

Herbert A. Meyer, Petra Vogt und Marco Glier

Performance und Usability

1. Einleitung

Das Thema „Performance“ ist für die meisten Computernutzer schon präsent, wenn sie ihren Rechner nur starten. Bis moderne Betriebssysteme hochgefahren sind, kann man nämlich problemlos einen Kaffee kochen. Und auch danach gehören – meist störende – Wartezeiten für viele zum Alltag bei der Computernutzung.

Bruce Tognazzini (1993) berichtet von einer der ersten Konsolen für Videospiele – Channel F von Fairchild aus dem Jahre 1976 –, deren Besonderheit ein erstklassig programmiertes Tic-Tac-Toe-Spiel war. Das Spiel hatte leider einen tragischen Mangel: Unabhängig davon, wie lange ein Spieler für seinen Zug brauchte, reagierte die Maschine innerhalb einer halben Sekunde mit dem nächsten Zug. Weil die Maschine immer den bestmöglichen Zug wählte, fühlten sich die Spieler schnell minderwertig und ließen von dem Spiel ab. Das Zeitverhalten interaktiver Systeme ist also nicht immer zu langsam, wie viele denken, sondern manchmal auch zu schnell. Bei Online-Versionen von Tic-Tac-Toe finden wir diesen Mangel übrigens heute noch (u.a. bei der für Kinder eingerichteten Website „Die Blinde Kuh“).

Die Beispiele deuten an, dass die Performance in vielfältiger Art und Weise Einfluss auf die Nutzungsqualität eines interaktiven Systems nehmen kann. Der Mensch reagiert höchst sensibel auf zeitliche Merkmale der Benutzungsschnittstelle, nicht nur in kognitiver Hinsicht, sondern häufig auch emotional. Ist das Zeitverhalten eines Systems nicht vorhersagbar, werden Benutzer unsicher. Ist es zu langsam oder zu schnell, fühlten sich Benutzer unter- oder überfordert und entwickeln innerlich eine kritische Haltung zum gesamten System.

Insbesondere das Web lieferte in den vergangenen Jahren Anschauungsmaterial für die Performance-Problematik. Der von Jakob Nielsen vor genau einer Dekade geäußerte Traum einer persönlichen T1-Leitung (Nielsen, 1995) ist zwar für einige Benutzer mittlerweile in Erfüllung gegangen, die Problematik wird dadurch jedoch nicht grundsätzlich gelöst. Zur Kopplung von Eingabe und Ausgabe sind in verteilten Systemen immer Transaktionen mit einer Datenübertragung notwendig. Diese Netzlaufzeit führt dazu, dass die von Desktop-Anwendungen gewohnte direkte Manipulation von Objekten mit Antwortzeiten im Bereich von 100 Millisekunden nahezu unmöglich ist. Insbesondere bei komplexen Anwendungen wird das Web den von Desktop-Anwendungen abgeleiteten Ansprüchen nicht gerecht. Unter Performance-Gesichtspunkten befindet sich der moderne Benutzer sozusagen gleichzeitig in zwei verschiedenen Welten: dem Desktop- und dem Web-Paradigma (Müller-Prove, 2005). Das Grundsatzproblem ist keineswegs unlösbar, vielmehr eine Herausforderung. Seit Jahren werden innovative Lösungen diskutiert (vgl. z.B. Garrett, 2005).

Auch jetzt schon kann demonstriert werden, wie einfach Interaktionsprobleme zu lösen sind, die durch eine unangemessene Performance hervorgerufen werden. Beim Online-Tic-Tac-Toe kann das Spiel beispielsweise mit einem adaptiven Mechanismus ausgestattet werden, der die Antwortzeit der Maschine proportional an der Denkzeit des Benutzers ausrichtet. Solche Lö-

sungen gelingen allerdings erst dann, wenn Performance aus ihrem Schattendasein als allein technischer Faktor heraustritt und von Usability Professionals aktiv gestaltet wird.

