

# Zur anpaßbaren Gestaltung von Groupware

Habilitationsschrift im Fach Informatik am Fachbereich Informatik  
der Universität Hamburg

Volker Wulf

Hamburg  
Dezember 2000

## **Inhalt**

### **0 Zur anpaßbaren Gestaltung von Groupware: Einleitende und zusammenfassende Bemerkungen**

### **1 Evolving Cooperation when Introducing Groupware – A Self-Organization Perspective**

***Volker Wulf***

In: Cybernetics & Human Knowing, Vol. 6, No. 2, 1999, S. 55 - 73

### **2 Storing and Retrieving Documents in a Shared Workspace: Experiences from the Political Administration**

***Volker Wulf***

In: Howard, S.; Hammond, J.; Lindgaard, G. (Hrsg.): Human Computer Interaction: INTERACT 97, Chapman & Hall 1997, S. 469 – 476

### **3 Towards an Integrated Organization and Technology Development**

***Volker Wulf und Markus Rohde***

In: Proceedings of the Symposium on Designing Interactive Systems, 23. - 25.8.1995, Ann Arbor (Michigan), ACM-Press, New York 1995, S. 55 – 64

### **4 Tailoring Groupware for Different Scopes of Validity**

***Volker Wulf, Oliver Stiemerling und Andreas Pfeifer***

In: Behaviour & Information Technology, Vol. 18, No. 3, 1999, S. 199 - 212

## **5 Beyond "Yes and No" - Extending Access Control in Groupware with Awareness and Negotiation**

***Oliver Stiemerling und Volker Wulf***

In: Group Decision and Negotiation, Vol. 9, 2000, S. 221 - 235

## **6 Kokonstruktive Weiterentwicklung eines Groupwareproduktes - Diskutiert am Beispiel der Reimplementierung eines Suchtools**

***Helge Kahler und Volker Wulf***

In: Proceedings der achten GI-Fachtagung Software-Ergonomie'97, am: 3.3. - 6.3.1997 in Dresden, Teubner, Stuttgart 1997, S. 211 - 220

## **7 "Let's see your Search-Tool!" – Collaborative Use of Tailored Artifacts in Groupware**

***Volker Wulf***

In: Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP '99), 14.11. - 17.11.1999, Phoenix (Arizona), ACM-Press, New York, 1999, S. 50 - 60

## **8 „Why did that happen?“ – Building Appropriate Mental Models on Groupware Functions**

***Volker Wulf***

In: Bullinger, H.-J.; Ziegler, J. (Hrsg.): Human-Computer Interaction: Communication, Cooperation and Application Design, Lawrence Erlbaum, Mahwah, 1999, S. 338 - 342

**9 Exploration Environments: Supporting Users to Learn Groupware Functions**

***Volker Wulf***

In: Interacting with Computers, Vol. 13, No. 2, 2000, S. 265 - 299

**10 On Search for Tailoring Functions: Empirical Findings and Implications for Design**

***Volker Wulf***

In: Proceedings of OZCHI '99, 28.11. - 30.11.1999, Wagga Wagga, Australia, 1999, S. 105 - 111

**11 Design for Self-directed Learning – Evaluating the Concepts of Direct Activation and Explorative Execution**

***Volker Wulf***

Arbeitsbericht, ProSEC - Institut für Informatik III, Universität Bonn, Oktober 1999

## **Zur anpaßbaren Gestaltung von Groupware: Einleitende und zusammenfassende Bemerkungen**

Die hier als Habilitationsleistung zusammengefaßten elf Schriften befassen sich mit der benutzergerechten Gestaltung anpaßbarer Anwendungen zur computerunterstützten kooperativen Arbeit. Die Arbeit ist insofern in dem Überschneidungsbereich von Software-Ergonomie und computerunterstützter kooperativer Arbeit angesiedelt. Methodisch verbinden sich empirische Studien über die Nutzung von Computeranwendungen in verschiedenen Organisationen mit theoriegeleiteter Konzeptentwicklung, innovativen Implementierungen und deren Evaluation. Ein solches Vorgehen erscheint für die Forschung in der angewandten Informatik angemessen (vgl. Henderson 1998, Oberquelle u. a. 1999).

Software-Ergonomie und computerunterstützte kooperative Arbeit sind Teilgebiete der Informatik, die durch die Ausbreitung von Computeranwendungen in die verschiedensten Lebensbereiche und deren zunehmende Vernetzung besondere Bedeutung gewinnen. Die Software-Ergonomie erforscht theoretische Grundlagen, Konzepte, Realisierungsmöglichkeiten und Vorgehensweisen für eine menschengerechte Gestaltung computerunterstützter Tätigkeiten. Sie beschränkt sich nicht, wie häufig fälschlicherweise angenommen wird, auf die Präsentationsaspekte interaktiver Software, sondern bezieht die Gestaltung der Funktionalität und deren organisatorische Einbettung als Untersuchungsgegenstand mit in die Betrachtung ein.

Die software-ergonomische Diskussion etablierte sich international vor etwa 25 Jahren mit dem Aufkommen interaktiver Computeranwendungen. Im deutschen Sprachraum werden seit 1983 in jedem zweiten Jahr Tagungen zur Software-Ergonomie ausgerichtet. Der Fokus der deutschen software-ergonomischen Diskussion lag dabei bisher stärker auf der theoretischen – insbesondere auch arbeitswissenschaftlichen - Fundierung, der internationalen Normung und der Evaluation von Software- und Arbeitssystemen. Bedingt insbesondere durch das starke Interesse der Hersteller marktgängiger Standardprodukte hat sich der software-ergonomische Diskurs in den USA deutlich breiter und intensiver entwickelt. Im Vergleich zur bundesdeutschen Diskussion wird dabei ein stärker gestaltungsorientierter Fokus verfolgt. Den Status, den die Software-Ergonomie mittlerweile in der Informatikforschung in den USA genießt, wird beispielsweise auch dadurch deutlich, daß die SIGCHI (Special Interest Group on Computer-Human Interaction) zu einer der größten und erfolgreichsten Unterorganisationen der ACM zählt.

Mittlerweile entsteht aber auch in der BRD ein zunehmendes wissenschaftliches und industrielles Interesse an software-ergonomischer Gestaltung (z.B. Enjoy-Initiative der SAP). Diese Entwicklung wird in dem von mir mitverfaßten Memorandum „Mensch & Computer 2000“ aufgegriffen. Es ruft zu einer Intensivierung - insbesondere der gestaltungsorientierten – Forschungsanstrengungen auf. Neuartige, durch die zunehmende Verbreitung von Computeranwendungen entstehende Fragestellungen sind evolutionär und unter Beteiligung von Nutzern zu erforschen (vgl. Oberquelle u. a. 1999).

Computerunterstützte kooperative Arbeit bezeichnet auch ein Forschungsgebiet der angewandten Informatik. Es beschäftigt sich mit den theoretischen Grundlagen, Konzepten, Realisierungsmöglichkeiten und Vorgehensweisen der Unterstützung von Kommunikation, Koordination und Kooperation zwischen Nutzern bei der Arbeit. Die in diesem Feld entstehende Software wird als Groupware bezeichnet. Wesentlich jünger als die Software-Ergonomie hat sich dieses Forschungsgebiet in der zweiten Hälfte der 80er Jahre in den USA geformt. Im deutschen Sprachraum finden Tagungen zu diesem Thema in mittlerweile zwei jährlichem Rhythmus seit 1991 statt. Im Gegensatz zur Software-Ergonomie finden sich im Bereich computerunterstützter kooperative Arbeit sehr viel geringere Diskrepanzen in Forschungsfokus und -methodik zwischen dem deutschsprachigen und dem angelsächsisch geprägten internationalen Diskurs.

Software-Ergonomie und computerunterstützte Gruppenarbeit weisen erhebliche thematische Überschneidungen auf. Wissenschaftshistorisch betrachtet läßt sich - insbesondere im deutschen Sprachraum - feststellen, daß viele Forscher im neu entstandenen Feld computerunterstützter Gruppenarbeit software-ergonomisch geprägt waren. Dies trifft auch auf mich und meine Arbeiten zu. Die hier behandelte Fragestellung der anpaßbaren Gestaltung von Groupware hat eine software-ergonomische Tradition insofern als die Forderung nach Individualisierbarkeit von Anwendungen auf Grund von intra- und interindividuell unterschiedlichen Nutzungsweisen ein wichtiges Ergebnis der arbeitswissenschaftlichen Fundierung der Software-Ergonomie darstellt (vgl. Ulich 1978, Ackermann und Ulich 1987). Im Hinblick auf die Gestaltung von Groupware ist das Konzept der Individualisierbarkeit allerdings nur bedingt tragfähig (vgl. Oberquelle 1994). Individuell und kollektiv wirksame Anpassungen müssen in Bezug zueinander gesetzt werden, um kooperative Nutzung der Anwendungen zu ermöglichen. Auch die mit Anpaßbarkeit eng verknüpfte Frage nach einer lernförderlichen Gestaltung von Computeranwendungen wurde in der software-ergonomischen Diskussion bereits früh aufgeworfen (z.B. Darlington, Dzida und Herda 1983; Carroll und Mack 1983, 1984; Herrmann 1986). Allerdings wirft die lernförderliche Gestaltung von anpaßbaren und vernetzten Anwendungen neue Problemstellungen auf.

Vor dem Hintergrund des in den Forschungsgebieten Software-Ergonomie und computerunterstützter Gruppenarbeit bereits gegebenen Diskussionsstandes bin ich in den in dieser Arbeit zusammengefaßten Schriften den folgenden Forschungsfragen nachgegangen:

- Gibt es bisher in der Literatur nicht thematisierte Gründe, die eine anpaßbare Gestaltung von Groupware notwendig erscheinen lassen?
- Wie sollte der Entwicklungs- und Einführungsprozeß anpaßbarer Groupware gestaltet sein?
- Welche Anforderungen bestehen hinsichtlich der anpaßbaren Gestaltung ausgewählter Groupwarefunktionalitäten und wie lassen sich die aus den Anforderungen abgeleiteten Konzepte auf innovative Weise implementieren?
- Welche Konzepte der Dialogschnittstelle erleichtern das Erlernen und die Nutzung anpaßbarer Groupware?

Zur Bearbeitung dieser Forschungsfragen bot das POLITeam- Forschungsvorhaben einen geeigneten Kontext. Für dessen inhaltliche Ausgestaltung war ich am Projektbereich Software-Ergonomie und CSCW des Instituts für Informatik III der Universi-

tät Bonn in den Jahren 1994 – 1998 verantwortlich. Gemeinsam mit der gedas GmbH, Berlin, und dem GMD - Forschungszentrum Informationstechnik GmbH, St. Augustin, wurde in diesem Projekt eine marktgängige Groupware in einem Bundesministerium und Teilen einer Landesregierung eingeführt und entsprechend den Anforderungen der Nutzer in mehreren Zyklen weiterentwickelt. Durch die Möglichkeit, die Nutzung verschiedener Versionen einer Groupware über einen Zeitraum von insgesamt vier Jahren in den Anwendungsfeldern zu verfolgen, bot dieses Projekt günstige Voraussetzungen, Einführungsprozesse und Nutzungsformen von Groupware zu untersuchen, Anforderungen an anpaßbare Systemgestaltung abzuleiten und innovative Implementierungen zu evaluieren.

Geleitet durch die obigen Forschungsfragen wurden die folgenden Beiträge zum wissenschaftlichen Fortschritt in den Feldern Software-Ergonomie und computerunterstützter kooperativer Arbeit geleistet. Zunächst habe ich die für den wissenschaftlichen Diskurs notwendigen terminologischen Grundlagen weiterentwickelt. Dazu habe ich das IFIP-Modell der Benutzungsschnittstelle von Computeranwendungen um die für die Beschreibung von anpaßbarer Groupware notwendigen Aspekte erweitert.

In zwei empirischen Langzeituntersuchungen des Einführungsprozesses von Groupware in mehreren Organisationen bin ich auf Befunde gestoßen, die die Notwendigkeit einer anpaßbaren Gestaltung von Groupware untermauern. Zum einen ließen sich bedingt durch den Einsatz von Groupware Organisationsveränderungen feststellen, die zum Zeitpunkt der Einführung weder vorhergesehen noch geplant werden konnten. Zum anderen zeigte sich die Bedeutung von Konventionen für die gemeinsame Nutzung von Groupware und deren Wandel in längerfristigen Verhandlungsprozessen. Diese Befunde lassen sich mit der Theorie selbstorganisierter sozialer Systeme deuten. Da sowohl durch die Etablierung von Nutzungskonventionen als auch durch die Veränderung von Arbeitsprozessen neue Anforderungen an die Gestaltung von Groupware entstehen können, weisen diese Befunde auf die Notwendigkeit technisch flexibler Gestaltung von Groupware hin.

Die Einführung von Groupware in Organisationen benötigt ein Prozeßmodell, um die Aktivitäten der verschiedenen Beteiligten zu orientieren. Deshalb habe ich das Konzept *integrierter Organisations- und Technikentwicklung* ausgearbeitet. Es thematisiert die Entwicklung und den Einsatz anpaßbarer Anwendungen explizit und erweitert das STEPS-Modell des evolutionären und partizipativen Software-Engineerings (vgl. Floyd u. a. 1989) um arbeitspsychologische Methoden und Aspekte der Personal- und Organisationsentwicklung. Als Prozeßmodell leitet es die Entwicklung und Einführung anpaßbarer Groupware in Organisationen und berücksichtigt nicht antizipierte Organisationsveränderungen ebenso wie längerfristige Verhandlungsprozesse um Nutzungskonventionen.

