43, Tektronix, Inc., September 17, 1987. Präsentiert auf dem OOPSLA'87 Workshop on Specification and Design for OOP.  
URL <http://c2.com/doc/oopsla87.html>
- [BM98] Borchers, J. O.; Mühlhäuser, M.: Design Patterns for Interactive Musical Systems. In IEEE Multimedia, Bd. 5 (3):(1998), S. 36–46.

- [Bor97] Borchers, J. O.: WorldBeat: Designing a Baton-Based Interface for an Interactive Music Exhibit. In Proceedings of the CHI 97 Conference on Human Factors in Computing Systems (Atlanta, GA, USA, 22.–27.3.1997). ACM, New York, 1997, S. 131–138.
- [Bor00a] Borchers, J. O.: CHI meets PLoP: An Interaction Patterns Workshop. In SIGCHI Bulletin, Bd. 32 (1):(2000), S. 9–12. Bericht zum Workshop auf der ChiliPloP'99 Conf. on Pattern Languages for Programming, Wickenburg, AZ, 16.–19.3.1999.
- [Bor00b] Borchers, J. O.: A Pattern Approach to Interaction Design. In Proceedings of the ACM DIS 2000 International Conference on Designing Interactive Systems (New York, 17.–19.8.2000), 2000, S. 369–378.
- [Bor01] Borchers, J. O.: A Pattern Approach to Interaction Design. John Wiley & Sons, Chichester, UK, 2001.
- [CS95] Coplien, J. O.; Schmidt, D. C., Hg.: Pattern Languages of Program Design, Bd. 1 von Software Patterns Series. Addison-Wesley, 1995.
- [Dan99] Dannenberg, M.: Die Interaktive Fuge: Ein Pattern-basiertes Musikexponat. Diplomarbeit, Universität Ulm, 1999.
- [DD96] Denning, P.; Dargan, P.: Action-Centered Design. In Bringing Design to Software (Winnograd, T., Hg.), Addison-Wesley, Kap. 6, 1996, S. 105–119.
- [GHJV95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.
- [GL99] Granlund, Å.; Lafrenière, D.: A Pattern-Supported Approach to the User Interface Design Process, 1999. Workshop-Bericht, UPA'99 Usability Professionals' Association Conf. (Scottsdale, AZ, 29.6.–2.7.1999).
- [GPB99] Griffiths, R.; Pemberton, L.; Borchers, J.: Usability Pattern Language: Creating a Community. In Proceedings INTERACT'99, vol. 2 (Edinburgh, Scotland, 30.–31.8.1999), 1999, S. 135.
- [Haa99] de Haas, S.: Softwarearchitektur?! Ein Vergleich mit dem Bauwesen. In OBJEKTSpektrum.
- [HFR99] Harrison, N.; Foote, B.; Rohnert, H., Hg.: Pattern Languages of Program Design 4. Software Patterns Series. Addison-Wesley, 1999.
- [HLP97] Helander, M. G.; Landauer, T. K.; Prabhu, P. V., Hg.: Handbook of Human-Computer Interaction. Elsevier Science, Amsterdam, 1997.
- [Kim90] Kim, S.: Interdisciplinary Cooperation. In The Art of Human-Computer Interface Design (Laurel, B., Hg.), Addison-Wesley, 1990, S. 31–44.
- [Lan95] Landauer, T. K.: The Trouble with Computers: Usefulness, Usability, and Productivity. MIT Press, Cambridge, MA, 1995.
- [MD98] Meszaros, G.; Doble, J.: A Pattern Language for Pattern Writing. In Martin et al. [MRB98].
- [MRB98] Martin, R. C.; Riehle, D.; Buschmann, F., Hg.: Pattern Languages of Program Design 3, Bd. 3 von Software Patterns Series. Addison-Wesley, 1998.
- [ND86] Norman, D. A.; Draper, S. W.: User-Centered System Design: New Perspectives on Human-Computer Interaction. Lawrence Erlbaum, Hillsdale, NJ, 1986.

- [Nie93] Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco, 1993.
- [Nor88] Norman, D. A.: The Psychology of Everyday Things. Basic Books, New York, 1988.
- [Tid98] Tidwell, J.: Interaction Design Patterns, 1998. PLoP'98 Conference on Pattern Languages of Programming, Illinois, [http://www.mit.edu/~jtidwell/interaction\\_patterns.html](http://www.mit.edu/~jtidwell/interaction_patterns.html).
- [Tog92] Tognazzini, B.: TOG on Interface. Addison-Wesley, 1992.
- [Vli96] Vlissides, J. M.: Seven Habits of Successful Pattern Writers. In C++ Report, (11/12). <Http://hillside.net/patterns/papers/7habits.html>.