Der vorliegende Beitrag geht zunächst darauf ein, wie Performance traditionell aus software-technischer Sicht behandelt wird und wie sie beim Usability Engineering unter software-ergonomischen Gesichtspunkten berücksichtigt werden kann. Danach behandeln wir Befunde und Mythen und geben grundlegende Hinweise zur Gestaltung des Zeitverhaltens interaktiver Systeme.

2. Performance als software-technisches Merkmal

Performance gehört aus software-technischer Sicht neben der Usability und weiteren normativen und organisatorischen Randbedingungen zu den so genannten nicht-funktionalen Merkmalen. Diese Merkmale legen den Rahmen fest, in dem die gewünschte Verarbeitungsfunktionalität umgesetzt werden muss. Es geht hier also um die Gestaltung der Form, „wie“ ein Benutzer mit dem interaktiven System zusammenarbeitet. Performance und Usability werden dabei in aller Regel getrennt betrachtet. In dem Qualitätsmodell ISO 9126 beispielsweise ist Performance ein Teilmerkmal der Effizienz eines Softwareprodukts. Im Idealfall wird dieses Teilmerkmal in einer frühen Entwicklungsphase definiert und spezifiziert. In einem solchen Requirements Engineering, das u.a. der Kapazitätsplanung dient, wird genau festgelegt, in welchem Bereich sich die Performance unter festgelegten Bedingungen bewegen sollte. Wenn die Randbedingungen (z.B. Benutzeraufgaben, Hardware, Kosten) bekannt sind, kann die Performance mittels spezieller Verfahren (z.B. Last-, Zeit- oder Stresstests) gegen eine Anforderungsspezifikation getestet werden. Eine typische nicht-funktionale Anforderung ist beispielsweise: „95% aller Transaktionen sollen in nicht mehr als zwei Sekunden verarbeitet und dargestellt werden“.

Es stellt sich einerseits die Frage, ob das skizzierte Vorgehen eingehalten wird, wenn Zeit- und Kostendruck auf das Projektmanagement einwirken und womöglich historisch gewachsene Alt-Anwendungen die Freiheitsgrade beim Anlagenbau einschränken. Andererseits ist nicht offenkundig, auf welcher Grundlage die quantitativen Angaben festgelegt werden. Gibt es empirische Befunde, die kontextunabhängig Gültigkeit beanspruchen, können die für die Spezifikation benötigten exakten Angaben aus ihnen abgeleitet werden. Dies ist bei der Festlegung des Antwortzeitverhaltens bei der Direktmanipulation der Fall. Hier ist hinreichend empirisch gesichert, dass schnelle Antwortzeiten zwischen 20 und 100 Millisekunden eine elementare Voraussetzung für das Gelingen einer dynamischen Interaktion sind (vgl. Shneiderman und Plaisant, 2004). Doch wie wird vorgegangen, wenn keine abgesicherten Werte zur Verfügung stehen? Und: Welche Antwortzeitcharakteristik entsteht, wenn kein systematisches „Performance Engineering“ (Scholz und Schmietendorf, 2000) betrieben wird, sondern „Performance Tuning“ im nach hinein?

Insgesamt ist nicht auszuschließen, dass in software-technischer Hinsicht häufig ein Vorgehen angewendet wird, das Teal und Rudnicky (1992) „guesstimate“ nennen. Zukünftige Benutzer müssen sich dann beim Umgang mit den Softwaresystemen unter Umständen an ein relativ willkürlich festgelegtes Zeitverhalten gewöhnen. Auffällig ist, dass diese Gewöhnung in aller Regel gelingt. Wie sich im Laufe der noch jungen Geschichte der Mensch-Computer Interaktion gezeigt hat, sind Benutzer, wenn sie keine andere Wahl haben, in der Lage, ihr Handeln flexibel auf unterschiedliche Antwortschnelligkeiten einzustellen. Die Anpassung des Menschen an die Technik ist unter software-ergonomischen Gesichtspunkten allerdings nicht vertretbar.

