

## WIKI INSIDE

### Neue Nutzungsformen von Wikis am Beispiel einer communitybasierten Kontexthilfe

VON GUNNAR STEVENS  
UND TORBEN WIEDENHÖFER

#### I. DER ASPEKT DER INTERAKTION BEI DER ANEIGNUNG VON COMPUTERSYSTEMEN

Dieser Beitrag widmet sich dem Thema der Interaktion aus der Perspektive der Aneignung von Computersystemen. Der Begriff der Interaktion lässt sich semantisch dabei in der Nähe des Begriffs der Handlung und der der Kommunikation verorten, jedoch fungiert er unterhalb der Schwelle des Sinnhaften und bietet so die Möglichkeit, Sprache und Handeln auf einer gemeinsamen begrifflichen Basis miteinander zu verklammern. Des Weiteren erlaubt es ein solch voraussetzungsarmer Begriff der Interaktion auch, Mensch und Computer auf der gleichen begrifflichen Ebene anzusiedeln und darüber diverse Mensch-Computer-Konstellation als ein miteinander interagierendes Netzwerk in den Blick zu bekommen. Dies macht den Begriff für die Reflexion über die Frage nach der Aneignung von Computersystemen interessant. Umgekehrt ist die Aneignung von Computersystem für die Analyse der Mensch-Computer-Interaktion deshalb interessant, weil hier nicht von geschlossenen Bedeutungssystemen ausgegangen werden kann, sondern vielmehr die Frage nach der Entwicklung von Bedeutung in der Interaktion im Zentrum des Interesses steht. So erfährt man z.B. gerade in Aneignung eines neuen Computersystems die Widerständigkeit des Artefakts, bei der sich seine Bedeutung erst einmal negativ, als eine Erwartungsenttäuschung, erschließt.<sup>1</sup>

Eine Unterstützung zur Aneignung von Computersystemen wird vom Hersteller meist in Form von Hilfesystemen bereitgestellt.<sup>2</sup> Entsprechend aufschlussreich ist die Analyse der dahinter liegenden Gestaltungskonzepte solcher Systeme, insbesondere weil hier diverse Zuschreibungen bezüglich der Kompetenzen von Computern, Nutzern, sozialen Netzwerken und deren Interaktionen vorgenommen werden. Aus diesem Grunde sollen in diesem Beitrag die verschiedenen Gestaltungskonzepte einer Analyse unterzogen werden. Hierbei wird jedoch keine explizite Definition des Begriffs der Interaktion vorgenommen, vielmehr schwingt in der Perspektive, die in der Analyse eingenommen wird, eine voraussetzungsarme Interpretation des Begriffs unterschwellig mit. Im zweiten Teil des Beitrags wird anschließend das Design eines neuen Hilfesystemansatzes vorge-

---

1 Vgl. Stevens u.a.: „Zum Handeln in Krisensituationen“.

2 Vgl. Stevens u.a.: „Infrastrukturen zur Aneignungsunterstützung“.

stellt, welcher durch eine semio-pragmatische Vorstellung der Erkenntnisentwicklung<sup>3</sup> beeinflusst wurde, bei der Zeichen weder als etwas rein Statisches noch als etwas rein Subjektives verstanden werden, sondern als etwas Dynamisches, das sich in der sozialen Interaktion mit der Welt entwickelt.

## 2. KONZEPTE VON HILFESYSTEMEN: VON BEDIENUNGSANLEITUNGEN UND FREUNDLICHEN ASSISTENTEN

*Dokumentenzentrierte Hilfesysteme* bilden den ältesten Hilfeansatz und kommen in nahezu allen Betriebssystemen oder Softwarepaketen vor. Vor allem in den Anfängen waren Computersystemen gedruckte Bedienungsanleitungen beigelegt, die als lineare Texte verfasst wurden. Diese waren darauf ausgelegt, dass Nutzer sich vor Inbetriebnahme eines technischen Gerätes die Bedienungsanleitung komplett durchlesen, um anschließend zu wissen, wie das Gerät *richtig* zu bedienen ist. In Folge wurden jedoch auch die neuen Möglichkeiten interaktiver Computersysteme aufgegriffen, so dass heutzutage auch Online-Handbücher, Hypertextsysteme oder sich selbst beschreibende Objekte, durch Tooltips und andere Formen von Interpretationshilfen, eine wichtige Rolle spielen.

Bei der Entwicklung dieser neuen Formen interaktiver Hilfesysteme spielten vor allem die empirischen Untersuchungen von John Carroll<sup>4</sup> eine wichtige Rolle, weil er sich insbesondere für die Nutzungspraktiken von Hilfesystemen interessierte. Bei seinen Untersuchungen zeigte sich, dass Nutzer sich anders verhielten, als es die Form der Bedienungsanleitungen nahe legte. Statt durch das Schmökern in der Bedienungsanleitung die richtige Bedienung des Geräts zu erlernen, verwendeten die Nutzer das System und griffen erst bei Problemen auf die beigelegte Information zurück. Das heißt: die der Gestaltung zugrunde gelegte Mensch-Buch-Interaktion entsprach nicht der Wirklichkeit. Aus dieser Erkenntnis wird aber die Frage nach einer geeigneten Gestaltung der Interaktion thematisch. So greift Carroll mit seinem *minimalism*-Ansatz das Interaktionsverhalten der Nutzer auf und versucht diesen durch einen aktionsorientierten Hilfeansatz auf Hilfeartikel zu übertragen. Vor dem Hintergrund des beobachteten Interaktionsverhaltens hat sich auch die Gestaltung von dokumentenzentrierten Hilfesystemen gewandelt.

In Folge haben sich neue Formen der Hilfetexte bzw. deren Einbettung in den Nutzungskontext entwickelt. Diese Formen lassen sich z.B. bei der integrierten Standardhilfe von *Windows XP* erkennen. Shneiderman<sup>5</sup> beschreibt das *Windows*-Hilfesystem als Ansammlung untereinander verlinkter Hilfetexte. Durch Auswahl eines sogenannten *topic* kann der Nutzer aus einer Liste von zugehörigen Hilfeartikeln wählen. Diese Hilfeartikel sind aufgabenorientiert aufgebaut und

---

3 Vgl. Hoffmann: Erkenntnisentwicklung.

4 Vgl. Carroll/Rosson: „Paradox of the Active User“; Carroll u.a.: *The Minimal Manual*.

5 Vgl. Shneiderman: *Designing the User Interface*.

können schrittweise abgearbeitet werden. Wird aber die lineare Form des Textes erst einmal aufgebrochen, ergibt sich fast unmittelbar das Nachfolgeproblem, wie der Text bzw. die Einstiegspunkte in die Textteile zu organisieren sind und welche Interaktionskonzepte zur Erschließung des Textes geeignet sind. So merkt z.B. Shneiderman an, dass die unüberschaubare Menge an Hilfeartikeln gerade für Anfänger zur Herausforderung wird – wobei immer noch davon ausgegangen wird, dass im Prinzip der richtige Hilfeartikel vorhanden ist, die Schwierigkeit aber darin besteht, diesen zu finden. Indirekt spricht Shneiderman das Problem an, das bei der Gestaltung der Interaktion im Produktionskontext von der konkreten Problemsituation *dekontextualisiert* werden muss, was umkehrseitig zur Notwendigkeit der *Rekontextualisierung* im Nutzungskontext einhergeht<sup>6</sup>.

Das Phänomen der *Dekontextualisierung* soll uns im Folgenden als Referenzpunkt dienen, um hierüber die verschiedenen Hilfesystemansätze als idealtypisch zu bestimmen. Denn um die beobachteten Phänomene der Mensch-Computer-Interaktion genauer zu verstehen, genügt es nicht, nur die Interaktion in der Nutzungspraxis zu bestimmen. Vielmehr gilt es auch zu verstehen, wie die Nutzungspraxis sich aus dem Blickwinkel der Produktionspraxis darstellt, bei der die Nutzungspraxis nicht konkret, sondern nur abstrakt gegeben ist. Das Phänomen der *Dekontextualisierung* stellt dabei einen geeigneten Referenzpunkt dar, um die in den verschiedenen Ansätzen anlegte Gestaltung der Interaktionen zwischen Herstellern, Computern und Nutzern zu bestimmen.

Innerhalb der Hilfesystemansätze lassen sich unseres Erachtens drei Idealtypen identifizieren:

- Wissenswerkzeuge und Wissensarbeiter,
- Avatare, Agenten und andere nützliche Helfer,
- Telesupport mittels Computer vermittelter Kommunikation.

### 3. VON WISSENSWERKZEUGEN UND WISSENSARBEITERN

Ein möglicher Umgang mit *Dekontextualisierung* besteht darin, nicht nur eine Menge von Hilfetexten auszuliefern, sondern auch Werkzeuge aus dem Bereich des Information Retrieval in das Metaartefakt Hilfesystem zu integrieren (vgl. Abb. 1). Damit soll der Nutzer in die Lage versetzt werden, das in den Hilfetexten vorhandene Wissen zu erschließen. Dadurch tritt die traditionelle Bedienungsanleitung dem Nutzer nicht mehr als ein einfacher linearer Gebrauchstext, sondern als ein interaktives Gebrauchswerkzeug gegenüber.

---

6 Vgl. Rolf: „Von der Theoriearbeit der Informatik zur Gestaltung“.

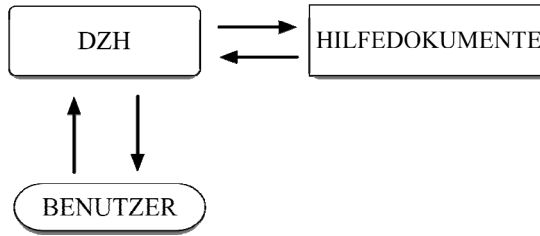


Abb. 1: Schematische Darstellung eines dokumentenzentrierten Hilfesystems (DZH) und die Rolle des Nutzers.