Das Prozeßmodell zur Entwicklung von Groupware hat den Entwurf dreier anpaßbarer Groupwarefunktionalitäten geleitet: eines Tools zur Spezifikation von Zugriffsrechten, eines Suchtools und eines Ereignisdienstes. Ausgehend von Problemen, die Nutzer im Umgang mit nicht ausreichend flexiblen Funktionen hatten, wurden dabei insbesondere die folgenden auch auf andere Kontexte übertragbaren Konzepte ausdifferenzierten Systemverhaltens von Groupware entwickelt:

*Verfeinerte Zugriffsrechte jenseits der Erlaubnis/Verbots-Dichotomie:* Um die Spezifikation von Zugriffsrechten über die Erlaubnis oder das Verbot des Zugriffs hinaus zu flexibilisieren, habe ich zwei weitere Formen der Zugriffsvergabe entwickelt. Zum einen kann der Zugriffsgewährende bestimmten Nutzern Zugriff er-

lauben unter der Maßgabe, darüber automatisch informiert zu werden. Zum andern kann der Zugriffsgewährende den Zugriff zunächst verweigern unter der Maßgabe, zum Zeitpunkt des Zugriffswunsches darüber jeweils entscheiden zu können.

*Einstellbare Fenster zur Ergebnisanzeige in einem Suchtool:* Um die Auswertung von Suchergebnissen zu erleichtern, habe ich flexibel einstellbare Fenster zur Anzeige von Suchergebnissen unterschiedlichen Typus konzipiert.

*Herstellerfilter eines Ereignisdienstes:* Um den Verlust der Privatsphäre bei der Nutzung eines automatisch Benachrichtigungen generierenden Ereignisdienstes zu vermeiden, habe ich das Konzept eines flexibel anpaßbaren Herstellerfilters entwickelt.

Die Implementierung des Zugriffsvergabetools und des Suchtools erfolgte unter Rückgriff auf die Programmierschnittstelle einer marktgängigen Groupware. Dabei wurde für die Spezifikation der Zugriffsrechte ein innovativer regelbasierter Ansatz entwickelt. Die Implementierung des Suchtools ist insofern innovativ, als die Anwendung während der Laufzeit in rekonponierbare Komponenten zerlegt wurde. Die Rekomposition erfolgt mittels einer graphischen Benutzungsoberfläche. Die Implementierung des anpaßbaren Ereignisdienstes erfolgt auf Basis eines gemeinsamen Ablagebereichs zum Austausch von Anpassungsdaten. Dieser wurde in Erweiterung einer marktgängigen Textverarbeitung entwickelt.

Empirische Untersuchungen zu Problemen im Einsatz anpaßbarer Anwendungen zeigten, daß die Gestaltung der Groupwarefunktionalität allein nicht ausreicht, sondern daß sie durch eine geeignete Gestaltung der Dialogschnittstelle einer Anwendung zu ergänzen ist. Zur Linderung der empirisch festgestellten Defizite habe ich die folgenden Konzepte entwickelt:

*Direkte Aktivierbarkeit:* Die Umsetzung dieses Konzepts unterstützt Nutzer dabei, Anpassungsfunktionen in dem Moment zu finden, in dem eine Anpassungsnotwendigkeit auftritt.

*Explorative Ausführbarkeit:* Die Umsetzung dieses Konzepts erlaubt es Nutzern, an ihrer Benutzungsschnittstelle auszuprobieren, wie sich eine von ihnen ausgeführte Anpassung auf das Systemverhalten bei andern Nutzern auswirkt.

Die beiden Konzepte der Dialogschnittstelle wurden auf Basis der oben genannten Funktionalitäten anpaßbarer Groupware implementiert und evaluiert.

## **1. Terminologische Klärung – Das erweiterte IFIP-Modell zur Beschreibung der Benutzungsschnittstelle anpaßbarer Groupware**

### *Beiträge:*

Exploration Environments: Supporting Users to Learn Groupware Functions, submitted to: Interacting with Computers

Tailoring Groupware for Different Scopes of Validity, in: Behaviour & Information Technology, Vol. 18, No. 3, 1999, S. 199 - 212

“Let’s see your Search-Tool!”– Collaborative Use of Tailored Artifacts in Groupware, in: Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP `99), ACM-Press, New York 1999 (in press)

On Search for Tailoring Functions: Empirical Findings and Implications for Design, in: Proceedings of OZCHI '99, November 28 - 30, 1999, Wagga Wagga, Australia, 1999 (in press)

Wissenschaftlicher Diskurs benötigt klare Begrifflichkeiten. In mehreren meiner Beiträge habe ich mich um die Weiterentwicklung einer problemangemessenen Terminologie für die Gestaltung anpaßbarer Groupware bemüht. Diese Terminologie soll im folgenden zusammenfassend dargestellt werden.

Ich benutze den Begriff *Groupware* in Abgrenzung zu Einzelplatzanwendungen in einem breiten Sinne. Es sollen damit Computeranwendungen bezeichnet werden, die ihre Nutzer bei Aufgaben der Kommunikation, Kooperation und Koordination durch technische Funktionen unterstützen. Im Gegensatz zu Ansätzen, die bei der Definition von Groupware einen qualifizierten Begriff kooperativer Arbeit voraussetzen (vgl. Oberquelle 1991), hat sich in der Diskussion um computerunterstützte kooperative Arbeit (im Englischen: Computer Supported Cooperative Work (CSCW)) obige breite Definition durchgesetzt (Bannon und Schmidt 1991; Greenberg 1991).

*Anpaßbarkeit* soll hier als die Veränderbarkeit der Funktionalität eines Systems als Reaktion auf lokale Bedürfnisse während des Einsatzes im Anwendungsfeld verstanden werden.<sup>1</sup> In der Literatur finden sich diverse Aufzählungs- und Klassifikationsansätze für die verschiedenartige Ausprägung von Anpassungsfunktionen (vgl. Oberquelle 1993; Herrmann und Just 1994; Mørch 1997). Henderson und Kyng (1991) unterteilen Anpassungsfunktionen hinsichtlich der Komplexität ihrer Aktivierung in die folgenden drei Stufen: Auswahl zwischen Alternativen antizipierten Verhaltens, Zusammenbauen neuen Verhaltens aus existierenden Modulen, Veränderung des Artefaktes. Im weiteren Verlauf der Arbeit werde ich im wesentlichen auf Anpassungsfunktionen des ersten und zweiten Typus fokussieren, weil diese Nutzer ohne Programmierkenntnissen am ehesten ansprechen.

Anpaßbarkeit ist eine Ausformung des übergeordneten Konstrukts „technische Flexibilität“. Technische Flexibilität wird als eine Eigenschaft von Software verstanden, die es menschlichen Akteuren erlaubt, die aus der Ausführung einer bestimmten Funktion resultierenden Zustandsübergänge an sich wandelnde Anforderungen der Nutzer anzupassen. Man kann im wesentlichen drei Ausprägungen technischer Flexibilität unterscheiden:

*Vielfältigkeit der Nutzung* erlaubt es Nutzern im Rahmen der Aktivierung einer Funktion Parameter anzugeben, die eine von mehreren möglichen Funktionsalternativen spezifizieren (vgl. Oppermann 1989).

*Anpaßbarkeit* ist realisiert durch eine zusätzliche Funktion (Anpassungsfunktion), die a priori ausgeführt, die Art der Ausführung der Nutzungsfunktion beeinflusst.

*Entwicklung einer neuen Systemversion* hat zur Folge, daß der bei der Ausführung einer Funktion bzw. von Funktionsalternativen bewirkte Zustandsübergang neu definiert oder neue Funktionen bzw. Funktionsalternativen implementiert werden.<sup>2</sup>

---

1 Im deutschen werden die Termini Adaptierbarkeit (vgl. Friedrich 1990; Haaks 1992) und Konfigurativität (vgl. Paetau 1993), im englischen "tailorability" (vgl. Trigg u.a. 1987; Henderson und Kyng 1991), "customization" (vgl. Bentley und Dourish 1995) "end-user modifiability" (vgl. Fischer und Girgensohn 1991) in ähnlichem Sinne verwandt.

<sup>2</sup> Ein besonderer Fall der Weiterentwicklung ist die vollständige Entfernung einer Anwendung und deren Ersatz durch eine neue Anwendung.

Die Abgrenzung der Anpassung von der Nutzung einerseits und von der Entwicklung einer neuen Systemversion andererseits ist nicht immer eindeutig zu treffen. Henderson und Kyng (1991, S. 223ff.) schlagen die folgenden Abgrenzungsmerkmale zwischen Anpassung und Nutzung vor:

- Anpassung verändert Aspekte der Funktionalität, die aus Sicht der Nutzer während der „normalen“ Nutzung stabil sind,
- Anpassung verändert nicht die Datenobjekte, sondern die Funktionen,
- Anpassung bewirkt immer länger andauernde Veränderungen, während es sich bei nur kurzfristig gültigen Effekten um Nutzung handelt.

Im folgenden soll insbesondere die Tatsache der Veränderung von Nutzungsfunktionen als Charakteristikum von Anpassungsfunktionen herangezogen werden. Die Geltungsdauer von Anpassungen und damit die Häufigkeit der Nutzung von Anpassungsfunktionen erscheint dagegen ein eher weniger geeignetes Unterscheidungskriterium zu sein. Zum einen wird diese durch den Anwendungskontext bestimmt und ist deshalb bei der Herstellung kaum antizipierbar. Zum anderen kann die Häufigkeit der Nutzung von Anpassungsfunktionen durch eine ungeeignete Gestaltung der Benutzungsschnittstelle beeinträchtigt werden.

Henderson und Kyng (1991, S. 223ff.) unterscheiden Anpassung von Herstellung dadurch, daß sie unterstellen, daß bei Anpassung keine neue, sondern eine auf die lokalen Bedürfnisse angepaßte Systemversion entsteht. Geht man von evolutionärer Systementwicklung im Anwendungsfeld aus, dann ist diese Abgrenzung unscharf, weil auch Programmieraktivitäten beliebiger Komplexität in dieser Definition subsumiert werden. Ich plädiere deshalb eher für eine an der software-technischen Komplexität und den sie durchführenden Akteuren orientierte Unterscheidung. Anpassungen sollten von Nutzern bzw. lokalen Experten durchführbar sein. Deshalb müssen sie eine von der normalen Entwicklungsumgebung unterschiedene Benutzungsschnittstelle mit speziell für die Anpassung gestalteten Funktionen aufweisen.

Zur Entwicklung einer geeigneten Terminologie, die zur Beschreibung einer solchen Benutzungsschnittstelle dient, bin ich vom IFIP-Modell ausgegangen. Das IFIP-Modell ist ein in der deutschsprachigen software-ergonomischen Diskussion weit verbreiteter Ansatz, die Benutzungsschnittstelle von Computeranwendungen zu untergliedern. In seiner ursprünglichen Version unterscheidet das IFIP-Modell zwischen: Ein- und Ausgabe-, Dialog-, Anwendungs- und Organisationsschnittstelle (Dzida 1983 und 1988). Die Organisationsschnittstelle ist im Hinblick auf eine Beschreibung verteilter Systeme ausdifferenziert worden. Oppermann u.a. (1992) unterscheiden hinsichtlich der Organisationsschnittstelle einen technischen und einen nicht-technischen Aspekt. Die technische Organisationsschnittstelle bildet dabei die aus der Perspektive der Nutzer relevanten Leistungsmerkmale verteilter Systeme ab, während der nicht-technische Teil der Organisationsschnittstelle die organisatorische bzw. soziale Einbettung der Software thematisiert.<sup>3</sup>

---

<sup>3</sup> Aufbauend auf den Arbeiten von Ellis and Wainer (1994) hat Teege (1998) ein Systemstrukturmodell für Groupware entwickelt. Dort wird zwischen der Benutzungsoberfläche, dem Organisationsmodell, dessen Notation und Speicherung, dem Zugriff auf dieses Modell und darauf aufsetzenden Diensten unterschieden. Im Vergleich zum IFIP-Modell unterscheidet dieser Ansatz nicht zwischen der Ein- und Ausgabeschnittstelle und der Dialogschnittstelle. Er bietet dagegen eine mögliche Klassifikation der Funktionalität auf der Ebene der technischen Organisationsschnittstelle. Diesbezüglich habe ich einen alternativen Klassifikationsansatz im Rahmen meiner Dissertation entwickelt (vgl. Wulf 1997).

Eine Schwäche des IFIP-Modells liegt darin, daß es über die Struktur der einzelnen Ebenen der Schnittstelle keine Aussage macht. Insbesondere werden die von der Funktionalität jeweils manipulierten Daten nicht modelliert (vgl. Höller 1993, Rauterberg 1995). Diese sind aber von großer Relevanz für die Gestaltung der Benutzungsschnittstelle (vgl. Akscyn, Yoder und McCracken 1988; Höller 1993).