3. Performance als Merkmal der Usability

In der Software-Ergonomie hat sich Usability als Leitkonzept für die Nutzungsqualität eines interaktiven Systems durchgesetzt. Maßgeblich ist heute das in der ISO 9241-Serie dokumentierte Qualitätsmodell. Usability ist hier das Ausmaß, in dem Benutzer effektiv, effizient und zufriedenstellend ihre Ziele erreichen. Kennzeichnend ist, dass die Usability eines interaktiven Systems davon abhängig gemacht wird, in welchem Kontext das System genutzt wird. Performance wird in der Normenserie nicht ausdrücklich berücksichtigt. Zur Sprache kommt sie allerdings bei den Empfehlungen zum Gestaltungsgrundsatz „Erwartungskonformität“ und als Systemvoraussetzung für die direkte Manipulation. Da davon auszugehen ist, dass sich die Performance eines Systems sowohl auf die Effektivität und Effizienz, als auch auf die Zufriedenstellung der Benutzer auswirken kann, wäre es wünschenswert, wenn sie in der Normenserie deutlicher berücksichtigt wird. Auch könnte die Gestaltung der Performance allein unter dem Gesichtspunkt der Erwartungskonformität zu Schwierigkeiten bei der Einführung ergonomischer Verbesserungen führen.

Wie kann nun eine in software-ergonomischer Hinsicht akzeptable Festlegung der Performance erfolgen, ohne auf das im vorangegangenen Abschnitt genannte „guesstimate“-Vorgehen zurückzugreifen? Ein aktueller Forschungsbericht von Geis, Dzida und Redtenbacher (2004) mit Vorschlägen zur Revision der ISO 9241-Serie gibt dazu Auskunft: Gibt es bei bestimmten Interaktionsformen empirische Evidenz dafür, dass fast alle Benutzer in fast allen Situationen ein bestimmtes Antwortzeitverhalten benötigen, kann natürlich auch aus software-ergonomischer Sicht darauf verzichtet werden, den Nutzungskontext zu berücksichtigen. Eine bestimmte Performance kann in diesem Fall direkt als Produktmerkmal festgeschrieben werden (Beispiel Direktmanipulation). Gibt es hingegen keine allgemeingültige und quantitativ exakte Grundlage, und dieses ist bei anspruchsvollen kognitiven Tätigkeiten zu erwarten (s. Ausführungen zum Stand der Forschung, Abschnitt 4.1), muss die Performance auf den konkreten Nutzungskontext eingestellt werden. Eine Anforderungsspezifikation lautet dann beispielsweise wie folgt: „Ein Benutzer soll nicht auf die Verarbeitung der Eingabe und Darstellung der Ausgabe warten“.

Aus software-technischer Sicht ist diese Formulierung ungewöhnlich vage und auf den ersten Blick nicht zu akzeptieren. Auf den zweiten Blick wird eine gemeinsame Sicht – und ein gemeinsames Vorgehen – von Requirements Engineering und Usability Engineering möglich, denn auch aus software-ergonomischer Sicht muss die allgemeine Formulierung operationalisiert und damit quantifiziert werden. Geis et al. (2004) schlagen dazu eine Vorgehensweise vor, die als „Usability Requirements Engineering“ bezeichnet werden könnte. Verfahren der Wahl, um Anforderungen an die Usability festzustellen, ist u.a. die teilnehmende Beobachtung. Wie dieses Verfahren im Hinblick auf die Performance-Problematik genau angewendet werden kann und ob mit diesem Verfahren zuverlässig eine bessere interaktive Qualität erreicht wird als über das „guesstimate“-Vorgehen ist eine offene Frage, die zu untersuchen ist. Die im Qualitätsmanagement vorgesehenen Verfahren zur Überprüfung der Performance (Lasttests, Zeittests, Stresstests u.a.) sollten nach jedenfalls erst dann durchgeführt werden, wenn die Anforderungen an die Performance sowohl nutzungsangemessen als auch quantitativ formuliert wurden.

4. Befunde, Mythen und Empfehlungen

4.1 Befundlage zur Antwortzeitproblematik

Aus den Forschungsbemühungen im Bereich der Mensch-Computer-Interaktion und insbesondere der Software-Ergonomie ist bekannt, dass Benutzer sensibel auf das Zeitverhalten von Computersystemen reagieren und ihr eigenes Verhalten darauf einstellen – sowohl kognitiv als auch emotional (Boucsein, 1987; Holling, 1989; Meyer und Hildebrandt, 2002; Miller, 1968; Shneiderman und Plaisant, 2004). Die menschliche Anpassungsfähigkeit ist in dieser Hinsicht sehr groß. Schnell bilden sich jedoch Erwartungshaltungen heraus, die die Wahrnehmung der Anwendung und den Umgang mit ihr deutlich beeinflussen.