Im Verhältnis Hersteller – Computer – Nutzer wird dabei dem Nutzer eine neue zusätzliche Kompetenz zugemutet. Neben der Kompetenz, Hilfetexte zu lesen und in Hinblick auf sein Problem richtig zu interpretieren, wird ihm auch die Kompetenz zugeschrieben bzw. übertragen, mit dem Werkzeug Hilfesystem richtig umzugehen, um die richtigen Texte zu finden. Das heißt: hinsichtlich der Arbeitsteilung übernimmt der Nutzer die Position des *Wissensarbeiters*, wie umkehrseitig die Softwaretechnik für die Bereitstellung geeigneter *Wissenswerkzeuge* verantwortlich ist. Die Rolle des Herstellers liegt in der *Befüllung der Hilfesystem-Datenbank* mit einer geeigneten Menge von Texten.



Abb. 2: Hilfesystem mit integriertem Werkzeug, den richtigen Hilfetext zu finden.

An der Gestaltung der integrierten Hilfe von Microsoft XP (vgl. Abb. 2) lassen sich verschiedene Strategien des Zugangs erkennen:

- *Index*  
Es gibt eine alphabetisch sortierte Liste von Themengebieten. Diese Art der Organisation bemächtigt sich zum einen der allgemeinen Kulturtechnik, die Reihenfolge des Alphabets zu kennen, zum anderen geht es von der Heuristik aus, dass die Hilfesystemredakteure das Problem des Nutzers unter den richtigen Begriff verschlagwortet haben.
- *Verweise*  
Mittels Hyperlink-Struktur kann der Nutzer seine Kompetenz zur Assoziation nutzen, um sich zum richtigen Artikel zu klicken.
- *Themengliederung*  
Eine hierarchische Gliederung von Themen mit direkten Verweisen auf Hilfeartikel. Diese Art der Organisation beruht quasi auf der gegensätzlichen Heuristik, dass der Nutzer sein Problem einen der dargebotenen Themen zuordnen kann und so durch immer genauere Themengliederungen sukzessive zu dem passenden Hilfetext kommt.
- *Suche*  
Anhand von Schlüsselwörtern kann in allen Hilfeartikeln eine Volltextsuche ausgeführt werden. Hierbei muss der Nutzer die Kompetenz besitzen, auf seine Problemsituation treffende Schlüsselwörter abzuleiten, die ein Autor in einem Hilfeartikel verwendet haben könnte, welcher eine Lösung für sein Problem beschreibt.
- *Antwortassistent*  
Der Antwortassistent erlaubt es, natürlichsprachig formulierte Fragen zu stellen. Dabei filtert der Assistent alle relevanten Schlüsselwörter heraus und bietet eine Liste mit passenden Themen dem Hilfesuchenden an. Diese ist in drei Kategorien unterteilt: ‚Wie mache ich ...‘, ‚Erkläre mir ...‘ und ‚Programmier- und Sprachreferenz‘. Im Wort *Assistent* wird schon eine Verlagerung der Kompetenzzuschreibung – weg vom Nutzer hin zur Maschine – sichtbar. Insofern hätten wir den Antwortassistenten auch in die nächste Kategorie einordnen können. Hieran erkennt man, dass die idealtypische Kategorisierung primär dazu dient, unsere Analyse zu strukturieren, und dass sie nicht die Differenziertheit der Praxis wiedergeben kann.

#### 4. AVATARE, AGENTEN UND ANDERE NÜTZLICHE ANTHROPOMORPHE HILFESYSTEME

In der KI-nahen Forschung wird der Technik die Rolle bzw. die Kompetenz eines weisen *Agenten* zugeschrieben. Wie im Begriff *Avatar* konnotiert,<sup>7</sup> besitzt dieser übermenschliche Kräfte, die in den Dienst der Menschheit gestellt werden. Je

---

7 Avatar (v. Sanskrit Avatara, „Herabkunft“) bezeichnet: die körperliche Manifestation eines Gottes, etwa in Menschen- oder Tiergestalt, vgl. <http://de.wikipedia.org/wiki/Avatar%28Internet%29>, 23.3.2007.

nach dem weltanschaulichen Vorzeichen wird der Nutzer dabei von lästiger Arbeit befreit oder – unter kulturpessimistischen Vorzeichen – in seiner Unmündigkeit gefangen gehalten.<sup>8</sup> Bei den Hilfesystemen unterscheiden Preece u.a. und Dix u.a.<sup>9</sup> drei Arten intelligenter Systeme, die auf die Erfassung und Interpretation von Nutzungsdaten abzielen:

- *Kontextsensitive Hilfe*  
Der Hauptgedanke bei kontextsensitiven Hilfesystemen ist, den Kontext des Nutzers zu beobachten und zu deuten, um ihm die Hilfe anzubieten, die aus dessen Arbeitskontext abgeleitet wurde. Diese Möglichkeit stellt einen der wichtigsten, wenn auch einen der schwierigsten Schritte in der Entwicklung von Hilfesystemen dar. Die Risiken liegen dabei zum einen in der Bestimmung der zu erfassenden Informationen (Art, Umfang, Zeitraum, Priorität etc.) und in der Interpretation der erfassten Daten. Z.B. können die unterschiedlichen psychischen Situationen des Nutzers zu Fehlinterpretationen führen. Der einfachste Weg, den Kontext zu erfassen, ist, die Position des Cursors zu beobachten und Hilfe zu den unter dem Cursor befindlichen Objekten zu liefern.
- *Anpassbare Hilfe*  
Anpassbare Hilfe stellt ebenso wie die aktive Hilfe eine Art der kontextsensitiven Hilfe dar. Dabei wird durch das System der Hilfeinhalt an die Interessen des Nutzers angepasst. Die Interessen wurden vorher entweder manuell durch den Nutzer eingegeben oder durch Interpretation der Nutzungsdaten bestimmt. Dabei wird z.B. auf Grund von vorab spezifizierten Nutzermodellen eine Klassifikation in Nutzertypen vorgenommen. Beispielsweise würde ein solches System die Hilfe an den Expertisegrad des Nutzers anpassen.
- *Aktive Hilfe*  
Die Initiative zum Aufruf passender Hilfe kann durch den Nutzer selbst geschehen, aber auch durch die Anwendung bzw. durch das Hilfesystem selbst erfolgen. Bei der aktiven Hilfe versucht das System die Handlungen des Nutzers auf Grund der zur Verfügung stehenden Informationen zu deuten und dem Nutzer – ohne dessen bewusste Anforderung – Hilfe zur Verfügung zu stellen.

Eine softwaretechnisch affine Interpretation des Konstrukts *Kontext* wurde dabei von Dey u.a. geliefert.<sup>10</sup> Für sie ist ein Kontext „any information that can be used

---

8 Wobei die Unmündigkeit wieder als Motivation herangezogen werden kann, intelligenteren Maschinen zu entwickeln, um der Trägheit der Masse entgegenzuwirken.

9 Vgl. Preece: Human Computer Interaction; Dix u.a.: Human-Computer Interaction.

10 Dey u.a.: „A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications“. Dass die Deutung des Konstrukts ‚Kontext‘ dabei nicht allein von den softwaretechnischen Möglichkeiten, sondern auch von Denkschulen der Wissenschaftler abhängt, hat unter anderen Leonardo Ramirez in seiner Aufarbeitung des Diskurses um kontextbewusstes Rechnen im aktuellen Forschungsfeld des Ubiqui-

to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the applications themselves“<sup>11</sup>.

Das sehr unspezifische *any* markiert dabei aber auch schon die grundsätzlichen Schwierigkeiten bei der Gestaltung kontextbewusster Systeme. Preece u.a. und Dix u.a.<sup>12</sup> fassen dies in einer Menge grundsätzlicher Fragen zusammen:

- Welche Art von Informationen soll gesammelt werden?
- Wie viele Informationen sollen gesammelt werden?
- Welche Teile der Anwendung sollen zusammenhängend betrachtet werden?
- Welche zeitlichen Rahmen sind zu berücksichtigen?
- Etc.

Sind diese Fragen beantwortet, stellt sich schon das nächste Problem, wie aus den gesammelten Daten auf die Problemsituation geschlossen werden kann. Hinzu kommt noch der Punkt, dass Nutzer sich verunsichert fühlen können, wenn klar wird, dass Informationen über sie gesammelt werden, sie aber nicht genau wissen, welche dies sind und wofür sie verwendet werden.<sup>13</sup>

Zusammenfassend lassen sich Konzepte von *kontextbewusster* Hilfe dadurch charakterisieren, dass sie den *Kontext* des Nutzers und der Applikation beobachten und zu verstehen versuchen, um ihm im richtigen Moment Hilfe anzubieten. Damit wird die Lösung des *Dekontextualisierungsproblems* primär im Bereich der Technik verortet.

Die Geschichten rund um den wohl berühmtesten Avatar, Karl Klammer von *Microsoft Word* – der als lästig, unterhaltsam, aber auch von manchen als hilfreich empfunden wird<sup>14</sup> – zeigt dabei die ambivalenten Haltungen gegenüber solchen kontextbewussten Ansätzen.

## 5. TELESUPPORT MITTELS COMPUTERVERMITTELTER KOMMUNIKATION

Als nächsten Idealtypus wollen wir Ansätze des *Telesupport mittels computervermittelter Kommunikation* vorstellen. Bei diesem Idealtypus steht nicht die direkte Bereitstellung von Hilferessourcen durch Computersysteme, sondern die *compu-*

---

tous Computing dargelegt. Vgl. Ramirez: Social Construction of End User Adaptations in Context Awareness Systems.

- 11 Dey u.a.: „A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications“, S. 11.
- 12 Preece u.a.: Human Computer Interaction; Dix u.a.: Human-Computer Interaction.
- 13 Vgl. Herrmann: „Probleme bei der Konstruktion und beim Einsatz von Hilfesystemen“.
- 14 Man lese hierzu auch die Diskussion über das Verschwinden von Karl Klammer unter: [http://blogs.msdn.com/inside\\_office\\_online/archive/2007/01/30/yes-clippy-is-dead.aspx](http://blogs.msdn.com/inside_office_online/archive/2007/01/30/yes-clippy-is-dead.aspx), 22.02.2008.

tervermittelte Mensch-zu-Mensch Interaktion im Vordergrund (vgl. Abb. 3). Hierbei fungiert ein Computersystem als Kommunikationsmedium.