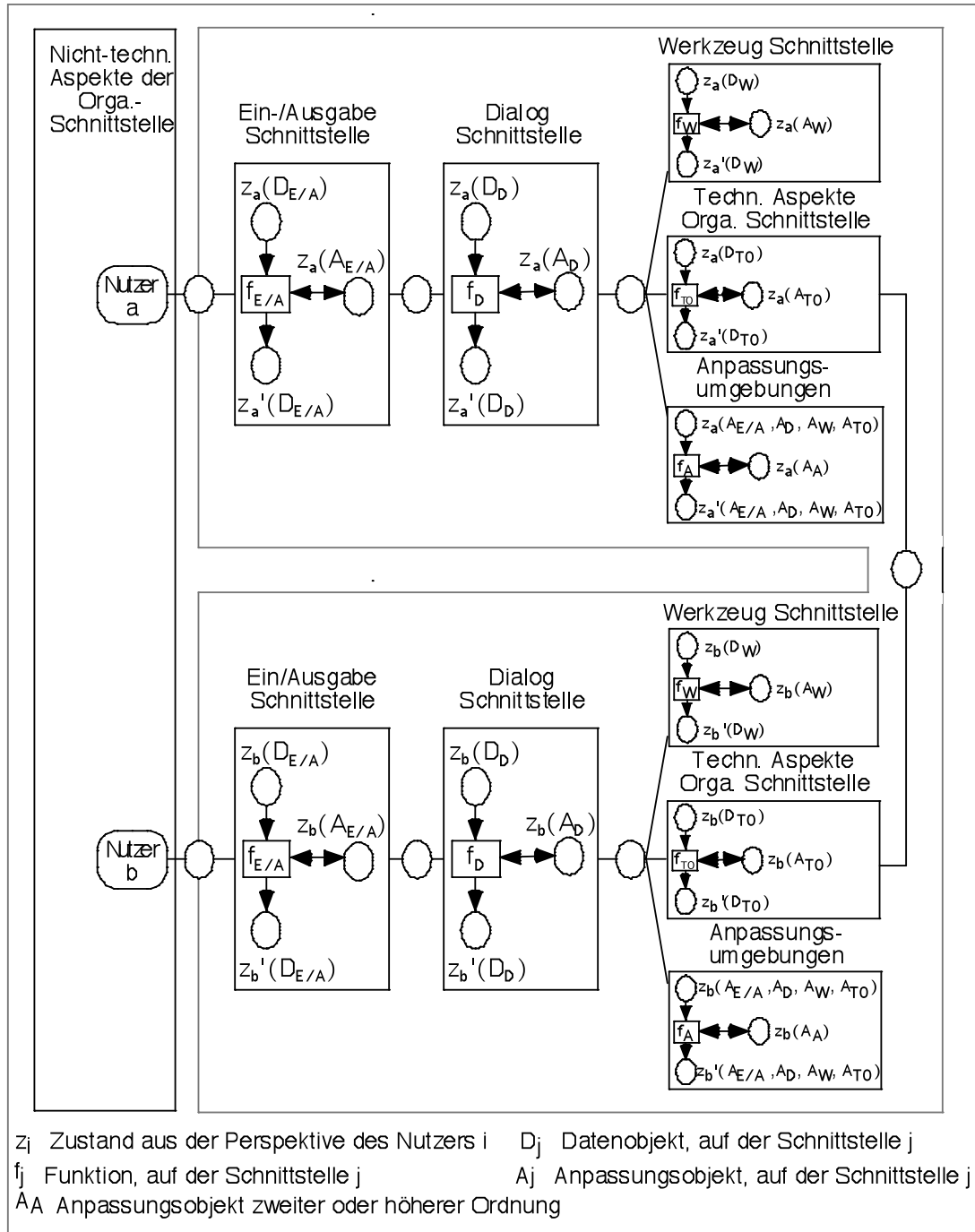


Abb. 1: Das erweiterte IFIP-Model zur Beschreibung anpaßbarer Groupware

Ich unterscheide deshalb auf den technischen Ebenen des IFIP-Modells zwischen Funktionen und Datenobjekten. Die Ausführung von Funktionen verändert den Zustand der Datenobjekte. Hinsichtlich der Funktionen unterscheide ich zwischen normalen und getriggerten (vgl. Oppermann und Simm 1994). Normale Funktionen

werden durch die Eingabe eines Befehls durch einen Nutzer aktiviert. Getriggerte werden durch das Auftreten eines vorsezifizierten Systemereignisses aktiviert. Darüber hinaus unterscheide ich zwischen Nutzungs- und Anpassungsfunktionen. Die Ausführung von Anpassungsfunktionen erzeugt, modifiziert oder löscht Anpassungsobjekte, die die Ausführung von Nutzungsfunktionen determinieren. Das Verhalten von getriggerten Funktionen wird insofern durch Anpassungsfunktionen determiniert. Anpassungsfunktionen sind immer normale Funktionen.

Die Existenz normaler Funktionen wird in bestimmten Dialogzuständen an einzelnen Stellen der Ausgabeschnittstelle dargestellt. Für die Befehlseingabe sind ebenfalls bestimmte Stellen vorgesehen. Bei Anwendungen, die gemäß den Prinzipien der direkten Manipulation entwickelt sind, fallen die beiden Stellen in der Regel räumlich zusammen. Rauterberg (1995) nennt diese für die Aktivierung von normalen Funktionen zentralen Stellen der Ein- und Ausgabeschnittstelle funktionale Interaktionspunkte.

Basierend auf diesen Vorüberlegungen habe ich das IFIP-Modell erweitert, um die speziellen Aspekte anpaßbarer Groupware darzustellen (vgl. Abb. 1). Neben der Ein/Ausgabe-, Dialog-, Werkzeug- sowie der technischen und nicht technischen Organisationsschnittstellen werden die Anpaßungsumgebungen als eigenständiger Bestandteile der Benutzungsschnittstelle modelliert. Auf jeder Schnittstelle des Modells beeinflußt der Zustand der *Anpassungsobjekte* die Ausführung der anzupassenden Funktionen (vgl. Oberquelle 1994). Dieses Faktum ist in Abb. 1 durch den Doppelpfeil ausgedrückt, mittels dessen der Zustand der Anpassungsobjekte auf den durch die Funktionsausführung bewirkten Zustandsübergang einwirkt. Anpassungsobjekte werden in den Anpassungsumgebungen von Anpassungsfunktionen erzeugt, modifiziert und gelöscht.

Die Aktivierungsmöglichkeiten von Anpassungsfunktionen können selbst wieder von Anpassungsobjekten zweiter (oder höherer) Ordnung beeinflusst werden. Dieses Faktum ist in Abb. 1 durch den Doppelpfeil ausgedrückt, mittels dessen die Anpassungsobjekte zweiter (und höherer) Ordnung auf die Ausführung der Anpassungsfunktionen einwirken. Diese Anpassungsobjekte werden mittels *Anpassungsfunktionen zweiter Ordnung* erzeugt, modifiziert oder gelöscht. Wenn die Anpassungsfunktionen niedriger Ordnung aus Sicht der Nutzer weniger komplex zu handhaben sind als die höherer Ordnung, spreche ich von *gestuften Anpassungsfunktionen*. Gestufte Anpassungsfunktionen sind ein Mittel, die Komplexität der Handhabung von Anpassungsfunktionen niedriger Ordnung gering zu halten und schrittweises Lernen zu ermöglichen (vgl. MacLean u. a. 1990).

Teil der Erzeugung oder Modifikation eines Anpassungsobjekts kann die Spezifikation eines bestimmten *Geltungsbereichs* sein. Der Geltungsbereich eines Anpassungsobjekts ist eine Teilmenge der bei der Aktivierung der anzupassenden Funktion automatisch unterscheidbaren Nutzungssituationen. Bei der Ausführung der anpaßbaren Funktion wird dann automatisch das für die jeweilige Nutzungssituation spezifische Anpassungsobjekt ausgewählt. Die Spezifikation differenzierter Geltungsbereiche von Anpassungsobjekten erhöht die Komplexität von Anpassungsaktivitäten und legt die Implementierung gestufter Anpassungsfunktionen nahe.

## **2. Groupware in Anwendung: Organisationsveränderungen und sich wandelnde Nutzungskonventionen**

### *Beiträge:*

Evolving Cooperation when Introducing Groupware – A Self-Organization Perspective, in: *Cybernetics & Human Knowing*, Vol. 6, No. 2, 1999, S. 55 - 73

Storing and Retrieving Documents in a Shared Workspace: Experiences from the Political Administration, in: Howard, S.; Hammond, J.; Lindgaard, G. (eds): *Human Computer Interaction: INTERACT 97*, Chapman & Hall 1997, S. 469 – 476

Da mit der der Herstellung und dem Einsatz anpaßbarer Groupware erheblicher Aufwand einhergeht, ist die Notwendigkeit anpaßbarer Gestaltung näher zu begründen. Da betriebliche Organisationen ein wesentliches Anwendungsfeld von Groupware darstellen, habe ich die Einführung von Groupware in zwei Anwendungsfeldern in einer Langzeitstudie empirisch untersucht und analysiert. In der CSCW-Literatur finden sich solche Langzeitstudien bisher kaum (aber: Karsten und Jones 1998, Orlikowski 1997). Als Ergebnis dieser Untersuchung haben sich zwei Sachverhalte herausgestellt, die eine anpaßbare Gestaltung von Groupware motivieren. Zum einen lassen sich die mit der Einführung von Groupware einhergehenden, ungeplanten Organisationsveränderungen zum Zeitpunkt der Groupwareeinführung nicht vollständig antizipieren. Aus anfänglich weder intendierten noch vorherzusehenden Veränderungen können sich Anforderungen an eine technisch flexible Gestaltung der Groupware ergeben. Zum anderen hat sich gezeigt, daß der Gebrauch von Groupware die Etablierung von Nutzungskonventionen erfordert (vgl. Wulf und Mark 1997, Mark u. a. 1997). Diese Konventionen haben sich in dem hier betrachteten Fall als Ergebnis längerfristiger Verhandlungsprozesse mehrfach verändert. Wenn Nutzer bei der Umsetzung der Konventionen auf Groupwarefunktionen zurückgreifen, so ergibt sich aus der dynamischen Entwicklung von Konventionen ein Bedarf nach technischer Flexibilität. Die genannten empirischen Befunde werden mit der Theorie selbstorganisierter sozialer Systeme gedeutet (vgl. Maturana und Varela 1980 und 1987; Luhmann 1984). Die Anwendung von Selbstorganisationstheorien zur Analyse von Einführungsprozessen für Groupware wird meines Wissens in dieser Arbeit erstmals versucht.

Der Aufsatz "*Evolving Cooperations when Introducing Groupware: A Self-Organization Perspective*" ist in seinen Grundzügen bereits Anfang 1996 entstanden. Er beschreibt das erste Jahr des Einführungsprozesses einer Groupware in zwei Organisationen der politischen Verwaltung (ein Bundesministerium und eine Landesvertretung am Sitz der Bundesregierung).

Vor dem Hintergrund der organisationswissenschaftlichen Diskussion um posttayloristische Arbeitsformen wird insbesondere auf Veränderungen von Arbeitsteilung und Koordination eingegangen. Dabei wird sowohl der Prozeß als auch das Ergebnis des organisatorischen Wandels beleuchtet. Es wird untersucht, inwiefern die beschriebenen Einführungsprozesse mittels der Theorie selbstorganisierter sozialer Systeme interpretiert werden können. Hinsichtlich des Ergebnisses des organisatorischen Wandels im Kontext der Einführung von Groupware wird gefragt, inwiefern sich die Fähigkeit der Organisationen zu selbstorganisiertem Verhalten erhöht hat.

Hinsichtlich der systemtheoretischen Grundlagen beziehe ich mich auf die chilenischen Biologen H. Maturana und F. Varela sowie die Arbeiten des deutschen Soziologen N. Luhmann. Maturana und Varela (1980 und 1987) nennen ein System auto-

poietisch, wenn es in der Lage ist, sich durch Interaktion seiner internen Elemente zu reproduzieren. Autopoietische Systeme sind selbstorganisiert in dem Sinne, daß ihr Verhalten nicht von ihrer Umwelt determiniert werden kann (operationale Geschlossenheit). Diese Systeme stehen zwar in Austauschbeziehung zur Umwelt, es gibt aber keine deterministische Input-/Outputbeziehungen. Darin unterscheidet sich dieser Ansatz von der traditionellen Sichtweise der Systemtheorie. Luhmann (1984) wendet entgegen der Bedenken von Maturana und Varela das Konzept der Autopoiesis auf soziale Systeme an. Soziale Systeme unterscheiden sich dabei von anderen Systemen durch ihren Operationsmodus: Kommunikation.

Die Untersuchung des Einführungsprozesses einer Groupware in eine Landesvertretung und ein Bundesministerium zeigt, daß sich Organisationen für externe Beobachter wie operational geschlossene Systeme verhalten. Trotz umfangreicher Untersuchungen vor der Einführung der Groupware wurde der für die Kooperationsunterstützung bestgeeignete Arbeitsprozeß in der Landesvertretung nicht entdeckt. Erst nachdem die Anwendung sich einige Wochen im Einsatz befand, wurde bei einer eher zufälligen Diskussion zwischen Betreuern und Beschäftigten ein interner Abstimmungsprozeß über das Verhalten des Landes im Bundesrat als für die Groupwareunterstützung geeignet befunden. Um diesen Prozeß zu unterstützen, mußte die in der ersten Projektphase nur begrenzt zur Verfügung stehende Hardware in der Organisation neu verteilt werden. Außerdem zeigte sich nach einiger Zeit, daß einzelne Nutzer auf Grund der Besonderheiten des jetzt neu unterstützten Kooperationsprozesses neue Anforderungen an die Dokumentverweiskfunktion hatten. Diese wurden von der eingeführten Systemversion nicht erfüllt. Insofern führte der bei der Einführung nicht antizipierte Einsatz der Groupware zu selbst zum Zeitpunkt der Einführung nicht vorhergesehenen technischen Anforderungen.