Die Auswirkungen der technischen Performance auf das menschliche Verhalten sind bislang nur unzureichend erforscht. Es wurden Einzelbefunde erhoben, die selten aufeinander bezogen sind. Zudem ist die methodische Güte der Untersuchungen nicht als sehr hoch einzuschätzen und die erzielten Resultate sind kaum generalisierbar (vgl. Holling, 1989). Es mangelt vor allem an theoretischen Konzepten, die als roter Faden zum Verständnis der Antwortzeitproblematik beitragen könnten und Anhaltspunkte für systematische Untersuchungen liefern. Ben Shneiderman beklagte die Theoriearmut bereits 1984 in dem ersten Überblicksartikel zum Thema Systemantwortzeit (Shneiderman, 1984). Bezeichnenderweise wiederholt er die Klage in der Neuauflage seines Standardwerks „Designing the User Interface: Strategies for Effective Human-Computer Interaction“ (Shneiderman und Plaisant, 2004). 20 Jahre sind vergangen, ohne dass es einen Erkenntnisfortschritt gegeben hätte.

4.2 Mythen

„Die unendlich schnelle Maschine“

Software-Techniker neigen bei der Gestaltung von Software-Produkten dazu, die Performance als Qualitätsmerkmal zu unterschätzen. Alan Dix zeigte bereits 1987, wie sich der Irrglaube, dass ein Computersystem immer von sich heraus schnell genug reagieren wird, stillschweigend in das Software Engineering eingeschlichen hat (Dix, 1987). Dieser Mythos existiert auch und gerade unter Netzbedingungen. So gibt es viele Anwendungen mit einer faszinierenden grafischen Web-Oberfläche, deren Benutzer aufgrund einer schwachen Performance bereits nach kurzer Zeit das Interesse verlieren. Der von Raskin (2000) beklagte Zustand, dass Anwendungen beim Starten viel zu lange brauchen und dadurch Benutzer in ihrer Interaktion schwerwiegend behindern, ist vermutlich auch ein Ausläufer des Glaubens an die unendlich schnelle Maschine.

„Je schneller, desto besser“

Die Maxime „Je schneller, desto besser“ wird spätestens seit Einführung der Direktmanipulation regelmäßig geäußert (z.B. Smith, 1983). Sie geht unter anderem auf Untersuchungen zurück, die bis Mitte der 1980er Jahre in IBM-Forschungszentren durchgeführt wurden. Es sollte nachgewiesen werden, dass schnelle Antwortzeiten der Schlüssel zu mehr Produktivität bei der Arbeit mit Timesharing-Systemen sind (Doherty und Thadhani, 1982). Der kommerzielle Erfolg der im Anschluss an diese Untersuchungen erfolgten Umstellung der Systemperformance spricht für die Plausibilität der Annahme. Andere Studien kommen allerdings zu dem Ergebnis, dass es keinen linearen Zusammenhang zwischen Performance und Produktivität gibt (z.B. Barber und Lucas, 1983).

Die für die Direktmanipulation berechnete Forderung von sehr schnellen Antwortzeiten kann nicht verallgemeinert werden. Nach Hüttner, Wandke und Rätz (1995) ist es sogar der häufigste Fehler bei der Gestaltung des Zeitverhaltens interaktiver Systeme, sie in jedem Fall auf das technische Minimum zu reduzieren. „Der dadurch erzielte sehr schnelle Ablauf führt zwar meist momentan zu einer hohen Zufriedenheit mit dem System, jedoch haben Untersuchungen gezeigt, dass insbesondere bei sehr kurzen, konstanten Antwortzeiten die Beanspruchung des ungeübten Benutzers sich langfristig deutlich erhöht. Dies äußert sich in einer messbaren Zunahme der körperlichen Beschwerden und in einem Anstieg der Fehlerrate“ (Hüttner et al., 1995, ohne Paginierung; vgl. Boucsein, 1987). Gegen „Je schneller, desto besser“ sprechen noch weitere Sachverhalte. Beispielsweise scheint es so zu sein, dass längere Antwortzeiten akzeptiert und sogar erwartet werden, wenn ihnen komplexe Eingaben vorausgehen. Des Weiteren kann eine verzögerte Ausgabe durchaus produktiv genutzt werden, wenn sie zur Vorbereitung der nächsten Eingabe dienlich ist.