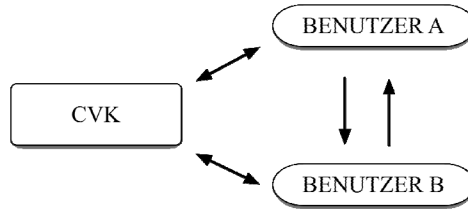


Abb. 3: Die Rolle des Metaartefakt Hilfesystem bei Computer vermittelter Kommunikation (CVK).

Das Hilfesystem kann dabei asynchrone (E-Mail, Newsgroups, Foren, etc.) oder synchrone Kommunikationstechniken wie *instant messaging* bereitstellen, um zwischen dem Nutzer und dem professionellen Hilfeanbieter zu vermitteln.<sup>15</sup> Fragen können z.B. in E-Mails geschrieben werden und diese an einen oder mehrere Experten versandt werden. Somit weisen kommunikationsorientierte Hilfesysteme den Vorteil auf, dass Individuen innerhalb kürzester Zeit den persönlichen Kontakt zu einem oder auch mehreren Hilfeanbietern aufbauen können. Dadurch, dass die Akteure im Dialog treten können, besteht zudem die Chance, dass nicht nur der Computerlaie vom Fachmann lernt. Der Fachmann kann so auch Feedback über Probleme in der Praxis bekommen und etwas über Erfolg bzw. Misserfolg der Produktgestaltung erfahren. Durch das Feedback kann so der Hersteller sein Wissen über praxismgerechte Gestaltung erweitern und in den Entwicklungsprozess einfließen lassen.

Covi/Ackerman<sup>16</sup> stellen dabei fest, dass kommunikationsorientierte Hilfesysteme zu den erfolgreichen und populären Techniken gehören, nicht zuletzt bei unvorhersehbaren oder mehrdeutigen Problemsituationen. Die Probleme dieses Ansatzes sind jedoch in den hohen Bereitstellungskosten zu verorten. Neben zusätzlich bereitgestellten Fachkräften müssen auch die Hilfsbereitschaft und die Kompetenz von Experten vorhanden sein, genervten Computernutzern zum wiederholten Male geduldig das System zu erklären.

## 6. PRODUKTIONSBEDINGTE SCHWACHSTELLEN UND NUTZUNGSPRAKTIKEN HERSTELLER ZENTRIERTER HILFESYSTEME

Im vorigen Abschnitt haben wir verschiedene Formen moderner Hilfesysteme vorgestellt, die sich vor dem Hintergrund der Komplexität heutiger Anwendungs-

15 Vgl. Covi/Ackerman: „Such easy-to-use systems“.

16 Ebd.



systeme entwickelt haben, um so den Anstieg des Lernaufwandes des ‚richtigen‘ Gebrauchs zu verringern. Dabei haben wir die Ansätze als Lösungsstrategie auf das Problem interpretiert, dass Bedienungsanleitungen zu mehrere hundert Seiten dicken Schwarten angewachsen sind, die von Nutzern nicht auf Vorrat gelesen, sondern vor dem Hintergrund praktischer Probleme zu Rate gezogen werden.

Hieraus erwächst das allgemeine Problem der *Dekontextualisierung*, das wir an dieser Stelle weiter in das Problem der *situativen Dekontextualisierung* und der *kulturellen Dekontextualisierung* aufspalten wollen. Mit *situativer Dekontextualisierung* ist gemeint, dass das Hilfesystem samt der Hilfetexte vor Auslieferung der Anwendung erstellt werden muss und somit die oben genannte Schwierigkeit aufwirft, Problemsituationen künftiger Nutzer mit den komplexen Systemen voraussagen zu können und Aneignungsphänomene im Vorhinein zu antizipieren<sup>17</sup>. Hinsichtlich der *situativen Dekontextualisierung* hat insbesondere die ethnomethodologisch orientierte Arbeit von Suchman gezeigt, dass auf Grund der Situiertheit menschlichen Handelns KI-orientierte Hilfesysteme prinzipielle Schwierigkeiten haben, den Nutzungskontext zu erfassen.<sup>18</sup> In dem von ihr untersuchten Fall – die Bedienung eines neuen Kopierertyps mit integriertem intelligentem Hilfesystem – zeigte sich, dass die von den Entwicklern vorgenommene und durch den Algorithmus umgesetzte Antizipation im konkreten Nutzungskontext zu unpassenden bzw. verwirrenden Hinweisen des Hilfesystems führte. Dies hatte zur Folge, dass die Nutzer *workarounds* angewandt haben, um an die richtige Antwort auf ihre Fragen zu kommen.

Die Versprechungen der künstlichen Intelligenz, dem Problem durch immer exaktere Vermessung des Kontexts mittels intelligenteren Algorithmen und einer immer umfangreicheren Datensammlung Herr zu werden, klingen wie der Wunsch der Wirkungsforschung, durch immer exaktere Messung detaillierte Nutzermodelle zu erhalten. Die kritischen Anmerkungen von Ien Ang in Bezug auf die Wirkungsforschung lassen sich dabei fast eins zu eins auf den Bereich der Technikgestaltung übertragen:

Die Paradoxie des elektronischen Zuschauerzählers liegt aber darin, dass er den Wunsch ausdrückt, einer detaillierten Landkarte der sozialen Welt tatsächlicher Sehgewohnheiten. Diese fortschreitende Annäherung von Repräsentationsstrategien an das Soziale muss, wie ich meine, letztlich das Chaos statt Ordnung enthüllen.<sup>19</sup>

Mit der *kulturellen Dekontextualisierung* ist gemeint, dass die Ersteller von Bedienungsanleitungen meist einen anderen kulturellen Hintergrund haben als die Nutzer des Systems. In Zeiten der Globalisierung bezieht sich dies auch auf geographisch unterschiedliche Kulturräume, was man manchmal an ins Deutsche über-

17 Vgl. Suchman: *Plans and Situated Actions*; Stevens u.a.: „Breaking it up“.

18 Vgl. Suchman: *Plans and Situated Actions*.

19 Ang: „Im Reich der Unsicherheit“.

setzten Bedienungsanleitungen chinesischer oder schwedischer Herkunft erkennen kann.

Zudem, und hierauf wollen wir genauer eingehen, macht sich die soziokulturelle Kluft auch im technischen Duktus der Bedienungsanleitungen bemerkbar. Sie stellt zwar dem Nutzer eine Vielzahl von Informationen bereit, die die Nutzer aber größtenteils nicht mit ihrem Arbeitskontext in Verbindung bringen können. Zur Illustrierung dieser Art von *kultureller Dekontextualisierung* wollen wir auf die Arbeiten von Hinds und Pfeffer<sup>20</sup> aus dem Bereich des organisatorischen Lernens zurückgreifen. Sie haben den Wissensaustausch zwischen Experten und Novizen in Organisationen untersucht. Hinds und Pfeffer<sup>21</sup> fassen den Wissensaustausch zwischen Experten und Novizen primär als kognitives Problem auf. Bezug nehmend auf Sternberg<sup>22</sup> gehen sie davon aus, dass Experten tendenziell zu abstrakterem und vereinfachtem Wissen neigen. Beim Wissensaustausch führt die Vereinfachung der Aufgabe dazu, dass Experten weniger den Schwerpunkt auf Details legen und somit weniger spezifizierte Schritte in der Aufgabenbearbeitung besitzen. Dieses hingegen benötigen Personen mit geringerem Erfahrungsschatz, um sich so durch Details und Hintergrundwissen ihr Verständnis zu bilden. Auf Grund verschiedener Experimente kommen Hinds und Pfeffer dabei zu dem Schluss, dass es sich bei den Experten um kognitive Barrieren (*cognitive limitations*) handelt, die sie daran hindern, ausreichend konkrete und detaillierte Erläuterungen zu geben, „even when they know their explanations are intended for novices“<sup>23</sup>.

Ein weiter Umstand, der im wissenschaftlichen Diskurs meist übersehen wird, ist, dass heutige Softwareprojekte durch sehr eng kalkulierte zeitliche und finanzielle Rahmenbedingungen geprägt sind. Deswegen kann am Projektende die nötige Auseinandersetzung mit der Hilfe und dem Schreiben geeigneter Hilfetexte nur beschränkt oder mitunter gar nicht wahrgenommen werden.

Nachdem wir einige Aspekte der Produktion von Hilfesystemen dargelegt haben, wollen wir die Nutzungspraktiken heutiger Hilfesysteme betrachten. Hierzu sollen einige Ergebnisse einer Online-Umfrage über das ‚Rezeptionsverhalten‘ der Nutzer vorgestellt werden. Eine genauere Darstellung der Umfrage findet man bei Wiedenhöfer.<sup>24</sup> Bei der Umfrage ging es darum, den Umgang mit den integrierten Standardhilfen von Softwareprodukten zu untersuchen. Hierzu wurden per E-Mail Personen aus unterschiedlichen Bereichen angeschrieben. Diese setzten sich größtenteils aus Studenten unterschiedlicher Fachrichtungen, Verwaltungsangestellten und Mitarbeitern produzierender Unternehmen zusammen. Wir bekamen von 58 Personen eine Antwort, was einer Rücklaufquote grö-

---

20 Pfeffer/Hinds: „Why Organizations Don’t ‚Know What They Know“.

21 Ebd.

22 Sternberg: „Cognitive Conception of Expertise“.

23 Pfeffer/Hinds: „Why Organizations Don’t ‚Know What They Know“., S. 3.

24 Vgl. Wiedenhöfer: Help in Context.

ber Eins entsprach. Dies hing damit zusammen, dass die angeschriebenen Personen die Umfrage an Bekannte und Freunde weiterleiteten. Die Befragung, die per E-Mail an die Befragten versendet wurde, erhebt keinen Anspruch auf Repräsentativität im statistischen Sinne. Trotzdem gibt sie doch ein interessantes Bild über die (Nicht-)Nutzung von Standardhilfen ab und illustriert in schöner Weise, was mit *Dekontextualisierung* in der Praxis gemeint ist.