Durch diese Prozeßinnovationen wurde die Koordination durch Selbstabstimmung zwischen den Nutzern erheblich erleichtert. Diese Erleichterung der Koordination führte zu einer Veränderung der zwischenmenschlichen Kommunikation (vgl. Mark und Wulf 1999). Dieser eher langfristige Effekt war weder bei der Einführung der Groupware noch bei der Umsetzung der Prozeßinnovation in seinem Umfang antizipierbar.

Im Gegensatz zur Landesvertretung wurden im Bundesministerium die Möglichkeiten zur Prozeßinnovation während der Voruntersuchung bereits erkannt und in einem Workshop mit den Mitarbeitern ausgearbeitet. Die Dokumentverweiskfunktion erlaubte die Einrichtung gemeinsamer Arbeitsbereiche innerhalb des beteiligten Fachreferats und zwischen Fachreferat und zentralem Schreibdienst. Daraus ergab sich eine im Projektverlauf zunehmende Verlagerung der Schreibarbeit vom Schreibdienst in das Fachreferat. Mit verstärkter Nutzung des Systems forderten die Nutzer einen verbesserten Ereignisdienst, um über den Stand der Bearbeitung gemeinsam erstellter Dokumente besser informiert zu werden. Diese bei der Einführung der Groupware nicht antizipierte Forderung konnte im Rahmen der gegebenen Funktionalität nur unzureichend realisiert werden. Die intensive Nutzung der gemeinsamen Arbeitsbereiche führte zu einer raschen Zunahme der Anzahl der dort gespeicherten elektronischen Dokumente. Daraus ergaben sich neue, bisher von einer weiteren Organisationseinheit, der Registratur, wahrgenommene Aufgaben im Fachreferat: die Ablage und Klassifizierung von Dokumenten. Mangelnde Motivation der Beschäftigten und fehlende Unterstützungsfunktionalität verhinderte letztendlich die Einbeziehung der Registratur in das Projekt. Insofern waren Art und Umfang der Verschiebung von

Aufgaben zwischen Referat und Registratur und zwischen Referat und Schreibdienst bei Einführung der Groupware nur schwer abzusehen.

Betrachtet man die Ergebnisse der empirischen Studien, so können die einführenden Anwenderorganisationen als selbstorganisierte soziale Systeme interpretiert werden. Weder ließen sich alle Anwendungsmöglichkeiten der Groupware zum Zeitpunkt ihrer Einführung antizipieren noch konnten die aus der Anwendung resultierenden Konsequenzen hinsichtlich Arbeitsteilung, Koordination und zwischenmenschlicher Kommunikation genau vorhergesagt werden. Insofern läßt sich die Einführung einer Groupware in eine Organisation als Irritation eines operational geschlossenen sozialen Systems beschreiben, deren Effekte von außen nicht determiniert werden können (vgl. Luhmann 1984).

Um vor diesem Hintergrund Einführungsprozesse beeinflussen zu können, ist ein evolutionäres Vorgehen unabdingbar. Der zu Beginn nicht antizipierbare Verlauf von Einführungsprozessen führt im Laufe des Einsatzes zu neuen Anforderungen an die Groupware. Um solchen Anforderungen der Nutzer gerecht werden zu können, ist es erforderlich, daß die Groupware in hohem Maße technisch flexibel ist. Die Funktionalität der im Projekt eingesetzten Groupware erfüllte verschiedene bei der Einführung nicht antizipierbare Anforderungen. Allerdings entsprachen der Ereignisdienst und die Dokumentverweisfunktion nicht den Anforderungen der Nutzer. Sie mußten unter Rückgriff auf die verfügbare Anwendungsprogrammierschnittstelle reimplementiert werden. Damit ging ein erheblicher Entwicklungsaufwand einher, der bei anpaßbarer Gestaltung der Funktionen geringer ausgefallen wäre und von Nutzern oder Nutzerbetreuern im Anwendungsfeld zu leisten gewesen wäre.

Neben einem evolutionären Vorgehensmodell und technischer Flexibilität trägt auch die Partizipation der Nutzer zu einer erfolgreichen Groupware-Einführung bei. Ohne deren Kenntnisse des konkreten Arbeitsprozesses im Anwendungsfeld ließe sich das durch die Groupware gegebene Potential zur Prozeßinnovation kaum nutzen. Daneben müssen Akteure am Einführungsprozeß beteiligt sein, die die Funktionalität der Groupware überblicken. Die Entdeckung des verbesserten Abstimmungsprozesses in der Landesvertretung ist ein Beispiel für das kreative Zusammenwirken von Trägern dieser beiden Kompetenzen. Darüber hinaus haben sich die Fähigkeit, Konflikte zwischen Nutzern zu moderieren, und Erfahrung mit ähnlichen Einführungsprozessen als für die Entstehung organisatorischer Innovationen förderliche Faktoren erwiesen. Diese Maßnahmen können systemtheoretisch eher als Unterstützung interner selbstorganisierter Prozesse, denn als externe Irritation verstanden werden.

Meine Untersuchung deutet insofern darauf hin, daß selbstorganisierte soziale Systeme eine interessante Basis für das Verständnis von Einführungsprozessen von Groupware bilden. Insbesondere lassen sich auf diese Weise die für die Motivation von Anpaßbarkeit wichtigen nicht antizipierten Organisationsveränderungen erklären.

Der Tagungsbeitrag "*Storing and Retrieving Documents in a Shared Workspace: Experiences from the Political Administration*" ist Ende 1996 entstanden. In diesem Aufsatz wird das Suchverhalten von Nutzern und die Bildung von Konventionen zur Ablage von elektronischen Dokumenten in gemeinsamen Arbeitsbereichen über einen Zeitraum von zwei Jahren untersucht. Die Studie fand in denselben Unterorganisationen des Bundesministeriums statt wie die zuvor dargestellte Untersuchung. Empirische Langzeitstudien über das kollektive Such- und Ablageverhalten in gemeinsamen Arbeitsbereichen und die Bildung von Konventionen fanden sich bis dato nicht in der Literatur. Es lagen bereits Untersuchungen vor, die die Differenziertheit individueller

Ablageformen für papiergebundene und elektronische Dokumente belegten. Darüber hinaus waren empirische Ergebnisse publiziert, die bezogen auf Einzelplatzanwendungen darauf hindeuten, daß ortsbezogene Suche (entlang von Verzeichnisstrukturen) im Arbeitsalltag eine wichtigere Rolle spielt als logische Suche (mit einer Suchmaschine) (vgl. Barreau und Nardi 1995 und 1997; aber: Fertig, Freeman und Gelernter 1996).

Wenn diese Ergebnisse sich als zutreffend erwiesen, mußten Ablage und Suchen in gemeinsamen Arbeitsbereichen zu erheblichen Problemen führen. Wir untersuchten deshalb das Such- und Ablageverhalten der Nutzer einer Groupware, die gemäß der Desk-Top-Metapher gestaltete gemeinsame Arbeitsbereiche bietet. Es zeigte sich, daß die Nutzer eine starke Präferenz für ortsbezogene Suche hatten. Daraus ergab sich die Notwendigkeit, Konventionen für eine gemeinsame Ablagestruktur zu bilden.

Vor Nutzung der Groupware wurden alle Schriftstücke des Ministeriums vom zentralen Schreibdienst am Rechner erstellt, elektronisch archiviert und in Papierform an die beauftragenden Fachreferate verschickt. Für das elektronische Archiv wurden die einzelnen Dokumente entsprechend der folgenden Merkmale in der dreistufigen Verzeichnisstruktur abgelegt: beauftragendes Referat, Auftraggeber und Dokumenttyp. Darüber hinaus wurde der Erstellzeitpunkt des Dokuments im Dokumentnamen codiert. Neben dem elektronischen Archiv wurden die Dokumente physisch in der Registratur des Ministeriums abgelegt. Im Gegensatz zum elektronischen Archiv legt die Registratur eine Vielzahl von inhaltlich zusammengehörenden Einzeldokumenten unterschiedlichen Typus als Vorgänge ab. Einzelne Vorgänge werden gemäß ihres inhaltlichen Bezuges unter Rückgriff auf ein ministeriumsweites Schema (Aktenplan) von Mitarbeitern der Registratur klassifiziert und abgelegt. Die Mitarbeiter der Fachreferate waren mit dem vorgangsbezogenen Ablageschema vertraut, ohne dieses in der Regel selbst zur Klassifikation nutzen zu können. Dieser Arbeitsschritt war der Registratur vorbehalten.

Durch die Nutzung der Groupware zur gemeinsamen Dokumenterstellung und –bearbeitung stellte sich die Frage nach einem gemeinsamen Ablageschema. Diese Frage wurde in mehreren Workshops, an denen Mitarbeiter von Fachreferat und Schreibdienst teilnahmen, diskutiert. Dabei wurden verschiedene, sich im zeitlichen Verlauf wandelnde Ablagekonventionen vereinbart. Im ersten Workshop, der begleitend zur Groupwareeinführung stattfand, verständigten sich die Nutzer darauf, das Ablageschema des Schreibdienstes leicht abgewandelt zu übernehmen, d. h. Dokumente nach den beauftragenden Fachreferenten zu klassifizieren und den Dokumenttyp und das Erstelldatum im Dokumentnamen eindeutig zu verschlüsseln. Während eines Workshops ein halbes Jahr später beklagten die Mitarbeiter des Fachreferats Probleme mit dem gefundenen Klassifikationsschema. Für sie sind nicht die Dokumenttypen, sondern die inhaltlich definierten Vorgänge das entscheidende Kriterium bei ortsbezogener Suche. Als Ergebnis des Workshops wurde jedem Mitarbeiter freigestellt, in seinem Teil des gemeinsamen Arbeitsbereichs eine vorgangsbezogene Verzeichnisunterstruktur anzulegen. Diese Konvention führte zu interindividuell unterschiedlichen Ablageschemata in den Verzeichnissen der Referenten. Ein Jahr nachdem die neue Konvention gefunden wurde, beklagte der Referatsleiter, der auf die Arbeitsbereiche aller seiner Referenten zugreifen muß, die mangelnde Standardisierung der vorgangsbezogenen Unterverzeichnisse seiner Mitarbeiter. Deshalb einigte man sich letztendlich, auf dieser Ebene des Klassifikationsschemas den ministeriumsweiten Aktenplan anzuwenden. Die Umsetzung dieser Ablagekonvention führte

in der Folge aber zu erheblichen Problemen u. a. auf Grund der mangelnden Klassifikationskompetenz der Referenten.

Die Etablierung der Konventionen erforderte aus drei Gründen einen längerfristigen Verhandlungsprozeß:

- vor Einführung der Groupware gab es in den beteiligten Organisationseinheiten unterschiedliche Ablagekonventionen,
- die unterschiedlichen Nutzergruppen wurden sich ihrer speziellen Interessen und den daraus resultierenden Konflikten erst im Laufe der Nutzung der Groupware bewußt,
- divergierendes individuelles Verhalten bei der Ablage elektronischer Dokumente war feststellbar.

Ablagekonventionen sind nur ein Beispiel für Nutzungskonventionen bei Groupware (vgl. Mark u. a. 1997, Mark 1997). Unsere Erfahrungen lassen erwarten, daß sich auch andere Nutzungskonventionen im zeitlichen Verlauf ändern werden. Die Umsetzung von Nutzungskonventionen erfordert häufig den Rückgriff auf Groupwarefunktionalität. In unserem Fall ließ es beispielsweise die Anpaßbarkeit der Verzeichnisstruktur zu, neue Ablageschemata schnell zu realisieren. Dagegen mußte auf die Anwendungsprogrammierschnittstelle zurückgegriffen werden, um Möglichkeiten einer interindividuell unterschiedlichen Darstellung der Dokumente in einem Verzeichnis zu implementieren. Später wurde das ministeriumsweite Klassifikationsschema so in die Groupware integriert, daß eine einmal getroffene Zuordnung automatisch als Dokumentattribut abgespeichert werden kann. Diese Beispiele zeigen, daß das Ausmaß der Anpaßbarkeit das Ergebnis der kollektiven Verhandlungsprozesse über Nutzungskonventionen mitbestimmt und diese Prozesse deshalb durch ein hohes Maß an Anpaßbarkeit erleichtert werden.