4.3 Empfehlungen

Aufgrund der lückenhaften Forschungsbasis können gesicherte Empfehlungen nur auf einem recht allgemeinen Niveau gegeben werden. Darüber hinaus ist zu beachten, dass konkretere Leitlinien nur für definierte Kontexte gegeben werden können und es z.B. derzeit noch unterschiedliche Regeln für Web- und Desktop-Anwendungen geben sollte. Die folgenden Empfehlungen sind angelehnt an Hüttner et al. (1995), jedoch in einigen Punkten erweitert. Sie bilden einen Ausgangspunkt für zukünftige Arbeiten in Theorie und Praxis. Die einzelnen Hinweise sind nicht unabhängig voneinander zu verstehen (zur Begründung und Vertiefung sei auf den online zur Verfügung stehenden Text von Hüttner et al. verwiesen).

- Das Zeitverhalten eines interaktiven Systems sollte in einer Anforderungsanalyse spezifiziert werden.
- Die Einhaltung der festgelegten Antwortzeiten sollte systematisch überprüft werden (vor der Übergabe und bei Web-Anwendungen auch im laufenden Betrieb).
- Antwortzeiten sind möglichst kurz zu gestalten, aber es ist darauf zu achten, dass sie für den jeweiligen Anwendungskontext auch nicht zu kurz sind.
- Antwortzeiten sind immer im engen Zusammenhang mit der Anwendung und der Dialogsituation zu gestalten.
- Antwortzeiten sollten nicht zu stark variieren. Die Streuung der Antwortzeiten ist kleiner als die Hälfte des Mittelwertes zu gestalten. Dies ist vor allem für Web-Anwendungen eine Herausforderung.
- Antwortzeiten im Sekundenbereich sind durch eine besondere Anzeige zu kennzeichnen. Bei längeren Pausen (spätestens ab 10 Sekunden) bietet sich eine Information über den Systemzustand an.
- Bei sehr langen Antwortzeiten (spätestens ab 30 Sekunden) sollte die verbleibende Wartezeit analog angezeigt werden. Möglich ist auch eine Ausgabe von Zwischenergebnissen als Vorabinformation bzw. das Ermöglichen der parallelen Bearbeitung von anderen Aufgaben.
- Antwortzeiten und die Art und Weise der Rückmeldung über diese Zeiten sollten für die zukünftige Benutzung anpassbar sein, so dass für ungeübte Anfänger und versierte Benutzer ein unterschiedliches Interaktionsverhalten realisiert werden kann.

Um diese Empfehlungen umsetzen zu können, ist es ratsam, das Zeitverhalten interaktiver Systeme schon während der Entwicklung systematisch zu überwachen. Die Praxiserfahrung

zeigt, dass gerade die fehlende Beachtung der Antwortzeit-Problematik später zu Anwenderbeschwerden führt und dem durch frühzeitige Einbeziehung des Themas vorgebeugt werden kann.

5. Diskussion und Ausblick

Das Zeitverhalten eines interaktiven Systems hat ohne Zweifel Einfluss auf die Form, in der Benutzer mit dem System zusammenarbeiten. Spätestens wenn Performance-Probleme auftreten, wird deutlich, dass die Schnittstelle nicht nur aus statischen Oberflächeneigenschaften besteht, sondern auch dynamische Eigenschaften besitzt. Viele Anhaltspunkte sprechen dafür, dass insbesondere die dynamischen Eigenschaften der Schnittstelle eine wichtige Grundlage dafür sind, dass ein belastungsfreier und Freude vermittelnder Umgang mit computergestützten Systemen möglich wird.