Die Untersuchung zeigte, dass 74% der Probanden Probleme im Umgang mit einem Softwareprodukt innerhalb des letzten Monats hatten. Im Allgemeinen zogen aber 86% der Nutzer die integrierte Standardhilfe nur selten oder gar nicht zur Problemlösung heran. Diese Ergebnisse untermauern die Tatsache, dass Anwender die Nutzung traditioneller Hilfesysteme meiden, obwohl die Notwendigkeit nach Unterstützung besteht und nach Hilfe gesucht wird. Diese Abneigung wird dadurch forciert, dass 62% der Nutzer angeben, in nur geringem Maße passende Hilfe durch das integrierte Hilfesystem zu erhalten. 71% beklagen zudem den hohen Aufwand, Hilfeartikel in Bezug auf deren Problemkontext zu finden. Darüber hinaus zeigt sich eine Unzufriedenheit bei 68% der Personen über den Inhalt dieser Hilfeartikel. Die *kulturelle Dekontextualisierung* spiegelt sich ebenfalls in der Studie wider. 58% der Probanden gaben an, Verständnisprobleme mit den Hilfetexten zu haben, die z.B. mit unverständlichem Vokabular einhergehen.

Diese negativen Erfahrungen mit traditionellen Hilfesystemen führten bei den Befragten dazu, andere alternative Hilferessourcen zur Problemlösung heranzuziehen. 84% der Befragten bevorzugten, Freunde oder Bekannte zu Rate zu ziehen. Dies stellt dabei die am häufigsten genutzte Hilfe dar. Das Internet und speziell Suchmaschinen werden von ca. 61% zur Hilfesuiche genutzt. 92% vertrauten dabei eher Hilfetexten, die nicht vom Hersteller, sondern von Nutzern selbst verfasst worden sind. Dies zeigt sich z.B. auch bei der Suche nach thematisch passenden Diskussionsforen, in denen Nutzer ihr Wissen austauschen. Gedruckte Fachliteratur oder der Supportdienst werden hingegen von ca. 80% gemieden.

Vor den Schwierigkeiten bei der Produktion und den beobachteten Nutzungspraktiken kann es auch für die Hersteller von Anwendungssystemen attraktiv sein, sich auf das *Reich der Ungewissheit*<sup>25</sup> einzulassen und die Entwicklung und Pflege von Hilfesystemen ganz oder teilweise an die Nutzercommunity outzusourcen.

## 7. COMMUNITY HELP – ZWISCHEN MONTESSORIPÄDAGOGIK UND TOMSAWYER-PRINZIP

Ein Beispiel für diesen Trend, den Support für ein Produkt an die Nutzercommunity auszulagern, findet sich z.B. in den aktuellen *Office-2003*-Produkten von *Microsoft*. Hier wurde das integrierte Hilfesystem an ein von *Microsoft* bereitgestelltes Communitysystem gekoppelt, mittels dessen die Nutzer ihr Wissen aus-

---

25 Ang: „Im Reich der Unsicherheit“.

tauschen und Fragen zur Bedienung stellen können. Neben der traditionellen Suchfunktion nach Hilfeartikeln, die von *Microsoft* bereitgestellt werden, wird dem Nutzer ebenfalls die Möglichkeit gegeben, in Foren der *Word-Community* nach passender Hilfe zu suchen. War die Suche in Foren und in der traditionellen Hilfe von *Word* erfolglos, kann der Nutzer sein Problem an die *Community* formulieren (vgl. Abb. 4). Dadurch integriert die *Word-Hilfe* das Feature, in konkreten Problemsituationen auf das Wissen der *Community* direkt aus dem integrierten Hilfesystem zuzugreifen. Dies hat den netten Nebeneffekt, dass die auf dem *Communitysystem* geführten Diskurse zwischen den Nutzern zu einer Befüllung der Datenbank des Hilfesystems führen.

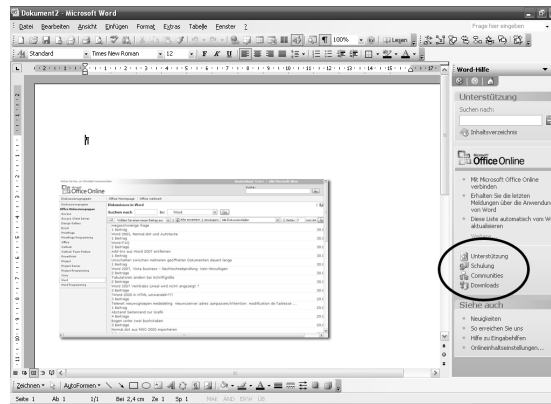


Abb. 4: Integrierung communitybasierter Hilfe bei Windows Word.

Je nach politischem Standpunkt kann man darin eine softwaretechnische Umsetzung des Montessoriprinzip der Hilfe zur Selbsthilfe oder die Anwendung des Tom-Sawyer-Prinzips – Tätigkeiten, die man selbst nicht leisten kann oder möchte, anderen Leuten zu übertragen<sup>26</sup> –, erkennen. Wir denken, dass der Idee von *produktspezifischen Communitysystemen* beide Interpretationen gleichermaßen eingeschrieben sind. Diese Gleichzeitigkeit kann deshalb sowohl seitens des Herstellers und der Nutzer gleichermaßen zu einer Ablehnung als auch Anerkennung des Ansatzes führen. Auf Seiten des Herstellers könnte bezüglich des Konzepts communitybasierter Hilfesysteme skeptisch angemerkt werden, dass durch die Öffnung dem Missbrauch durch konsumkritische Aktivisten, Scherzkekse, Spam, Werbekampagnen der Konkurrenz etc. Tür und Tor geöffnet wird. Ebenso können Käufer die Technologie ablehnen, weil sie es z.B. für eine Zumutung halten, dass – wenn etwas bei der Applikation nicht intuitiv bedienbar ist, sie bei

26 Vgl. dazu auch den Beitrag auf Wikipedia unter <http://de.wikipedia.org/wiki/Tom-Sawyer-Prinzip>, 31.12.2006. Der Hinweis ‚Dieser Artikel wurde zur Löschung vorgeschlagen‘ entbehrt dabei nicht einer unfreiwilligen Selbstironie.

schlechter Software dem Hersteller auch noch aus der Patsche helfen und anderen Leuten mitteilen sollen, wie man das Problem umgeht.

## 8. DIE SEMIOTISCHEN ÜBERLEGUNGEN ZUM KONZEPT DES WIKI INSIDE

„Zeichen ist alles, was zum Zeichen erklärt wird, und nur was zum Zeichen erklärt wird. Jedes beliebige Etwas kann (im Prinzip) zum Zeichen erklärt werden.“<sup>27</sup> Das Konzept des *Wiki inside* greift die Idee communitybasierter Hilfesysteme auf und verbindet die Idee mit der Vorstellung von objektbezogener Annotation. Dabei gehen wir davon aus, dass die Mensch-Computer-Interaktion als ein semiotischer Prozess beschrieben werden kann.<sup>28</sup> Bei unseren semiotischen Überlegungen spielt dabei die Nutzerschnittstelle *user interface* eine zentrale Rolle, da sie der Ort ist, an dem der Nutzer mit einer Anwendung interagiert.

Im Laufe der Zeit wurden verschiedene Typen von Interfaces entwickelt. In den Frühzeiten der Computertechnologie herrschte die Bedienung mittels Kommandozeilen-*interpreter* (*Command Line Interfaces* bzw. *CLI*) vor, bei denen der Nutzer durch seine *shell* Kommandos eingeben bzw. die Ausgaben des Computers lesen konnte. Heutzutage haben sich grafische Nutzerschnittstellen (*Graphical User Interface* bzw. *GUI*) durchgesetzt, so dass nahezu alle modernen Desktopcomputer auf der Verwendung von grafischen Nutzerschnittstellen basieren.

Ben Shneiderman hat den Erfolg grafischer Nutzerschnittstellen auf die *direkte Manipulation* zurückgeführt:

The central ideas seemed to be visibility of the object of interest; rapid, reversible, incremental actions; and replacement of complex command language syntax by direct manipulation of the object of interest – hence the term ‚direct manipulation‘.<sup>29</sup>

Semiotisch lässt sich Shneidermans Aussage in der Weise interpretieren, dass der Computerbildschirm uns nicht als ein Haufen darauf abgebildeter Pixel erscheint. Vielmehr findet ein – durch Ein- und Ausgabe vermittelter – semiotischer Prozess statt, bei dem die Interaktion derart strukturiert ist, dass uns die Pixel auf dem Monitor als Zeichen erscheinen, die auf etwas anderes verweisen. Nach Shneiderman sind dies – bzw. sollten dies sein – die *objects of interest*, die wir durch die Eingabewerkzeuge *direkt manipulieren* können. Shneidermans Interpretation hat dabei auch Eingang in die Standardisierung der Ergonomie der Mensch-System-

27 Bense: Semiotik, S. 9.

28 In der Forschung wurde diese Vorstellung z.B. durch Frieder Nake, Peter Borg Anderson bzw. der Forschungsgruppe um De Souza und Barbosa entwickelt. Jedoch kann nicht von der Computersemiotik gesprochen werden, da die einzelnen Wissenschaftler sich auf ganz unterschiedliche Semiotikansätze berufen.

29 Shneiderman: „Direct Manipulation“, S. 57.

Interaktion gefunden: „Bei der Dialogführung mittels direkter Manipulation lösen Nutzer Operationen dadurch aus, dass auf dem Bildschirm angezeigte Objekte ähnlich wie physikalische Gegenstände behandelt werden.“<sup>30</sup>

Die sich im Anschluss an Shneiderman aufdrängende Frage lautet nun, wie diese *objects of interest* bestimmt werden. Hier konnotiert die *direkte Manipulation* eine substanzialische Vorstellung, bei der sowohl vermittelnde Zeichenträger – als auch die *objects of interest* fixe Atome darstellen. An dieser Stelle unterscheiden wir uns und greifen die Überlegung einer strukturalistischen Semiotik auf. Demnach besitzen Zeichen zwar eine materielle Gebundenheit, sie sind aber als relative, oppositiv aufeinander bezogene Größen innerhalb einer Lebenspraxis zu verstehen, die sich wiederum in einen historisch-materiellen Prozess herausgebildet haben.