### **3. Prozeduraler Rahmen: Integrierte Organisations- und Technikentwicklung**

#### *Beitrag:*

Towards an Integrated Organization and Technology Development; in: Proceedings of the Symposium on Designing Interactive Systems, 23. - 25.8.1995, Ann Arbor (Michigan), ACM-Press, New York 1995, S. 55 – 64

Die anpaßbare Gestaltung von Groupware muß im Software-Entwicklungsprozeß berücksichtigt werden. Betrachtet man die im vorigen Kapitel dargestellten Ergebnisse, dann stellt sich darüber hinaus die Frage, wie im Prozeß der Einführung von Groupware Möglichkeiten zur Prozeßinnovation aufgegriffen werden können. In der Fallstudie, über die im vorigen Teilkapitel berichtet wurde, lag das Interesse der Anwendungspartner nicht vornehmlich auf der Verbesserung der Arbeitsprozesse, sondern eher darin, erste Erfahrungen mit Groupware zu sammeln, um räumlich verteilte Regierungsarbeit zu ermöglichen. Die beobachteten Organisationsveränderungen ergaben sich eher als Nebeneffekt eines partizipativen und evolutionären Software-Entwicklungsprozesses. Darüber hinausreichende organisatorische Fragestellungen, wie im Falle des Bundesministeriums beispielsweise die Zukunft des zentralen Schreibdienstes und der Registraturen, des Zuschnitts der Fachreferate und des hierarchischen Aufbaus der Gesamtorganisation konnten im Rahmen des Projektes nicht

aufgegriffen werden, weil ein entsprechender Auftrag zur Organisationsentwicklung nicht Bestandteil der Projektdefinition war. Das führte dazu, daß bestehende Möglichkeiten zur Effektivierung technisch unterstützter Arbeitsprozesse ungenutzt blieben. Insofern weisen diese Erfahrungen auf die Notwendigkeit zur Integration von Technik- und Organisationsentwicklungsprozessen hin. In der software-ergonomischen Diskussion wurde schon immer die Bedeutung der gesamten Arbeitsaufgabe zur Beurteilung der Güte computergestützter Arbeit am Einzelarbeitsplatz betont (z. B.: Hacker 1987; Rödiger 1993). Nichtsdestotrotz stellen sich bei der Einführung von Groupware in Organisationen aufgrund der sich ergebenden engen Wechselwirkungen zwischen technischer Gestaltung und den Möglichkeiten der Innovation der technisch unterstützten Geschäftsprozesse neue Herausforderungen an die Forschung im Bereich computerunterstützter kooperativer Arbeit. Insbesondere ist die Entwicklung von Prozeßmodellen erforderlich, an denen sich die unterschiedlichen Akteure in einem Einführungsprozeß von Groupware orientieren können.

In dem Aufsatz "*Towards an Integrated Organization and Technology Development*" wird ein solches Prozeßmodell entwickelt. Der Aufsatz entstand bereits 1995 und war motiviert durch die Diskussion um post-tayloristische Managementkonzepte und deren Bedeutung für die Forschung in der Informatik. Das Prozeßmodell einer integrierten Organisations- und Technikentwicklung erweitert zunächst das STEPS-Prozeßmodell der Software-Entwicklung um die Berücksichtigung anpaßbarer Systemgestaltung sowohl in der Herstellungs- als auch in der Einsatzphase.

Im Gegensatz zu klassischen Modellen des Software-Engineerings betont das STEPS-Modell insbesondere den evolutionären und partizipativen Charakter von Software-Entwicklungsprozessen. Es ist damit in besonderer Weise für die Software-Entwicklung in betrieblichen Organisationen geeignet (vgl. Floyd 1994, S. 38). Software-Entwicklung wird als ein zyklischer Prozeß verstanden, bei dem die Herstellung verschiedener aufeinanderfolgender Systemversionen durch die beim Einsatz dieser Versionen festgestellten neuen Anforderungen ausgelöst wird. Ein solches Vorgehen erfordert die aktive Beteiligung der Nutzer bei der Gestaltung und der Bewertung einzelner Softwareversionen. Durch die Tatsache, daß Software als ein im zeitlichen Verlauf sich wandelnder Artefakt gesehen wird, erlaubt das STEPS-Prozeßmodell in besonderer Weise, auf die Differenziertheit und Dynamik einzelner Anwendungsfelder einzugehen.

Allerdings führen neue, während des Einsatzes einer Systemversion sich ergebende Anforderungen im STEPS-Modell immer zur Entwicklung einer neuen Systemversion. Dies erfordert Aktivitäten von Systementwicklern und bringt erheblichen Reimplementierungsaufwand mit sich. Deshalb habe ich vorgeschlagen, die Gestaltung anpaßbarer Systeme im STEPS-Prozeßmodell explizit vorzusehen (vgl. Abb. 2). Anpassung wird neben Pflege und Nutzung als eine zusätzliche Aktivität während der Einsatzphase einer Software verstanden. Diese Aktivität kann von Nutzern oder lokalen Experten durchgeführt werden. Möglichkeiten zur Anpassung einer Anwendung müssen während der Herstellungsphase konzipiert und implementiert werden. Im Herstellungsprozeß ist die Differenziertheit und Dynamik der Anwendungsfelder im Rahmen der Anforderungsspezifikation zu erfassen. Diese verschiedenen Anforderungen müssen in geeigneter Weise implementiert werden. Dabei muß versucht werden, die nicht antizipierbare Dynamik der in einzelnen Anwendungsfeldern sich entwickelnden Anforderungen durch die Berücksichtigung der Differenziertheit der Anforderungen verschiedener Nutzergruppen und Organisationen zu kompensieren (vgl. Stiemerling, Kahler und Wulf 1997).

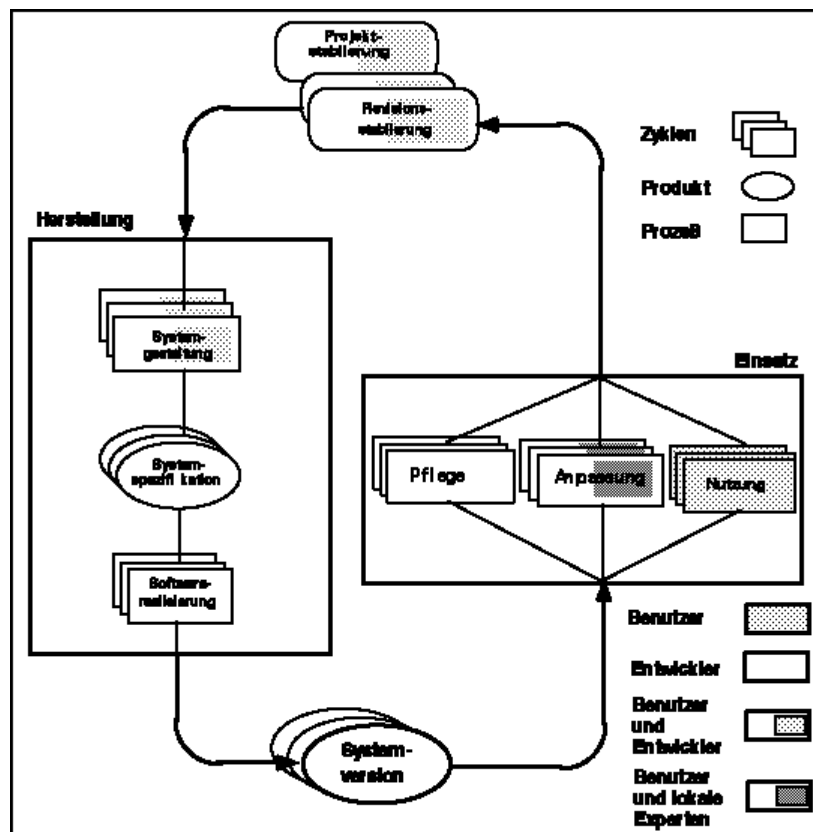


Abb. 2: Erweitertes STEPS Modell zur Entwicklung anpaßbarer Anwendungen

Um die sich beim Einsatz technisch flexibler Groupware ergebenden Möglichkeiten zur Prozeßinnovation zu nutzen und dabei zur Schaffung menschengerechter Arbeit beizutragen, habe ich den Ansatz integrierter Organisations- und Technikentwicklung ausgearbeitet. Dabei wird vorgeschlagen, Instrumente der Organisations- und Personalentwicklung sowie arbeitspsychologische Analyseverfahren mit dem erweiterten STEPS-Modell zu verbinden, um mit Problemen der bestehenden Arbeitspraxis umzugehen.<sup>4</sup> Durch eine solche Verbindung ergänzen sich die einzelnen Methoden. Klassische Ansätze der Organisationsentwicklung berücksichtigen nicht die durch Technikeinführung bestehenden Möglichkeiten zu organisatorischem Wandel, die dabei häufig notwendig werdende Software-Entwicklung und bieten keine angemessenen Kriterien, um ihren Erfolg zu evaluieren. Die vorgeschlagenen arbeitspsychologischen Instrumente bieten zwar theoriegeleitete Kriterien zur Evaluation menschengerechter Arbeit, beinhalten aber kein Methodenrepertoire zur Intervention. Ansätze des Software-Engineering gehen auf die mit der Entwicklung und Einführung von Anwendungen einhergehenden Organisations- und Qualifikationsveränderungen kaum ein. Deshalb erscheint ein integriertes, evolutionäres und partizipatives Vorgehen angemessen.

Angelehnt an in der Organisationsentwicklung übliche Vorgehensmodelle werden die fünf u. U. ineinander verschachtelten Phasen: Projektetablierung, Analyse des IST-

<sup>4</sup> Dieser Ansatz ersetzt damit die während der Herstellungsphase im ursprünglichen STEPS-Modell vorgesehene Vorbereitung der Einführung.

Zustands, Entwicklung alternativer Handlungsoptionen, Planung von Interventionen und Durchführung der Interventionen durchlaufen. Je nach dem Ausgangsproblem sind in jeder der vier letztgenannten Phasen technologische, organisationsbezogene und qualifikatorische Aspekte zu berücksichtigen. Die beteiligten Beschäftigten sind während dieser integrierten Organisations- und Technikentwicklungsprozesse zur Beteiligung zu qualifizieren. Ein solcher integrierter Organisations- und Technikentwicklungsprozeß ist zyklisch zu verstehen, insofern als die Mitarbeiter der durch die Interventionen veränderten Organisation fortlaufend neue Probleme ihrer Arbeitspraxis konstatieren können (vgl. Abb. 3).

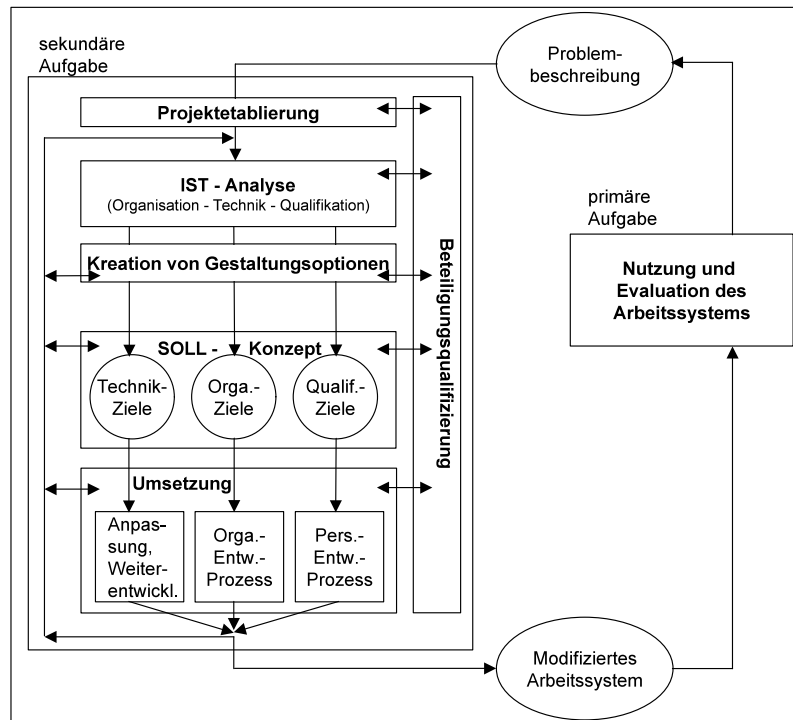


Abb. 3: Modell für den Prozeß integrierter Organisations- und Technikentwicklung

Durch die Erweiterung des STEPS-Modells zur Entwicklung anpaßbarer Software und dessen Verknüpfung mit dem Prozeßmodell für integrierte Organisations- und Technikentwicklung wurde ein Prozeßrahmen für die Entwicklung und Einführung von Groupware in Organisationen gegeben. Die Relevanz dieses Prozeßmodells wird durch die im vorigen Kapitel konstatierten Befunde zur Wechselwirkung zwischen der Einführung von Groupware und den damit einhergehenden organisatorischen Veränderungen untermauert. Das erweiterte STEPS-Modell hat die Entwicklung der im nächsten Kapitel dargestellten anpaßbaren Groupwareanwendungen geleitet.

## **4. Entwicklung anpaßbarer Groupware: Suchtool, Zugriffsrechte und Ereignisdienst**

Bei der Entwicklung neuer anpaßbarer Funktionalität kann auf bereits dokumentierte Anforderungen und innovative Realisierungen zurückgegriffen werden. Deshalb will ich im folgenden über den Entwicklungsprozeß von drei anpaßbaren Groupwarefunktionalitäten berichten. Die drei Funktionalitäten Suchtool, Zugriffsrechte und Ereignisdienste sind der technischen Organisationsschnittstelle zuzuordnen.