Die „Bindung“ an eine Anwendung geschieht erst dann, wenn die Schnittstelle in den Hintergrund rückt und die Benutzer ihre Ziele mit voller Aufmerksamkeit verfolgen können. Wie kaum ein anderes technisches Merkmal kann eine unangemessene Systemantwortzeit die Aufmerksamkeit auf die Schnittstelle ziehen und von der Sachaufgabe ablenken. Die Interaktion ist dann nicht mehr „nahtlos“ (Ishii und Ullmer, 1997), der Computer „sichtbar“ (Norman, 1998) und das Ideal der „strukturellen Kopplung“ von Mensch und Computer (Bonsiepe, 1996; Winograd und Flores, 1992) wird unterlaufen. Dieser Sachverhalt kann durch eine neue Interpretation der Wortwendung „look and feel“ verdeutlicht werden. Der Begriff „look“ steht dann für die graphische Gestaltung der Schnittstelle, „feel“ hingegen kennzeichnet die interaktiven Aspekte. Während es zur Strukturierung und Kodierung von Bildern und Bildfolgen in verschiedenen Fachgebieten eine lange Forschungstradition gibt, hat die Dimension „feel“ bis heute zu wenig Beachtung gefunden (vgl. Svanæs, 2000). Um dieser „feel“-Dimension Rechnung zu tragen, ist es notwendig, die Übergänge zwischen den Interaktionsschritten in zeitlicher Hinsicht zu gestalten. Hildebrandt, Dix und Meyer (2004) nennen diesen Ansatz „Time Design“.

Zusammenfassend kann festgehalten werden, dass die Berücksichtigung der Performance bei der Gestaltung und Bewertung interaktiver Systeme empfohlen wird, da sie Einfluss darauf nimmt, inwiefern Benutzer ihre Ziele effektiv, effizient und zufriedenstellend erreichen. Auch wenn der Stand der Kunst in Theorie und Praxis zu verbessern ist, ist es für Usability Professionals ein lohnenswertes Unterfangen, sich näher mit Performance zu befassen, da alle Anwendungen, die in verteilten Systemen ausgeführt werden, mit dem Antwortzeitproblem belastet sind und Lösungen dafür gefunden werden müssen.

Literatur

- Barber, R. E.; Lucas, H. C.: System response time, operator productivity, and job satisfaction. *Communications of the ACM* **26** (1983) 972–986.
- Bonsiepe, G.: *Interface. Design neu begreifen*. Berlin: Bollmann, 1996.
- Boucsein, W.: Psychophysiological Investigation of Stress induced by Temporal Factors in Human-Computer Interaction. In: *Psychological Issues of Human Computer Interaction in the Work Place* (Eds. Frese, M.; Ulich, E.; Dzida, W.) Amsterdam: Elsevier, 1987.
- Dix, A.: The Myth of the Infinitely Fast Machine. In: *People and Computers III, Proceedings of Third Conference of the British Computer Society Human-Computer Interaction Specialist Group* (Eds. Diaper, D.; Winder, R.) Cambridge, NY: Cambridge University Press, 1987.
- Doherty, W. J.; Thadhani, A. J.: *The Economic Value of Rapid Response Time* (IBM Technical Report GE20-0752-0), 1982. <http://www.vm.ibm.com/devpages/jelliott/evrrt.html> (Letzter Zugriff: 30.09.2005).