Bei der Analyse eines Zeichenprozesses greifen wir auf Charles Sanders Peirce zurück,<sup>31</sup> der ein Zeichen durch ein triadisches Verhältnis von Objekt – Repräsentamen – Interpretant bestimmt sieht:

A Sign, or Representamen, is a First which stands in such a genuine triadic relation to a Second, called its Object, as to be capable of determining a Third, called its Interpretant, to assume the same triadic relation to its Object in which it stands itself to the same Object. The triadic relation is genuine, that is its three members are bound together by it in a way that does not consist in any complexus of dyadic relations.<sup>32</sup>

Dabei muss man beachten, dass Zeichen nicht atomar sind, sondern in einem rekursiven Verhältnis zueinander stehen. Das heißt, dass man sich ein einzelnes Zeichen am besten als einen Schnitt im Kontinuum<sup>33</sup> einer Semiose denken sollte, die sich relativ zu einer Lebenspraxis vollzieht. Die Festlegung einzelner Zeichenentitäten stellen dabei (notwendige) Heuristiken vor einen pragmatischen Hintergrund dar.

Nach diesen Vorbemerkungen wollen wir die *Wiki inside* zu Grunde liegende Idee anhand eines fiktiven Beispiels aus der *Realwelt* illustrieren. Das Foto in Abb. 5. wurde in der Filiale einer Discounter-Kette aufgenommen und mit der erstbesten Frage versehen, die uns bei Betrachten des Bilds eingefallen ist. Damit haben wir versucht, exemplarisch die Idee von *Virtual PostIt*<sup>34</sup> zu illustrieren, die

---

30 DIN\_9241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 16, S. 4.

31 Siehe auch Hoffmann: Erkenntnisentwicklung.

32 Peirce nach Houser/Kloesel: *The Essential Peirce*, S.7.

33 Vgl. Zink: Kontinuum und Konstitution der Wirklichkeit.

34 Zur der Idee der *Virtual PostIt* siehe auch Grüttner: Entwicklung eines generischen Visualisierungs- und Interaktionskonzepts für kontextsensitive Hilfesysteme und prototypische Implementierung für das Eclipse RCP-Framework.

eine gestalterische Umsetzung des Konzepts eines communityorientierten Hilfesystems darstellt. Um nun die Idee der *Virtual PostIt* besser zu verstehen, sollte man deren Prinzip von der Physik in die Semiotik übertragen. Unter dieser neuen Perspektive ist das *PostIt* ein *indexikalischer Zettel*, der aber nicht mehr an einem *physikalisch bestimmten*, sondern an einem *semiotisch bestimmten* Objekt klebt.



Abb. 5: *Frag am besten Deine Mutter ;)*

Im Folgenden soll das Bild nun aus einer solchen semiotischen Perspektive analysiert werden, um die Grundprinzipien einer objekt-bezogenen Annotierung aufzuzeigen. Die Frage „Wie backe ich eigentlich einen Hefezopf?“ lässt sich dabei als Interpretant auf das in Abb. 5 abgebildete Foto lesen. Das heißt, umkehrseitig bildet das Foto das Repräsentamen für diesen Interpretanten. Nachdem wir also Repräsentamen und Interpretant bestimmt haben, können wir uns fragen, auf welches Objekt sich das Repräsentamen bezieht. Nun erkennen wir ein Regal mit diversen Packungen in einem Supermarkt und verschiedene Packungen, die aus quaderartigen Pappkartons zu bestehen scheinen. Daneben ist das blau-weiß-rote *Dr. Oetker*-Logo zu erkennen. Rechts auf der Packung ist ein angeschnittener Marmorkuchen zu erkennen. Es ist also anzunehmen, dass das Bild in der Backabteilung eines Supermarkts aufgenommen worden ist. Es bietet sich also an, *Backabteilung eines Supermarkts* als Objekt des Zeichens anzunehmen. Mittels dieser Überlegung haben wir also eine mögliche Triade (*Backabteilung eines Supermarkts, Foto, Wie backe ich eigentlich einen Hefezopf*) rekonstruiert, die das Zeichen bestimmt. Nun wollen wir die Akteure bestimmen, die in dem Bild objekt-bezogene Annotierungen hinterlassen haben. Auf dem Bild lassen sich hier drei verschiedene Typen von Akteuren ausmachen:

- die verschiedenen Hersteller der im Regal dargebotenen Produkte,
- der Supermarktbetreiber,
- derjenige, der das Bild mit dem ‚gelben Zettel‘ versehen hat.

Versuchen wir jetzt, aus der Perspektive der verschiedenen Akteure noch deren Annotierungsmöglichkeiten zu bestimmen.

Auf der *Ebene des Herstellers* bietet es sich an, sein Produkt im Regal als das antizipierte Objekt der Zeichen-Triade zu nehmen, das verkauft werden soll. Ebenfalls bietet sich an, dass die Verpackung als Repräsentamen der Zeichen-Triade antizipiert wird, insbesondere weil hier der Hersteller die Gestaltung des so antizipierten Repräsentamen beeinflussen kann. So kann er z.B. versuchen, auf den Interpretanten der Zeichen-Triade dadurch Einfluss zu nehmen, dass er – wie es z.B. bei Weinflaschen üblich geworden ist – einen passenden Verwendungszweck für das Produkt auf der Verpackung gleich mit abdruckt. Er kann auch versuchen, die Interpretation in seinem Sinne zu beeinflussen, indem er z.B. auf der Verpackung einen perfekt gelungenen Marmorkuchen abbildet. Jedoch ist der Hersteller genauso wie die Entwickler von Hilfesystemen der *situativen* und *kulturellen Dekontextualisierung* unterworfen, so dass er zwar die Verpackung als antizipierten Repräsentamen mitgestalten kann, er jedoch weder sicher sein kann, dass die Verpackung in seinem Sinne wahrgenommen, geschweige denn interpretiert wird. Die *situative Dekontextualisierung* erkennt man z.B. daran, dass manche Hersteller zum Zeitpunkt der Abfüllung nicht wissen in welchem Land das Produkt verkauft wird, weshalb sie z.B. die Inhaltstoffe auf der Verpackung in mehreren Sprachen abdrucken.<sup>35</sup>

Auf der *Ebene des Supermarktbetreibers* lässt sich auf dem Bild erkennen, dass bei der objektbezogenen Annotierung der räumliche Kontext des Supermarkts und der Waren ausgenutzt wurde, um so die Waren in Form von orangefarbenen Preisschildern zu annotieren. Auf dem Weg zu dem *Virtual PostIt* können wir aber die interessante Beobachtung machen, dass zwar mit den Preisschildern das einzelne physikalische Objekt ausgezeichnet wird, die Annotation sich aber auch auf das Objekt des Produkttyps bezieht. Insbesondere ist die Beziehung zum einzelnen physikalischen Objekt doppelt kodiert: Einmal mittels eines räumlichen Index – *das Ding in der näheren Umgebung* – und zur Sicherheit noch mittels eines symbolischen Index – *das Ding mit der gleichen Produktbezeichnung*.

Um auf der nächsten Ebene nun den gelben Marker auf dem Bild – ein Fake, um die Designidee der *Virtual PostIt* zu kommunizieren – zu analysieren, stellen wir uns einen Augenblick vor, dass derjenige, der das Bild mit einer Anmerkung versehen hat, eine neuartige Handykamera besitzt und wir das Bild ebenfalls durch eine solche Handykamera betrachten. Die neuartige Handykamera besteht darin, dass sie erlaubt, im Bild semiotisch bestimmte Objekte mit virtuellen *PostIts* zu annotieren bzw. sich die *PostIts* anderer Leute anzeigen zu lassen. In dem sich so vorgestellten Zukunftsszenario könnte die Bildunterschrift von Abb. 5 nun die Antwort darstellen, die wir per SMS an das eingeblendete virtuelle *PostIt* schicken, so dass auf dem *PostIt* ein Diskussionsthread entsteht.

Die softwaretechnische Herausforderung der so eben skizzierten Gestaltungsidee besteht dabei darin, Verfahren zur Konstruktion geeigneter semioti-

---

35 Eine andere Lesart, die dies nicht auf eine zeitliche Dekontextualisierung zurückführen würde, könnte darin aber auch einen Ausdruck sehen, dass der Hersteller damit seine internationale Bedeutung kommunizieren möchte.



scher Objekte zu entwickeln, damit die durch das Verfahren ausgezeichneten Objekte als Träger für die virtuellen *PostIts* fungieren können. Für eine algorithmische Konstruktion der Objekte haben wir dabei<sup>36</sup> drei Forderungen aufgestellt:

1. Der Algorithmus soll einen über die Zeit stabilen Träger auszeichnen.
2. Durch den Algorithmus sollen nur für den Nutzungskontext sinnhafte Objekte als Träger für *PostIts* konstruiert werden, d.h. insbesondere sollte der Algorithmus derart transparent sein, dass – im Idealfall – der algorithmisch bestimmte Träger mit den lebensweltlich bestimmten semiotischen Objekt zusammenfällt.
3. Der Algorithmus sollte es erlauben, die – an den Träger gehängten – *PostIts* möglichst einfach zu finden bzw. wieder zu finden.