Sie wurden als Reaktion auf im POLITeam-Projekt festgestellte Nutzungsprobleme entwickelt. Viele dieser Probleme hingen mit dem bei der Gestaltung von Groupware bestehenden Trade-off zwischen der Förderung der Kooperation und der Wahrung der Privatheit einzelner Nutzer zusammen (Belotti und Sellen 1993, Wulf und Hartmann 1994). Vor dem Hintergrund dieses Trade-offs zeigte sich, daß die implementierten Funktionen entweder kein hinreichend differenziertes Systemverhalten boten oder der Geltungsbereich ihrer Anpassungsobjekte nicht genau genug zu spezifizieren war.

Im genannten Projekt wurde zunächst eine auf die besonderen Bedürfnisse der anwendenden Organisationen hin konfigurierte Version einer marktgängigen Groupware (LinkWorks von DEC) eingesetzt. Diese Anwendung bot den Nutzern im wesentlichen die Funktionalitäten elektronische Laufmappe, gemeinsame Arbeitsbereiche, E-mail und einen rudimentären Ereignisdienst. Die Benutzungsschnittstelle war an der Desk-Top-Metapher orientiert und bot jedem Nutzer einen eigenen Schreibtisch, auf dem eigene und gemeinsam genutzte Daten grafisch dargestellt wurden. Zur Erweiterung der gegebenen Funktionalität stellte das Produkt eine Programmierschnittstelle zur Verfügung, die es erlaubte, extern entwickelte Programme einzubinden.

### **4.1 Suchtool für Groupware**

#### *Beiträge:*

Konstruktive Weiterentwicklung eines Groupwareproduktes - Diskutiert am Beispiel der Reimplementierung eines Suchtools; in: Proceedings der achten GI-Fachtagung Software-Ergonomie'97, am: 3.3. - 6.3.1997 in Dresden, Teubner, Stuttgart 1997, S. 211 – 220

“Let’s see your Search-Tool!” – Collaborative Use of Tailored Artifacts in Groupware, in: Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP '99), ACM-Press, New York 1999 (in press)

Die ursprüngliche LinkWorks-Version enthielt ein Suchtool, das unabhängig von der aktuellen Lage eines Dokument Suchergebnisse darstellt. Insofern unterschied diese Version nicht zwischen verschiedenen Geltungsbereichen der Suchfunktion, als sie Dokumente auf dem „eigenen Schreibtisch“ genau wie solche in gemeinsamen Arbeitsbereichen und solche auf „privaten Schreibtischen“ anderer Nutzer behandelte. Diskussionen mit Nutzern vor der Einführung von LinkWorks zeigten, daß der Einsatz eines solchen Suchtools auf erheblichen Widerstand stieß, weil damit der Zugriff auf und die Manipulation von Dokumenten in privaten Arbeitsbereichen möglich wurde. Dieses Problem vergrößerte sich durch eine unzulängliche Gestaltung der Zugriffsrechtsvergabe, die eine gezielte Einschränkung der Suchrechte anderer Nutzer ausschloß (vgl. Kap. 4.2).

Auf der Basis der Auswertung von 10 teilstrukturierten Interviews, die in vier verschiedenen Organisationen geführt wurden, und den Ergebnissen von zwei Workshops mit Nutzern im POLITeam-Projekt wurde eine erste Version des Suchtools in

Visual Basic erstellt. Die Version wurde von den Nutzern und Nutzerbetreuern in zwei Workshops evaluiert. Bei dieser Version stand die Entwicklung neuer Gestaltungskonzepte im Vordergrund. Basierend auf den in der Voruntersuchung und bei der Evaluation des ersten Prototypen gesammelten differenzierten Anforderungen wurde eine zweite, komponentenbasierte Version des Suchtools in Java entwickelt, als Ergebnis eines Workshops verbessert und in einem Anwendungsfeld eingesetzt. Im Gegensatz zur traditionellen Nutzung von Komponenten im Software-Engineering (z. B. Banavar u. a. 1998) kann das komponentenbasierte Suchtool während der Laufzeit durch Rekombination der Komponenten angepaßt werden (vgl. Stiernerling und Cremers 1998).

Als Ergebnis der Vorstudie und der Evaluation der ersten Systemversion ergaben sich die folgenden, im zeitlichen Verlauf und zwischen Nutzer(-gruppen) sich verändernden Anforderungen:

- verschiedene Kriterien, um die Suchanfrage zu spezifizieren (z.B. Dokumentenname, Dokumentenart, Erstellungsdatum, Änderungsdatum),
- verschiedene Fenster zur differenzierten Ergebnisdarstellung, um gefundene Dokumente getrennt nach bestimmten automatisch unterscheidbaren Kriterien anzuzeigen (z. B. nach der aktuellen Lage der Dokumente),
- verschiedene Arten von Anzeigefenstern (normale Anzeige jedes einzelnen Dokuments, Anzeige der Anzahl der gefundenen Dokumente und keine Anzeige),
- verschiedene Funktionen, um mit den gefundenen Dokumenten umzugehen (z. B. direkt öffnen, Verweis erzeugen, Kopie erzeugen).

Um diese Anforderungen erfüllen zu können, wurde das Suchtool in sechs Arten von elementaren Komponenten zerlegt. Anfragekomponenten erlauben es, eine auf die spezifischen Benutzerbedürfnisse zugeschnittene Eingabemaske zu erstellen, der Suchknopf ist eine Komponente, deren Aktivierung die Spezifikation der Suchabfrage abschließt. Die Suchmaschine übermittelt die Suchanfrage an die LinkWorks-Datenbank und erhält die Liste der gefundenen Dokumente zurück. Verschiedene Weichen erlauben es, die gefundenen Dokumente nach bestimmten Kriterien getrennten Fenstern zuzuweisen. Zwei verschiedene Darstellungsfenster erlauben es, die gefundenen Dokumente entweder einzeln darzustellen, oder lediglich die Anzahl der gefundenen Dokumente anzuzeigen. Verschiedene Kontrollknöpfe erlauben es, im normalen Anzeigefenster ausgewählte Dokumente entweder direkt zu öffnen oder einen Verweis bzw. eine Kopie des Dokuments auf dem Schreibtisch des Suchenden zu erzeugen.

Die verschiedenen Komponenten werden im Anpassungsmodus des Suchtools in einer graphischen Programmierumgebung durch Mausklickoperationen miteinander verbunden. Dazu werden Eingangs- und Ausgangsports der Komponenten visualisiert (leere Kreise für Eingangsports, volle Kreise für Ausgangsports). Um Nutzer bei der Konstruktion von Komponenten zu unterstützen, sind Eingabe- und Ausgabeports, die miteinander verknüpft werden können, in derselben Farbe dargestellt.

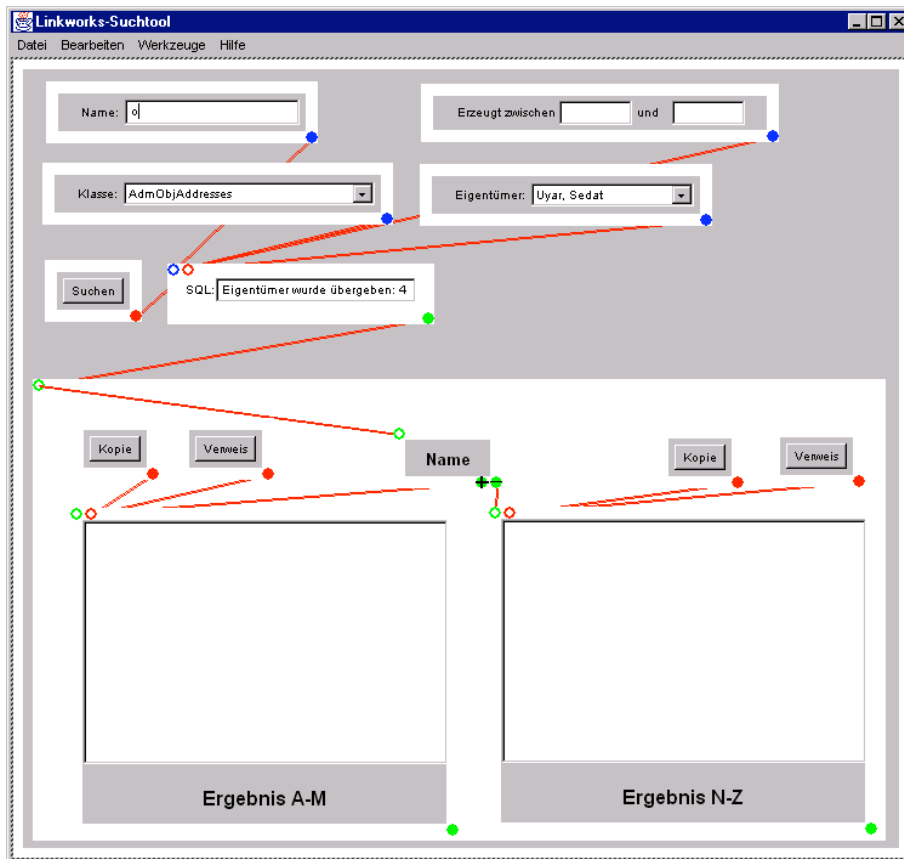


Abb. 3: Suchtool im Anpassungsmodus

Die Anpassungsfunktionalität erlaubt es, komplexe Komponenten als Aggregation verschiedener miteinander verbundener elementarer Komponenten zu definieren. In Abb. 3 ist ein Suchtool dargestellt, das aus zwei komplexen Komponenten besteht, einer komplexen Komponente zur Suche (bestehend aus den Komponenten, die die Suchabfrage spezifizieren, dem Suchknopf und der Suchmaschine) und einer komplexen Komponente zur Anzeige (bestehend aus einer Weiche, den Anzeigefenstern und Kontrollknöpfen). Um ein solches Suchtool zu erzeugen, müssen Benutzer lediglich zwischen vorgefertigten komplexen Komponenten für die Suche und Anzeige aussuchen und den Ausgangsport der Suchkomponente mit dem Eingangsport der Anzeigekomponente verbinden.

Um die Anpassung des Suchtools weiter zu vereinfachen, können Nutzer zwischen Alternativen bereits vordefinierter Suchtools auswählen. Ein Suchtool ist eine komplexe Komponente, die entweder aus anderen komplexen Komponenten oder aus elementaren Komponenten besteht. Insofern beinhaltet die Anpassungsumgebung mindestens drei Stufen unterschiedlicher Anpassungskomplexität.

## 4.2 Zugriffsrechte für Groupware

### Beiträge:

Tailoring Groupware for Different Scopes of Validity, in: Behaviour & Information Technology, Vol. 18, No. 3, 1999, S. 199 - 212

Beyond "Yes and No" - Extending Access Control in Groupware with Awareness and Negotiation, in: Group Decision and Negotiation - Special Issue zur COOP' 98 Tagung 1999 (in press)

Die ursprünglich in LinkWorks implementierte Zugriffskontrolle erlaubte Nutzern lediglich die Auswahl zwischen vordefinierten Zugriffsprofilen. Ausgewählt für ein Objekt gaben die Profile an, welcher Nutzer bzw. welche Nutzergruppe bestimmte Funktionen ausführen konnte. Die Zugriffsprofile mußten von Systemadministratoren vordefiniert werden und erlaubten es lediglich, Nutzer entsprechend der formalen zuvor in der Groupware zu modellierenden Hierarchie der Aufbauorganisationen zu gruppieren. Daneben war im System ein Automatismus der Art implementiert, daß Vorgesetzte gemäß der formalen Organisationshierarchie lesenden Zugriff auf die Dokumente ihrer Untergebenen hatten.

Insofern konnte der Geltungsbereich der die Zugriffskontrollfunktion spezifizierenden Anpassungsobjekte lediglich über die drei Dimensionen Nutzer, Dokumente und zu aktivierende Funktion bestimmt werden. Die Aggregationsmöglichkeiten auf der Nutzerdimension waren beschränkt, weil quer zur Aufbauorganisation liegende Gruppen (z. B. ad-hoc-Teams) nicht gebildet werden konnten. Außerdem erfolgte auf Basis der Organisationshierarchie eine automatische Vergabe von Leserechten, die manuell getroffene Spezifikationen durchbrach und von den Nutzern als ihre Privatheit verletzend empfunden wurde. Hinsichtlich des Systemverhaltens konnten die Nutzer lediglich zwischen Vergabe positiver und negativer Zugriffsrechte auswählen.

Auf der Basis der festgestellten Probleme und nach einer entsprechenden Literaturanalyse entstand die Idee, das Konfigurationsproblem mittels an Produktionsregeln in Expertensystemen angelehnte Regeln anzugehen. Nachdem verschiedene papierene Mock-ups mit anderen POLITeam-Entwicklern diskutiert worden waren, wurde ein erster Prototyp entwickelt. Ausgehend von den beim Design aufgeworfenen Fragen wurden neun teilstrukturierte Interviews mit Mitarbeitern aus drei Organisationen geführt. Dabei wurden insbesondere in den Organisationen angewandte Zugriffsstrategien auf physische und elektronische Dokumente erfragt. Außerdem wurden die Ergebnisse mehrerer, im POLITeam-Projekt durchgeführter Workshops im Hinblick auf in den Anwendungsfeldern feststellbare Zugriffsstrategien hin untersucht (vgl. Stiemerling, Kahler und Wulf 1997). Besonders charakteristische Zugriffsstrategien wurden in Form von Zugriffsszenarien dokumentiert. Die Ergebnisse der empirischen Untersuchung wurden in einem Prototypen implementiert und per Anwendungsprogrammierschnittstelle in LinkWorks integriert. Das regelorientierte Konfigurationswerkzeug wurde zunächst mittels Thinking-Aloud-Methode evaluiert und dann der gesamte Prototyp in einem Workshop mit Nutzern diskutiert.