- Garrett, J. J.: *Ajax: A New Approach to Web Applications*.
<http://www.adaptivepath.com/publications/essays/archives/000385.php> (Letzter Zugriff: 30.09.2005).
- Geis, T.; Dzida, W.; Redtenbacher, W.: *Specifying Usability Requirements and Test Criteria for Interactive Systems: Consequences for New Releases of Software-related Standards within the ISO 9241 Series* (Publication Series from the Federal Institute for Occupational Safety and Health, Research Report Fb 1010). Bremerhaven: Wirtschaftsverlag NW, 2004.
- Hildebrandt, M.; Dix, A.; Meyer, H. A.: Time Design. In: *Proceedings of the 2004 Conference on Human Factors in Computing Systems* (Eds. Dykstra-Erickson, E.; Tscheligi, M.) New York, NY: ACM Press, 2004.
- Holling, H.: *Psychische Beanspruchung durch Wartezeiten in der Mensch-Computer Interaktion*. Berlin: Springer, 1989.
- Hüttner, J.; Wandke, H.; Rätz, A.: *Benutzerfreundliche Software. Psychologisches Wissen für die ergonomische Schnittstellengestaltung*. Berlin: Paschke, 1995. <http://www.bmp.de/ISBN/3-929711-06-0/> (Letzter Zugriff: 30.09.2005).
- Ishii, H.; Ullmer, B.: Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In: *Proceedings of the 1997 Conference on Human Factors in Computing Systems* (Ed. Pemberton, S.) New York, NY: ACM Press, 1997.
- Meyer, H. A.; Hildebrandt, M.: Towards Time Design: Pacing of Hypertext Navigation by System Response Times. In: *Proceedings of the 2002 Conference on Human Factors in Computing Systems* (Eds. Terveen, L.; Wixon, D.) New York, NY: ACM Press, 2002.
- Miller, R. B.: Response Time in Man-Computer Conversational Transactions. *Proceedings AFIPS Fall Joint Computer Conference* **33** (1968) 267–277.
- Müller-Prove, M.: *60 Jahre nach Memex: Über die Unvereinbarkeit von Desktop- und Web-Paradigma*. Beitrag zur 35. Jahrestagung der Gesellschaft für Informatik, Bonn, 2005. <http://www.mprove.de/script/05/gi/> (Letzter Zugriff: 30.09.2005).
- Nielsen, J.: *Needing a Personal π Line*. Sidebar to Jakob Nielsen's Alertbox for November 1995. <http://www.useit.com/alertbox/9511.html> (Letzter Zugriff: 30.09.2005).
- Norman, D.: *The Invisible Computer*. Cambridge, MA: MIT Press, 1998.
- Raskin, J.: *The Humane Interface. New Directions for Designing Interactive Systems*. Reading, MA: Addison Wesley, 2000.
- Shneiderman, B.: Response Time and Display Rate in Human Performance with Computers. *ACM Computing Surveys* **16** (1984) 265–285.
- Scholz, A.; Schmietendorf, A.: Aspekte des Performance Engineerings. Aufgaben und Inhalte. In: *Tageband des 1. Workshops Performance Engineering in der Softwareentwicklung* (Hrsg. Dumke, R.; Rautenstrauch, C.; Schmietendorf, A.; Scholz, A.) Darmstadt, 2000. http://www-wi.cs.uni-magdeburg.de/pe2000/papers/scholz_schmiete.pdf (Letzter Zugriff: 30.09.2005).
- Shneiderman, B.; Plaisant, C.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (4th ed.). Reading, MA: Addison Wesley, 2004.
- Smith, D.: Faster is Better: A Business Case for Subsecond Response Time. *Computerworld* **18** (1983) 1–11.
- Svanæs, D.: *Understanding Interactivity: Steps to a Phenomenology of Human-Computer Interaction*. Trondheim: Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, 2000. <http://www.idi.ntnu.no/~dags/interactivity.pdf> (Letzter Zugriff: 30.09.2005).

- Tognazzini, B.: Principles, Techniques, and Ethics of Stage Magic and their Application to Human Interface Design. In: *Proceedings of the 1993 Conference on Human Factors in Computing System* (Eds. Arnold, B.; van der Veer, G; White, T.) New York, NY: ACM Press, 1993. <http://www.asktog.com/papers/magic.html> (Letzter Zugriff: 30.09.2005).
- Teal, S. L.; Rudnicky, A. I.: A Performance Model of System Delay and User Strategy Selection. In: *Proceedings of the 1992 Conference on Human Factors in Computing Systems* (Eds. Bauersfeld, P.; Bennett, J.; Lynch, G.) New York, NY: ACM Press, 1992.
- Winograd, T.; Flores, F.: *Erkenntnis, Maschinen, Verstehen. Zur Neugestaltung von Computersystemen*. Berlin: Rotbuch, 1992.