Auf der semiotischen Ebene unterscheidet sich das fiktive Beispiel, eines auf einem Wiki basierenden virtuellen *PostIt*, das in die *Realität* eingeflochten wird, nicht grundlegend von dem im nächsten Abschnitt vorgestellten, in eine Anwendung integrierten, Hilfesystem auf Wiki-Basis. Es ist jedoch anzunehmen, dass sowohl die technischen Bedingungen der Objektkonstruktion<sup>37</sup>, als auch dessen, was sinnvolle semiotisch bestimmte Objekte darstellen, sich unterscheiden.<sup>38</sup>

Im Folgenden wollen wir uns aber nur auf den Bereich der in Applikationen integrierten Hilfesysteme beschränken. Diese eben angestellten semiotischen Überlegungen helfen jedoch, die Funktionsweise unseres Verfahrens besser zu verstehen. Insbesondere lässt sich durch die semiotische Interpretation eine Binnendifferenzierung kontextspezifischer Hilfesystemansätze vornehmen. Kontextbewusste Ansätze, die auf KI-Algorithmen basieren, lassen sich dadurch charakterisieren, dass sie möglichst gut den *Kontext auf der pragmatischen Ebene eines Zeichens* zu erfassen suchen. Die unter dem Label kontextsensitiver Hilfe gefassten Techniken wie Tooltips oder die *Windows-FI*-Hilfe verzichten jedoch im Allgemeinen auf eine solche Erfassung des Kontexts auf der pragmatischen Ebene. Sie versuchen stattdessen auf der Ebene der *semiotischen Objekte*, diese am Interface zu identifizieren und bzgl. dieser Objekte erläuternde Hilfetexte einzublenden. Damit stellen diese Ansätze vielmehr *indexikalische* bzw. genauer *deiktische Hilfesystemansätze* dar. Innerhalb dieser deiktischen Ansätze lassen sich auch unsere Arbeiten verorten.

---

36 Vgl. Stevens/Wiedenhöfer: „CHIC – A pluggable solution for community help in context“.

37 Z.B. ließen sich GPS-Daten, von Hersteller bereitgestellte Produktinformation (Name, Barcode, etc.) oder biometrische Daten (Kopfform, Gesten, ...) zur Objektkonstruktion heranziehen.

38 Z.B. könnte man sich vorstellen, dass Produkte um verbraucherkritische Informationen angereichert werden, oder jemand der Kassiererin mit Kopftuch einen virtuellen Zettel „Islamistin – nicht rechnen“ anklebt, um an dieser Stelle alle Klischees einer nicht gewünschten Aneignung zu bedienen.

	Kontextsensitive Hilfe	Wiki inside	Virtual PostIt
<b>Auszeichnung von Objekten</b>	- vor Auslieferung - durch Hersteller	- vor Auslieferung - durch Hersteller	- während der Laufzeit, - durch heuristische Algorithmen
<b>Annotation von Objekten</b>	- vor Auslieferung - durch Hersteller	- nach Auslieferung - communityorientiert	- nach Auslieferung - communityorientiert

Tab. 1: Unterscheidung community- und herstellerzentrierter deiktischer Hilfesystemansätze.

Neben der Betrachtung der Interaktion im Nutzungskontext ist darüber hinaus hilfreich, sich den Produktionspraktiken zu betrachten: Beim traditionellen Verfahren der kontextsensitiven Hilfe fügen die Entwickler in den *source code* spezielle *help identifier* ein, welche User-Interface-Elemente, so genannte Widgets, zu geordnet werden. An einer anderen Stelle im *source code* bzw. in ausgelagerten Dateien schreiben dann die Entwickler dazugehörige Hilfetexte (dies wäre dann der Fall *Kontextsensitive Hilfe*: in Tabelle I). Während der Laufzeit liest dann das Hilfesystem den *help identifier* aus und zeigt den zugeordneten Hilfetext an.

Im ersten Schritt lassen sich die Herstellerabhängigkeiten dadurch verringern, dass zwar die Auszeichnung der Objekte mittels spezieller *help identifier* noch durch den Hersteller geschieht, aber es der Nutzercommunity möglich ist, die so bestimmten Objekte zu annotieren (dies wäre dann der Fall *Wiki inside* in Tabelle I).

Bei der Umsetzung im *BSCWeasel-Projekt*<sup>39</sup>, bei dem unsere Applikation aus weiten Teilen aus Komponenten von Drittanbietern besteht, zeigte sich leider, dass die Hersteller der anderen Komponenten gar keine oder nur rudimentär *help identifiers* eingefügt hatten. Deshalb mussten wir von außen Zeichenträger durch Inspektion des Systemzustands mittels heuristischer Algorithmen erzeugen, die die oben genannten drei Anforderungen möglichst gut umsetzen (dies wäre dann der Fall *Virtual PostIt* in Tabelle I).

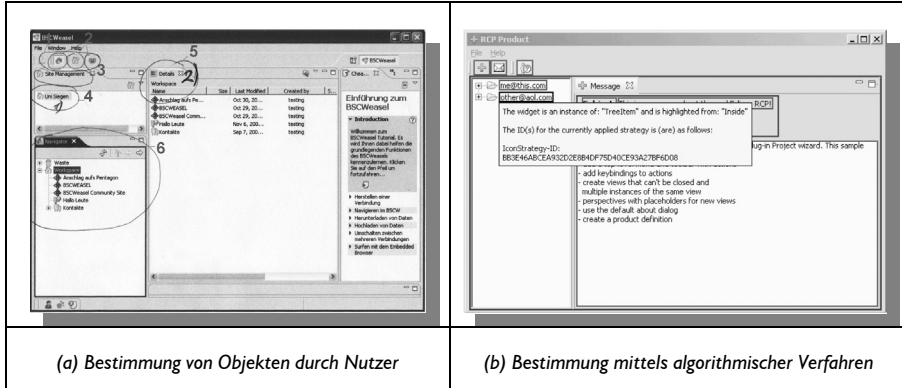
Ohne hier auf die technischen Details eingehen zu können,<sup>40</sup> wollen wir kurz das Vorgehen skizzieren, mittels dessen wir ein semiotisch orientiertes Verfahren entlang den obigen drei Anforderungen entwickelt haben. Dieses konstruiert algorithmisch Objekte in der GUI, auf die dann in einem Nutzungsdiskurs Bezug genommen werden kann.

Hierzu haben wir in einem ersten Schritt Nutzer gebeten, in einem *snapshot* jene Stellen einzukreisen, die sie interessieren. Im nächsten Schritt haben wir versucht eine Grammatik in den Auszeichnungen zu erkennen und im dritten Schritt diese mittels algorithmischer Verfahren zu simulieren. Hierbei haben wir insbe-

39 Vgl. <http://www.bscweasel.de>, 22.02.2008.

40 Siehe dazu Wiedenhöfer: *Help in Context*; Grüttner: *Entwicklung eines generischen Visualisierungs- und Interaktionskonzepts für kontextsensitive Hilfesysteme und prototypische Implementierung für das Eclipse RCP-Framework*; Stevens/Wiedenhöfer: „CHIC – A pluggable solution for community help in context“.

sondere davon profitiert, dass Entwickler heutzutage bei der software-technischen Umsetzung des *direct manipulation*-Konzepts auf standardisierte WIMP-Programm-bibliotheken<sup>41</sup> zurückgreifen und sich so Muster im Aufbau heutiger Anwendungssystemgestaltung wieder finden lassen, an denen unsere heuristischen Verfahren ansetzen können.



(a) Bestimmung von Objekten durch Nutzer

(b) Bestimmung mittels algorithmischer Verfahren

Tab. 2: Simulation der Bestimmung semiotisch bestimmter Objekte mittels heuristischer Verfahren.

## 9. COMMUNITY HELP IN CONTEXT: REALISIERUNG EINER WIKI-BASIERTEN HILFE

An dieser Stelle wollen wir eine prototypische Referenzimplementierung darstellen. Ziel dieses Prototyps war unter anderem, entscheidende Erkenntnisse über die Brauchbarkeit und die Nutzbarkeit unseres Konzeptes zu gewinnen. Die Herausforderung bei der Realisierung lag, zum einen in der Implementierung eines geeigneten Verfahrens zur Bestimmung der semiotischen Objekte, zum anderen in der Umsetzung einer Annotierbarkeit, die sich gut in das Interface bestehender Anwendungen integriert. Um diesen Anforderungen gerecht zu werden, wurde die zu Grunde liegende Architektur in die drei Module CBHS (*Community Based Help System*), AIM (*Application Integration Module*) und CAM (*Context-Aware Adaptation Module*) aufgeteilt. Diese Architektur basiert auf unserer Einschätzung, dass komponentenbasierte Systeme den Aufwand bei Anpassbarkeit und Veränderungen entscheidend verringern und es die Integration in jedes Anwendungssystem erleichtert. Das AIM (*Application Integration Module*) integriert die zu einem Objekt geführten Diskurse der Community direkt in die vom Nutzer verwendete Applikation. Er interagiert dabei über das AIM-Modul mit dem Hilfesystem und erhält Hilfeinformationen im Kontext seiner Arbeit auf seine Anforder-

41 WIMP steht für windows, interface, mouse, pointer. Im Kontext der Eclipseprogrammierung werden WIMP-Funktionalitäten durch einige Basiskomponenten bereitgestellt, auf die dann Anwendungsentwickler aufsetzen.

rung bereitgestellt. Drei Eigenschaften werden dabei durch das AIM-Modul realisiert:

1. *Integration in den Arbeitskontext des Nutzers*

Ziel des Moduls ist, trotz der Anwendungsunabhängigkeit dem Nutzer den direkten Zugriff auf Hilfeinformationen innerhalb seines Arbeitskontextes zu gewähren. Tritt während der Bearbeitung einer Aufgabe beim Anwender ein Problem auf, so ist gewährleistet, dass der Hilfesuchende das Hilfesystem innerhalb seiner momentanen Arbeitssituationen aufrufen kann und nicht durch das Verlassen des Anwendungskontextes den Bezug zum eigentlichen Problem verliert.

2. *Einfacher und schneller Zugriff auf Hilfeartikel*

Neben der direkten Integration in die Applikation geht die Implementation darüber hinaus, dass der Zugriff auf Hilfeinformationen mit möglichst geringem Aufwand geschieht. So muss die *usability* der Nutzeroberfläche des Hilfesystems gewährleisten, dass der Nutzer durch einen einfachen Klick die Hilfeinformationen bereitgestellt bekommt, die ihn bei der Problemlösung am effektivsten unterstützen.