Während dieses Entwicklungsprozesses zeigte sich, daß zur Spezifikation des Geltungsbereichs von Zugriffsrechten in desktoporientierten Groupwareanwendungen die folgenden Dimensionen von Relevanz sind: Nutzer, Nutzergruppen, Nutzerrollen, zu aktivierende Funktionen, Objekte, Objektklassen, Objektlagen, Objektattribute (z.B. gezeichnet von), Beziehungen zwischen Objekten und Nutzern (z.B. augenblickliche

oder zukünftige Bearbeiter eines Vorgangs), relative Zeitintervalle, absolute Zeitintervalle. Hinsichtlich des Systemverhaltens ergab sich, daß neben der Erlaubnis und dem Verbot des Zugriffs die folgenden beiden Vergabemöglichkeiten relevant sind:

- Zugriff zu gewähren, unter der Maßgabe, über einen erfolgten Zugriff automatisch informiert zu werden,
- Zugriff zu verweigern, unter der Maßgabe, im Moment des Zugriffswunsches jeweils darüber befinden zu können.

Hinsichtlich dieser Verfeinerung des Systemverhaltens auf der Erlaubnis/Verbots-Dimension erwies es sich als notwendig, die Zugriffskontrolle so anpassen zu können, daß nicht nur der Eigner eines Dokuments, sondern auch andere Nutzer – sogenannte „vertrauenswürdige Dritte“ – die automatisch generierten Benachrichtigungen und Abfragen empfangen können.

Benutzer definieren Zugriffsrechte durch die Erzeugung neuer Regeln und die Modifikation oder das Löschen existenter Regeln. Die verwandten Regeln enthalten einen Bedingungsteil zur Spezifikation des Geltungsbereiches und einen Aktionsteil zur Beschreibung des vergebenen Zugriffsrechts. Abb. 5 zeigt ein Fenster zur Eingabe neuer Regeln. Ein Nutzer wählt die relevanten Dimensionen des Geltungsbereiches einer Regel mittels der obigen Selektorboxen aus (Bedingungsteil der Regel). In der darunterliegenden Auswahlbox kann dann das Zugriffsrecht als Auswahl aus den vier vorgegebenen Zugriffsalternativen ausgewählt werden (Aktionsteil der Regel). Der natürlichsprachliche Text im unteren Teil des Fensters stellt die bis dato spezifizierte Regel dar und verändert sich automatisch, wenn der Benutzer die Definition der Regel verändert.



Abb. 5: Fenster zum Spezifizieren von Zugriffsregeln

Ein zentrales Problem bei der regelbasierten Konfiguration ist der Umgang mit inkonsistenten Regeln, d. h. solchen Regeln, die für eine bestimmte Zugriffssituation gegensätzliche Zugriffsrechte spezifizieren. Zum Umgang mit diesen Inkonsistenzen ist im Werkzeug ein Algorithmus zur eindeutigen Gewichtung der Regeln implementiert. Mittels dieses Algorithmus ergibt sich eine eindeutige Reihenfolge der einen, den bestimmten Zugriffsversuch determinierenden Regel. Dabei lassen sich zwei Arten von Inkonsistenzen unterscheiden:

- Inklusion, wenn der Geltungsbereich einer Zugriffsregel vollständig in dem einer anderen Regel enthalten ist,
- Orthogonalität, wenn keine Inklusionsbeziehung zwischen den Geltungsbereichen existiert.

Inklusionsbeziehungen zwischen zwei Regeln werden von dem Gewichtungsalgorithmus so aufgelöst, daß die spezifischere Regel höhere Priorität erhält. Orthogonalitätsbeziehungen werden aufgelöst, indem die einzelnen Dimensionen des Geltungsbereichs unterschiedlich hoch gewichtet werden. Die Regel, deren Bedingungsteil aus höher gewichteten Regeln besteht, erhalten höhere Priorität. Um diesen Umgang mit Inkonsistenzen auf die Anforderungen unterschiedlicher Anwendungsfelder anpassen zu können, kann die Gewichtung der einzelnen Dimensionen verändert werden. Außerdem kann die Gewichtung einzelner Regeln manuell festgelegt werden.

### **4.3. Filter eines Ereignisdienstes für Groupware**

#### *Beiträge:*

Exploration Environments: Supporting Users to Learn Groupware Functions, submitted to: Interacting with Computers

Design for Self-directed Learning – Evaluating the Concepts of Direct Activation and Explorative Execution, Arbeitsberichte, ProSEC - Institut für Informatik III, Universität Bonn, Oktober 1999

Untersuchungen im Kontext des POLITeam-Projekts zeigten die Notwendigkeit, Nutzer beim kooperativen Anpassen von Textverarbeitungssystemen zu unterstützen. Deshalb wurde ein marktgängiges Textverarbeitungsprogramm um eine Funktionalität erweitert, die Nutzer beim Austausch von Anpassungsobjekten unterstützt. Jeder Nutzer verfügt über private Ablagebereiche und einen Posteingang zum Empfang von Anpassungsdaten, die von anderen Nutzern gesendet wurden. Darüber hinaus ist ein gemeinsamer Ablagebereich implementiert, in den Nutzer Anpassungsdaten aus ihrem privaten Ablagebereich übertragen können und aus dem sie Anpassungsdaten in ihre lokale Ablage kopieren können (vgl. Kahler u. a. 1999).

Damit Nutzer sich über das gemeinsame Anpassungsgeschehen informieren können, ist ein zusätzlicher Ereignisdienst implementiert. Dabei werden neun verschiedene Ereignistypen automatisch aufgezeichnet, über das Netz verteilt und am Interface anderer Nutzer angezeigt. Das Design basiert im wesentlichen auf den Erfahrungen, die im POLITeam-Projekt für den Ereignisdienst in gemeinsamen Arbeitsbereichen entwickelt wurden. Auf der Basis einer Vielzahl von semi-strukturierten Interviews und der Dokumentation verschiedener Workshops, auf denen Nutzungsprobleme diskutiert wurden, ergab sich die Anforderung den Ereignisdienst anpaßbar durch Einfügen dreier Filter zu gestalten (vgl. Fuchs u. a. 1996).

Diese Ergebnisse zeigten, daß die automatisch erfolgende Ereigniserzeugung, -verteilung und -anzeige besser als getriggerte anpaßbare Funktionen zu implementieren sind, um zusätzliche Variabilität des Systemverhaltens (Unterdrückung der Erzeugung, Verteilung und Anzeige) zu erzielen. Da die Unterdrückung der Erzeugung, Verteilung und Anzeige nicht für alle Fälle vorab festgelegt werden kann, spielt die Definition des Geltungsbereichs einer Anpassung eine entscheidende Rolle.

Die konkrete Ausgestaltung des Geltungsbereichs hängt von den Spezifika des jeweiligen Ereignisdienstes ab. Im Falle des Empfangsfilters des Ereignisdienstes wurden der Ereignistyp und die Darstellungssituation als relevante Dimensionen des Geltungsbereichs erachtet. Hinsichtlich des Systemverhaltens werden drei verschiedene Darstellungsintensitäten unterschieden. Im Falle des Herstellerfilters wurden der Ereignistyp und der Empfänger, bzw. die Empfängergruppe des Ereignisses als für die Spezifikation des Geltungsbereichs relevante Dimensionen erachtet. Der Herstellerfilter bewirkt, daß nur solche Ereignisse beim Empfänger angezeigt werden können, deren Erzeugung zugelassen wurde.

Um die Komplexität der Anpassung zu reduzieren, wurden für die Voreinstellung beider Filter gestufte Anpassungsfunktionen realisiert. Beim Empfangsfilter konnten Benutzer pro Ereignistyp ein Interessenprofil aus einer Liste definierter, natürlichsprachlich beschriebener Profile auswählen. Das Profil legte fest, in welcher Nutzungssituation das Ereignis mit welcher Intensität dargestellt werden soll. Auf der zweiten Ebene der Anpassungsfunktionen kann der Nutzer Interessenprofile durch Kombination von Darstellungssituation und Intensität erzeugen und benennen. Beim Herstellerfilter kann der Nutzer auf der obersten Stufe festlegen, welches Ereignis an welchen Empfänger bzw. welche Empfängergruppe verteilt werden soll. Auf der zweiten Stufe der Anpassungsfunktionalität kann der Benutzer neue Empfänger definieren.

#### **4.4 Zusammenfassung**

Ich habe den Entwicklungsprozeß und die Ergebnisse von drei der technischen Organisationsschnittstellen zurechnenden Funktionen dargestellt. Obwohl der konkrete Softwareentwicklungsprozeß für jede einzelne der hier diskutierten Groupwarefunktionalitäten unterschiedlich verlief, weisen diese Fallstudien jedoch auf die Bedeutung der beiden Grundprinzipien des erweiterten STEPS - Modells, der evolutionären Systementwicklung und der Partizipation der Nutzer für die Gestaltung anpaßbarer Groupware hin.

Nutzungsprobleme stießen die Entwicklung anpaßbarer Groupwarefunktionen an. Diese Probleme waren auf den im ursprünglichen Design unzureichend gelösten Trade-off zwischen Förderung der Kooperation und der Wahrung der Privatheit der Nutzer zurückzuführen. Im Fall der Vergabe von Zugriffsrechten wurde das Systemverhalten über die Erlaubnis und das Verbot des Zugriffs hinaus differenziert. Außerdem wurde der Geltungsbereich der Zugriffsrechte genauer spezifiziert und für die Nutzer flexibler einstellbar gestaltet. Beim Suchtool wurden eine zuvor starre Funktionalität durch das Einfügen verschiedener Anfragekomponenten und einstellbarer Fenster zur Anzeige von Suchergebnissen anpaßbar gestaltet. Mittels der neu entwickelten Weichen wurde die Spezifikation von Geltungsbereichen für einzelne Anzeigefenster in einem Suchtool ermöglicht. Beim Ereignisdienst wurde eine neu zu entwickelnde Funktionalität durch das Einfügen des Hersteller- und Empfangsfilters anpaßbar gestaltet. Das Interessenprofil spezifiziert den Geltungsbereich einer Einstellung des Empfangsfilters; der Empfänger bzw. die Empfängergruppe differenziert den Geltungsbereich der Einstellung des Herstellerfilters.

Differenzierte Geltungsbereiche erhöhen die Anpassungskomplexität aus der Sicht der Nutzer. Im Fall der Spezifikation der Zugriffsrechte wurde ein regelbasierter Lösungsansatz entwickelt, um mit diesem Problem umzugehen. Dieser Ansatz erscheint besonders angemessen für komplexstrukturierte Geltungsbereiche. Im Fall des Such-

tools und der Filter des Ereignisdienstes wurden gestufte Anpassungsfunktionen zur Spezifikation differenzierten Systemverhaltens implementiert.

Rückblickend scheinen mir insbesondere die folgenden Gestaltungskonzepte relevant zu sein:

- Verfeinerung von Zugriffsrechten in Überwindung der Erlaubnis-/ Verbots- Dichotomie,
- Einbeziehung vertrauenswürdiger Dritter zur Flexibilisierung der Zugriffsrechtsvergabe,
- flexibel einstellbare Anzeigefenster zur differenzierten Darstellung von Suchergebnissen,
- Herstellerfilter zur Wahrung der Privatheit von Nutzern eines Ereignisdienstes,
- regelbasiertes Werkzeug zur Spezifikation des Geltungsbereiches von Anpassungen,
- gestufte Anpassungsfunktionen zur Erleichterung der Durchführung komplexer Anpassungen.

Die ersten vier Konzepte sind immer dann gestaltungsrelevant, wenn die zu entwickelnde Groupware über Funktionalitäten wie die hier diskutierten verfügt. Das vorletzte Konzept ist relevant für die Gestaltung anpaßbarer Anwendungen mit einem ausdifferenzierten Geltungsbereich der Anpassungsobjekte. Das letzte Konzept ist besonders wichtig bei der Gestaltung von Anwendungen die über weitreichende Anpassungsmöglichkeiten verfügen.

## **5 Dialogschnittstelle anpaßbarer Groupware: Die Konzepte direkte Aktivierbarkeit und explorative Ausführbarkeit**

Wie bei anderen Funktionen auch hängt die Nutzbarkeit von Anpassungsfunktionen wesentlich vom Design der Ein-, Ausgabe- und der Dialogschnittstelle ab. Während ich bisher stark die die anpaßbare Gestaltung von Groupware motivierenden Aspekte der Organisationsschnittstelle und die Gestaltung der technischen Organisationschnittstelle untersucht habe, will ich im folgenden zwei von mir entwickelte Konzepte zur Gestaltung der Dialogschnittstelle anpaßbarer Groupware vorstellen. Die Gestaltung der Dialogschnittstelle hat gerade im deutschsprachigen Raum eine lange wissenschaftliche Tradition (vgl. Dzida, Itzfeld und Herda 1978), die maßgeblich zur Formulierung entsprechender internationaler Normen beigetragen hat (vgl. ISO 9241 Teil 10).<sup>5</sup>

Ich habe mich insbesondere mit der Anforderung nach einer lernförderlichen Gestaltung anpaßbarer Groupware beschäftigt. Lernen kann als längerfristige Verhaltensänderung verstanden werden, die aufgrund von Erfahrungen zustande kommt. Das Erlernen einer Computeranwendung verändert das Verhalten der Nutzer immer dann, wenn die Funktionalität der Computeranwendung die gerade durchzuführende Aufgabe unterstützt. Es lassen sich zwei Formen des Erlernens einer Computeranwendung unterscheiden: Der Lernvorgang kann zum einen extern strukturiert werden, wie dies beispielsweise bei trainergesteuerten Schulungsveranstaltungen der Fall ist.