3. *Anwendungsunabhängiges Hilfesystem*

AIM realisiert die Integration des Hilfesystems in beliebige Anwendungssysteme. So muss gewährleistet sein, dass sowohl eine direkte Integration in den Arbeitskontext des Anwenders gegeben ist, als auch ein anwendungsunabhängiges Framework geschaffen wird, das für beliebige Softwareprodukte verwendbar ist.

Das CAM (*Context-Aware Adaptation Module*) ermittelt die semiotisch bestimmten Objekte bzw. eine Referenz darauf, die zum Nachrichtenaustausch herangezogen werden kann. Man kann sich das Modul auch als die Berechnung eines objektspezifischen Funkkanals vorstellen. Es fungiert dabei als Vermittler zwischen den Nutzungskontext und dem allgemeinen Wikisystem. Die Kunst des Moduls besteht darin, sinnvolle und über die Zeit stabile Referenzen bereitzustellen (siehe Diskussion im vorherigen Abschnitt).

Das CBHS-Modul stellt dabei die Communityfunktionalitäten bereit, d.h. den Punkt, an dem Nutzer zusammentreffen, um objektbezogene Messages auszutauschen oder eine Erläuterung zum referenzierten Objekt zu hinterlassen. Dabei liegt hier die Flexibilität in der Auswahl von Communitysystemen. Das Modul gewährleistet die Verwendung traditioneller Communitysysteme, jedoch bestückt mit speziellen Funktionalitäten, die für die Bereitstellung von kontextsensitiven Hilfen nötig sind. In unserem Fall haben wir für dieses Modul ein Wikisystem ausgewählt, das genau unsere Anforderungen trifft.

In der konkreten Umsetzung wurde das am Lehrstuhl für Wirtschaftsinformatik und Neue Medien der Universität Siegen entwickelte *groupware*-System

BSCWeasel<sup>42</sup> verwendet. BSCWeasel stellt als Desktopanwendung einen alternativen Zugang zum weitverbreiteten BSCW-Groupware-System bereit und diente als Basisanwendung für unser Hilfesystem. Als Vorteil erweist sich hier, dass das System auf dem Eclipse Framework aufbaute und so nützliche Funktionen, wie die Standardhilfe und kontextsensitive Hilfe, genutzt und an unsere Anforderungen angepasst werden konnten. Zur Realisierung des CBHS-Moduls wurde das Wiki-System Atlassian Confluence<sup>43</sup> eingesetzt, weil es Webservice API, die einen algorithmischen Zugriff auf das System erlauben, und die anpassbare Nutzeroberfläche besitzt. Beides zusammen erlaubte es uns, das System harmonisch in das Hilfesystem von Eclipse<sup>44</sup> zu integrieren und nötige Methoden zur deiktischen Referenzierung bereit zustellen.

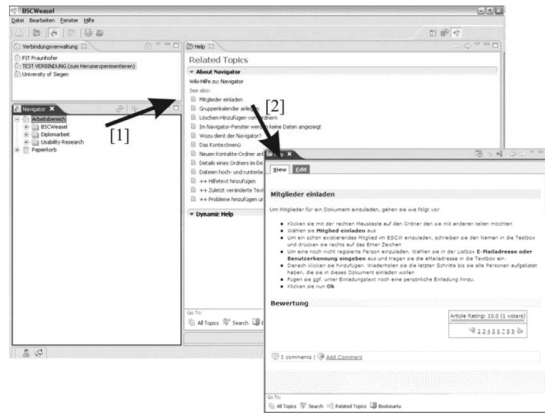


Abb. 6: Ablauf des Hilfeaufrufs in der Referenzimplementierung.

Abb. 6 zeigt die prototypische Realisierung des Konzeptes und den Ablauf des Hilfeaufrufs. Nach Aufruf des Hilfesystems (1) durch Drücken der Taste F1 oder entsprechenden Symbol- oder Menüeinträgen werden dem Nutzer, passend zu seiner Problemsituation, die entsprechenden Hilfeartikel bereitgestellt (2). Die Idee, die hinter der Objekterfassung steht, ist, dass das am Interface selektierte Objekt ermittelt wird.<sup>45</sup> Die Herausforderung lag hier in der Generierung einer Referenz, die ein Objekt im Sinne der drei obigen Anforderungen semiotisch bestimmt<sup>46</sup> und mit den entsprechenden Hilfeinformationen verknüpft. Im Falle des CHiC-Prototyps, der an die kontextsensitive Hilfe gekoppelt wurde, wurde auf

42 Siehe: <http://www.bscweasel.de>, 31.3.2007.

43 Siehe: <http://www.atlassian.com/software/confluence/>, 31.3.2007.

44 Siehe: <http://www.eclipse.org>, 31.3.2007.

45 Vgl. Wulf/Golombek: „Direct Activation“.

46 Das bedeutet, dass die Referenz zeitlich stabil sein musste und der Nutzer zu jeder Zeit in der Lage ist, die Referenzierungsbedingungen reproduzieren zu können, um z.B. auf bereits gelesene Hilfeartikel zurückgreifen zu können.

die so genannte *View-ID* zurückgegriffen, die die Entwickler zur allgemeinen Verwendung öffentlich zugänglich gemacht haben und deshalb stabil gehalten werden muss. Für den Nutzer sieht es dabei so aus, als ob die Hilfe sich auf die angeklickte Sicht aus Schritt 1 bezieht.

Nach Aufruf des Hilfesystems werden dem Nutzer die Überschriften der zu dem referenzierten Objekt verfassten Hilfeartikel angezeigt. Klickt der Nutzer im Schritt 2 auf eine der als Link hinterlegten Überschriften, gelangt er zum entsprechenden Hilfetext. Diese Hilfeartikel sind dabei Produkte der Community einzelner Nutzer oder des Herstellers, der diese (nach Auslieferung) erstellt hat. Da es sich bei den Hilfeartikeln um eine normale Wikiseite des zuvor beschriebenen Wikisystems handelt, kann jeder (je nach Konfiguration des Wikisystems) den Hilfeartikel bearbeiten, bewerten oder kommentieren.

## 10. ERFAHRUNG MIT DEM CHIC-PROTOTYPEN

Aufbauend auf dieser prototypischen Implementierung des CHiC-Konzeptes konnte das System einer ersten Evaluation unterzogen werden. Jedoch sind neue Konzepte äußerst schwer zu evaluieren, gerade wenn es sich um communitybasierte Systeme handelt, dessen volle Auswirkung erst nach der Etablierung einer aktiven Community überprüfbar ist, weshalb die hier vorgestellten Ergebnisse nur tentativen Charakter haben.

Die Erfahrungen mit dem CHiC-Prototypen beruhen auf Beobachtungen der alltäglichen Nutzung von CHiC als eines integrierten Hilfesystems des *BSCWeasel*. Das *BSCWeasel* ist eine Desktopanwendung, die einen alternativen Zugang zu dem populären *groupware*-System *BSCW* erlaubt und stellt eine Reihe von Funktionen bereit, die kollaboratives Arbeiten besser unterstützen sollen. Genutzt wird das *BSCWeasel* primär in einer Forschungsgruppe in Siegen, an der das *BSCWeasel* als *open source*-Projekt in Projektarbeiten maßgeblich entwickelt wurde. Die hier vorgestellte Evaluation bezieht sich jedoch primär auf eine *usability*-Studie, in der das Hilfesystem einer Reihe von Benutzungstests auf Basis der prototypischen Referenzimplementierung unterzogen worden ist. Die *usability*-Studie fokussierte dabei auf die Nutzbarkeit und Brauchbarkeit des Konzeptes. Die Untersuchung teilte sich in die Phasen, Benutzungstest mittels eines aufgabenorientierten *walkthroughs* und in eine Interviewphase. Mittels des szenarienbasierten *walkthroughs* sollten möglichst authentische Nutzungssituation geschaffen werden, um vor diesem Hintergrund den praktischen Umgang mit dem Hilfesystem im Anwendungskontext von *BSCWeasel* zu beobachten.

Um die aufgedeckten Handlungsmuster besser zu verstehen und weitere Erkenntnisse zur Akzeptanz des Konzeptes zu gewinnen, wurden im Anschluss narrative Interviews durchgeführt. Während der ein- bis anderthalbstündigen Untersuchungen zeigte sich, dass die direkte Integration des Hilfesystems in die zugrunde liegende Applikation einen kritischen Punkt darstellt. Die visuelle ‚Verschmelzung‘ des Hilfesystems mit *BSCWeasel* führte dazu, dass das Hilfesystem

durch den Nutzer nur als traditionelles Hilfesystem interpretiert wurde. Zwar wurde durch die Integration die *usability* deutlich verbessert, doch litt darunter stark der Communityaspekt des Systems. Dies führte in einem Fall sogar so weit, dass ein Nutzer nicht erkannte, dass er gerade mit einem Wiki basierten Hilfesystems gearbeitet hatte. In der anschließenden Diskussion gab er uns den Rat-schlag, dass ein Wikisystem als Hilfesystem doch eine tolle Idee wäre, da so Hilfetexte kooperativ erstellt und bearbeitet werden könnten. Diese Aussage zeigte, dass der Wiki Aspekt der integrierten Wiki-Hilfe unseres Systems nicht wahrgenommen wurde. Das heißt, auf Grund der traditionellen Nutzung von Hilfesystemen wurde zwar Wiki gewünscht, aber nicht erwartet und deshalb auch nicht entdeckt.

Des weiteren zeigte unsere Untersuchung noch ein anderes Problem: dass unser communitybasiertes Hilfesystem noch eine weitere Altlast traditioneller Hilfesysteme ‚erbt‘. Da die Nutzer nicht direkt beauftragt wurden, das Hilfesystem zu nutzen, konnten wir den Effekt beobachten, dass Probanden bei auftretenden Problemen gar nicht versuchten, mittels des Hilfesystems eine Lösung zu finden. In den anschließenden Interviews begründeten die Nutzer ihr Nutzungsverhalten mit den negativen Erfahrungen, die zuvor mit Hilfesystemen gemacht wurden. Frustration durch erfolglose oder aufwändige Suche nach passenden Problemlösungen führte zu einer allgemein ablehnenden Haltung gegenüber eingebetteten Hilfen und zu einer Skepsis gegenüber neuen Konzepten.

Zusammenfassend konnten wir aus unserer Untersuchung drei wichtige Erkenntnisse gewinnen:

1. *Usability*-Probleme bei der prototypischen Umsetzung des Konzepts

Trotz der frühen Phase der Entwicklung verhielt sich, aufgrund der Integration in die Standardhilfe, das System bei der Nutzung als traditionelle Kontexthilfe erwartungskonform. Es lässt sich jedoch deutlich erkennen, dass dem Prototyp noch wichtige Funktionalitäten wie eine Indexsuche fehlen bzw. die Gebrauchstauglichkeit an einigen Stellen, wie z.B. bei der Erstellung von Texten, noch verbesserungswürdig ist.

2. *Intensivere* Kennzeichnung der Communityaspekte durch *community engineering*

Während der Untersuchung ließ sich erkennen, dass die fehlende Präsenz des Communityaspekts eine kritische Größe darstellte. Auf technischer Ebene sollte eine bessere Präsenz sowohl innerhalb des Metaartefakts Hilfesystem (z.B. durch Herausstreichen der *Communityfeatures* auf den Wikiseiten), als auch das in dem eigentlichen Artefakt (z.B. durch Anzeigen, wie viele Nutzer gerade online sind, oder das Einblenden neuer Fragen in der Statusleiste) geschehen. Auf der nichttechnischen Ebene sollte überlegt werden, Methoden des *community engineering* anzuwenden, indem man z.B. Präsenze (Gratis-T-Shirts) oder Ehrungen (User des Monats) für eine aktive Teilnahme auslobt.

### 3. Integration in verschiedene Anwendungsfelder

Die ursprüngliche Zielgruppe des CHiC-Konzepts waren die Nutzer, denen durch die Weiterentwicklung des Hilfesystems ein Diskursmedium zur verbesserten kooperativen Aneignung bereitgestellt werden sollte. In der Realisierung des Systems zeigte sich, dass das CHiC-System auch für Hersteller und Entwickler auf Grund der direkten Annotierbarkeit aus dem Nutzungskontext Vorteile gegenüber der traditionellen Erstellung kontextsensitiver Hilfe bietet. Das heißt, technische Funktionen, die eigentlich für aktive Nutzer gedacht waren, können auch nur von den Softwareherstellern selbst beansprucht werden, um z.B. im *usability*-Test auf Grund von Interpretationsproblemen mit den Nutzern gemeinsam unmittelbar geeignete Erläuterungen zu den Objekten zu schreiben oder nach Auslieferung auf Grund gemeldeter Nutzungsprobleme das Hilfesystem eines Softwareproduktes zu erweitern.

## 11. SCHLUSSBETRACHTUNG

In diesem Beitrag wurde sich dem Thema der Aneignung von Computersystemen mit Hilfe einer ‚voraussetzungsgarmer‘ Interpretation des Interaktionsbegriffs als einer Analyseperspektive genähert, die es erlaubte, die verschiedenen Gestaltungsansätze miteinander in Bezug zuzu setzen. Zugleich haben wir versucht, den Weg einer neuartigen Gestaltungsidee – dem *Wiki inside* – nachzuzeichnen.

Dieses doppelte Ansinnen stellt dabei ein typisches Merkmal der Wirtschaftsinformatik dar, welche sich sowohl der Praxisrelevanz als auch der Wissenschaftlichkeit gleichermaßen verpflichtet fühlt.<sup>47</sup> Am Beispiel des *Wiki inside* sollte gezeigt werden, wie beides in eine produktive Spannung zueinander gebracht werden kann. Dabei stellt das Gestalten in der Praxis quasi ein qualitatives Experiment<sup>48</sup> dar, welches einer theoretischen Reflektion Orientierung bietet. Umgekehrt kann das zunächst einmal auf ein unpraktisches Verstehen gerichtete Theoretisieren neue Einsichten in die Praxis und nicht erkannte Handlungsoptionen für die Praxis eröffnen.

## LITERATURVERZEICHNIS

Ang, Ien: „Im Reich der Unsicherheit. Das Globale Dorf und die kapitalistische Postmoderne“, in: Hepp, Andreas/Winter, Carsten (Hrsg.): *Die Cultural Studies Kontroverse*, Lüneburg 2003, S. 84-110.

Bense, Max: *Semiotik. Allgemeine Theorie der Zeichen*, Baden-Baden 1967.

Carroll, John M./Rosson, Mary Beth: „Paradox of the Active User“, in: Carroll, John M. (Hrsg.): *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, Cambridge 1987, S. 80-111.

---

47 Vgl. Frank: „Informatik und Wirtschaftsinformatik“.

48 Vgl. Nett/Stevens: „Business Ethnography“.



- Carroll, John M. u.a.: „The Minimal Manual“, in: ACM SIGCHI Bulletin, Volume 20, New York, 1988, S. 80.
- Covi, Lisa/Ackerman, Mark: „Such Easy-to-use Systems: How Organizations Shape the Design and Use of Online Help Systems“, in: ACM (Hrsg.): Proceedings of the Conference on Organizational Computing Systems, Milpitas 1995, S. 280-288.
- Dey, Anind K. u.a.: „A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications“, in: Human Computer Interaction, Jg. 16, Nr. 2-4, 2001, S. 97-166.
- „Dialogführung mittels Direkter Manipulation“, in: Deutsches Institut für Normung e.V., Normenausschuß Ergonomie (FNERG) (Hrsg.): Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 16 , Berlin 2000, S. 1-28.
- Dix, Alan u.a.: Human-Computer Interaction, New York 1998.
- Frank, Ullrich: „Informatik und Wirtschaftsinformatik – Grenzziehungen und Ansätze zur gegenseitigen Befruchtung“, in: Desel, Jörg (Hrsg.): Das ist Informatik, Berlin 2001, S. 47-66.
- Grüttner, Marc: Entwicklung eines generischen Visualisierungs- und Interaktionskonzepts für kontextsensitive Hilfesysteme und prototypische Implementierung für das Eclipse RCP-Framework, Siegen 2007.
- Herrmann, Thomas: „Probleme bei der Konstruktion und beim Einsatz von Hilfesystemen“, in: Balzert, Helmut u.a. (Hrsg.): Einführung in die Software-Ergonomie, Berlin/New York 1988.
- Hoffmann, Michael H. G.: Erkenntnisentwicklung: Ein semiotisch-pragmatischer Ansatz, Frankfurt a.M. 2005.
- Houser, Nathan/Kloesel, Christian (Hrsg.): The Essential Peirce. Selected Philosophical Writings (1893-1913), Bloomington, IL 1998.
- Nett, Bernhard/Stevens, Gunnar: „Business Ethnography – Aktionsforschung als Beitrag zu einer reflexiven Technikgestaltung“, in: Track Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik auf der MKWI 2008, (MS.) erscheint vorauss. 2008.
- Pfeffer, Jeffrey/Hinds, Pamela J: „Why Organizations Don't ‚Know What They Know‘: Cognitive and Motivational Factors Affecting the Transfer of Expertise“, in: Ackerman, Mark u.a. (Hrsg.): Sharing Expertise – Beyond Knowledge Management, Cambridge 2003, S. 3-26.
- Preece, Jenny u.a.: Human Computer Interaction, 1994.
- Ramirez, Leonardo: Social Construction of End User Adaptations in Context Awareness Systems, Sankt Augustin 2007.
- Rolf, Arno: „Von der Theoriearbeit der Informatik zur Gestaltung“, in: Nake, Frieder u.a. (Hrsg.): Informatik zwischen Konstruktion und Verwertung –

- Materialien der 3. Arbeitstagung ‚Theorie der Informatik‘ Bad Hersfeld 3. bis 5.4.2003, Berlin 2004, S. 115-121.
- Shneiderman, Ben: „Direct Manipulation: A Step Beyond Programming Languages“, in: IEEE Computer, Jg. 16, Nr. 8, 1983, S. 57-69.
- Shneiderman, Ben: Designing the User Interface: Strategies for Effective Human-Computer Interaction, Massachusetts 1997.
- Sternberg, Robert J.: „Cognitive Conception of Expertise“, in: Feltouich, Paul J. u.a. (Hrsg.): Expertise in Context, Cambridge 1997, S. 45-62.
- Stevens, Gunnar u.a.: „Zum Handeln in Krisensituationen: Ein Beitrag zur Strukturationsdebatte“, in: Track Kooperationssysteme / Cooperation Systems auf der MKWI 2008, (MS.) erscheint vorauss. 2008.
- Stevens, Gunnar u.a.: „Breaking it up: An Industrial Case Study of Component-based Tailorable Software Design“, in: Liebermann, Henry u.a. (Hrsg.): End User Development, Dordrecht 2006, S. 269-294.
- Stevens, Gunnar/Wiedenhöfer, Torben: „CHIC – A Pluggable Solution For Community Help in Context“, in: Mörch, Anders u.a.: (Hrsg.) Proceedings of the 4th Nordic Conference on Human Computer Interaction, New York 2006, S. 212-221.
- Stevens, Gunnar u.a.: „Infrastrukturen zur Aneignungsunterstützung“, in: Oberweis, Andreas (Hrsg.): eOrganisation: Service, Prozess-, Market-Engineering: 8. Internationale Tagung Wirtschaftsinformatik, Karlsruhe 2007, S. 823-840.
- Suchman, Lucy: Plans and Situated Actions. The Problem of Human-Machine Communication, Cambridge 1987.
- Wiedenhöfer, Torben: Help in Context Konzeption und Umsetzung eines communityunterstützten Hilfesystems, (Dipl.-Arbeit) Siegen 2006.
- Wulf, Volker/Golombek, Björn: „Direct Activation: A Concept to Encourage Tailoring Activities“, in: Behaviour & Information Tech., Jg. 20, Nr. 4, 2001, S. 249-263.
- Zink, Julia: Kontinuum und Konstitution der Wirklichkeit. Analyse und Rekonstruktion des Peirce’schen Kontinuum-Gedankens, München 2004.