---

<sup>5</sup> In Weiterentwicklung dieser Normen hat Herrmann (1994) einen modifizierten Katalog von softwareergonomischen Gestaltungsgrundsätzen für Groupware vorgelegt.

Zum anderen können Anwendungen durch selbstgesteuerte Aktivitäten der Lernenden erlernt werden (vgl. Dutke 1994). Eine Vielzahl von Untersuchungen zeigen die Bedeutung letzterer Lernform (Carroll and Mack 1983 und 1984, Kamouri u. a. 1986, Greif und Jankowski 1987, Frese u. a. 1988). Insofern muß die Anforderung nach Lernförderlichkeit insbesondere im Sinne einer Unterstützung selbstgesteuerten Lernens durch die Nutzer verstanden werden.

Die enge Wechselwirkung zwischen der Bildung eines geeigneten Verständnisses der Funktionalität einer Anwendung und der Fähigkeit, diese anpassen zu können, ist in der Literatur wiederholt thematisiert worden (z. B. Herrmann 1986; Friedrich 1990; Koch, Reiterer und Tjoa 1991; Maaß 1995). Empirische Untersuchungen zeigen, daß selbstgesteuertes Lernen eine besondere Rolle beim Erlernen anpaßbarer Funktionen spielt (vgl. Mackay 1990, Oppermann und Simm 1994). Dies scheint insbesondere dann der Fall zu sein, wenn das Systemverhalten durch differenzierte Geltungsbereiche einzelner Anpassungsobjekte bestimmt wird. Im folgenden werden zwei technische Konzepte entwickelt, die Nutzer beim selbstgesteuerten Lernen der Funktionalität anpaßbarer Groupware unterstützen. Beispielhafte Implementierungen der Konzepte werden aufgezeigt, und es wird über die Ergebnisse qualitativer und quantitativer Evaluationsstudien berichtet.

### **5.1 Das Konzept der direkten Aktivierbarkeit**

#### *Beiträge:*

On Search for Tailoring Functions: Empirical Findings and Implications for Design, in: Proceedings of OZCHI '99, November 28 - 30, 1999 Wagga Wagga, Australia, 1999, (in press)  
 Design for Self-directed Learning – Evaluating the Concepts of Direct Activation and Explorative Execution, Arbeitsberichte, ProSEC - Institut für Informatik III, Universität of Bonn, Oktober 1999

Das Konzept der direkten Aktivierbarkeit dient dazu, Nutzern Unterstützung beim Auffinden von Anpassungsfunktionen in der Situation zu geben, in der sie diese zur Ausführung ihrer aktuellen Aufgabe benötigen. Winograd und Flores (1986) unterscheiden zwischen zwei Arten des Umgangs mit Artefakten. Während der normalen Nutzung sind sie zuhanden (ready-to-hand) und erfordern keinen zusätzlichen Aufwand. Erst im Falle einer Störung (breakdown) rücken sie ins Bewußtsein, sind als eigenständige Objekte vorhanden (present-at-hand) und werden reflektiert. Für das Design von Artefakten wird von den Autoren zunächst deren „Zuhandensein“ gefordert. Da Störungen aber nicht auszuschließen sind, unterstützt gutes Design die dann notwendig werdenden Aktivitäten der Nutzer durch Hilfen bei der Orientierung und beim Lernen.

Ich will mich im folgenden mit solchen Störungen im Umgang mit Computeranwendungen befassen, die durch Anpassungsaktivitäten behoben werden können. Es wird der Frage nachgegangen, wie Nutzer durch die Gestaltung der Dialogschnittstelle darin unterstützt werden können, Anpassungsfunktionen im Moment einer Störung im Umgang mit einer anpaßbaren Funktion zu finden. Das Problem mangelnder Orientierung betrifft sowohl die Suche nach Nutzungsfunktion als auch die nach Anpassungsfunktionen. Es ist besonders ausgeprägt, wenn Software nicht speziell als Auftragsentwicklung für ein Anwendungsfeld, sondern als Produkt für eine Vielzahl von verschiedenen Anwendungsfeldern entwickelt wird (Grudin 1991). Ihr Funktionsumfang nimmt typischerweise von Version zu Version ständig zu.

Der Aufsatz "*On Search for Tailoring Functions: Empirical Findings and Implications for Design*" ist im Jahre 1999 entstanden und untersucht Möglichkeiten der Integration von Nutzungs- und Anpassungsfunktionen im Rahmen der Dialogschnittstelle. Diese Frage wurde bisher wissenschaftlich kaum untersucht. In der Literatur fanden sich Hinweise, daß Nutzer Probleme beim Auffinden existenter Anpassungsfunktionen haben. Basierend auf einer empirischen Vorstudie über den zeitlichen Verlauf von Anpassungsaktivitäten bei einer marktgängigen Textverarbeitung<sup>6</sup> und die dabei auftretenden Probleme beim Finden von Anpassungsfunktionen wird das Konzept der direkten Aktivierbarkeit entwickelt. Es postuliert, die funktionalen Interaktionspunkte bestimmter Anpassungsfunktionen so in die Dialogschnittstelle zu integrieren, daß im Moment einer Störung im Umgang mit einer anzupassenden Funktion die zugehörige Anpassungsfunktion unmittelbar aktiviert werden kann. Dieses Konzept ist für anpaßbare Einzelplatzanwendungen ebenso relevant wie für Groupware.

Um Probleme beim Umgang mit Anpassungsfunktionen zu untersuchen, haben wir elf teilstrukturierte Interviews mit Nutzern eines marktgängigen Textverarbeitungsprogramms geführt. Dabei wurde insbesondere nach dem zeitlichen Verlauf von Anpassungsaktivitäten und Problemen bei der Nutzung der Anpassungsfunktionen gefragt. Die empirische Voruntersuchung zeigt, daß die Aktivierung von Anpassungsfunktionen nicht nur unmittelbar nach Einführung einer Anwendung, sondern auch während deren alltäglicher Nutzung erfolgt. Dies ist insbesondere dann der Fall, wenn Nutzer ein Problem bei der aufgabenbezogenen Verwendung von Nutzungsfunktionen haben. In solchen Situationen ist es häufig der Fall, daß die Anpassungsfunktionen von Nutzern kaum gefunden werden. Dies führt dazu, daß die Anpassungsfunktionen entweder gar nicht aktiviert werden oder ihre Suche zu erheblichem Aufwand führt. Die Vorstudie deutet außerdem darauf hin, daß eine geeignete Visualisierung der funktionalen Interaktionspunkte die Exploration der betreffenden Anpassungsfunktionen fördert.

Die Implementierung des Konzepts direkter Aktivierbarkeit erleichtert das Finden einer Anpassungsfunktion in Zuständen der Dialogschnittstelle, in denen die Funktion benötigt wird, d. h. sie ist vorhanden im Moment der Wahrnehmung einer Störung.<sup>7</sup> Ich will im folgenden das Konzept „direkte Aktivierbarkeit“ für Benutzungsschnittstellen, die gemäß dem Leitbild der direkten Manipulation realisiert sind, verdeutlichen.

Anpassungsfunktionen werden benötigt, wenn Nutzer den aus der Ausführung einer Funktion resultierenden Zustandsübergang wahrnehmen, dieser aber nicht ihren Erwartungen entspricht. Handelt es sich bei den anzupassenden Funktionen um normale Funktionen, kennen Nutzer typischerweise noch deren funktionalen Interaktionspunkt. Insofern kann der funktionale Interaktionspunkt der anzupassenden Funktion als Bezugspunkt für die Aktivierung der Anpassungsfunktion dienen. Getriggerte Funktionen haben definitionsgemäß keinen funktionalen Interaktionspunkt. Deshalb können als Bezugspunkte für die Aktivierung der zugehörigen Anpassungsfunktion

---

<sup>6</sup> Die weitverbreitetsten Textverarbeitungsanwendungen werden, ähnlich wie Groupware, in der Regel als Produkt entwickelt. Aufgrund des mittlerweile stark angewachsenen Funktionsumfangs lassen sich am Beispiel von Textverarbeitungsanwendungen Orientierungsprobleme bei der Suche nach einzelnen Funktionen in besonderer Weise untersuchen.

<sup>7</sup> Das Konzept läßt sich nur auf solche Anpassungsfunktionen anwenden, die andere Funktionen verändern. Damit werden solche Anpassungsfunktionen nicht behandelt, die neue Nutzungsfunktionen kreieren (z. B. Makroaufzeichnung).

die Visualisierung der Ausführung der getriggerten Funktion oder die aus der Ausführung resultierenden Zustandsübergänge verwendet werden.

Nachdem visuelle Bezugspunkte für die anpaßbare Funktion ausgemacht sind, ist zu klären, wie die Aktivierung der Anpassungsfunktionen erfolgen kann. Dazu habe ich zwei Ansätze vorgeschlagen: Zum einen kann der funktionale Interaktionspunkt der Anpassungsfunktion mit "visuellem Bezug" zum Bezugspunkt realisiert werden. Zum anderen kann auf den visuellen Bezug verzichtet werden, wenn ein konsistenter Modus besteht, bei dem sich aus den Merkmalen des Bezugspunkts das Aktivierungsgeschehen für die Anpassungsfunktion ableiten läßt.

Die Realisierung des visuellen Bezugs zwischen dem Bezugspunkt und dem funktionalen Interaktionspunkt der Anpassungsfunktion ist eine Frage der Ein- und Ausgabeschnittstelle und ist für jede Anwendung konkret auszugestalten. Bei der Anpassung normaler Funktionen läßt sich der visuelle Bezug auf verschiedene Weise realisieren. Müssen bei vielfältigen Funktionen Parameter während der Aktivierung spezifiziert werden, so sollte der funktionale Interaktionspunkt der Anpassungsfunktion in der Nähe der zu den Parametern gehörigen Interaktionspunkte implementiert werden (z. B. in demselben Fenster). Wird die Funktion unmittelbar aus einem Menue oder mittels eines Icons aktiviert, so sollte der Interaktionspunkt der Anpassungsfunktion in räumliche Nähe zu dem der anpaßbaren Funktion realisiert sein. Bei getriggerten Funktionen sollte der funktionale Interaktionspunkt in die Darstellung der Funktionsausführung oder die Darstellung des aus der Funktionsausführung resultierenden Zustandsübergangs integriert werden.

Ein konsistenter Aktivierungsmodus der Anpassungsfunktionen ist beispielsweise bei Mørch (1997) realisiert. In der von ihm implementierten Grafikanwendung können die Nutzer alle Anpassungsfunktionen dadurch aktivieren, daß sie die für die Aktivierung der anzupassenden Funktion notwendigen Eingaben machen und gleichzeitig eine bestimmte Taste drücken.

Der Aufsatz „*Design for Self-directed Learning - Evaluating the Concepts of Direct Activation and Explorative Execution*“ ist im Herbst 1999 entstanden und beschreibt, wie das Konzept direkte Aktivierbarkeit für die beiden anpaßbaren Filter des Ereignisdienstes exemplarisch implementiert wurde. Bei den beiden Filtern handelt es sich um getriggerte Funktionen. Im Fall des Empfangsfilters wurde die Anzeige eingegangener Ereignisse als Bezugspunkt zur Aktivierung der ersten Stufe der Anpassungsfunktionen genutzt. Im Falle des Herstellerfilters fungierte die Visualisierung der Funktionsausführung als Bezugspunkt zur Aktivierung der ersten Stufe der Anpassungsfunktionen. Die gestuften Anpassungsfunktionen sind normale Funktionen. Als Bezugspunkt zur Aktivierung der zweiten Stufe der Anpassungsfunktionen fungiert dort die Liste zur Auswahl von Funktionsalternativen. Die Direktheit der Aktivierung wurde also in diesem Fall durch visuellen Bezug realisiert, weil der Ereignisdienst auf Basis einer marktgängigen Textverarbeitung implementiert wurde (vgl. Kap. 4.3) und in diesem Umfeld ein konsistenter Aktivierungsmodus für Anpassungsfunktionen nicht zu realisieren war.

Basierend auf dieser Implementierung wurde das Konzept direkte Aktivierbarkeit in einer teststatistischen Untersuchung evaluiert. Dabei bin ich von den Hypothesen ausgegangen, da die Implementierung von direkter Aktivierbarkeit dazu führt, daß Anpassungsfunktionen besser gefunden werden und deshalb die Anpassungen erfolgreicher und schneller durchgeführt werden können. Um diese Hypothesen zu testen, wurde im Rahmen eines breiter angelegten Experiments die Nutzung zweier Versio-

nen des anpaßbaren Ereignisdienstes untersucht. In der einen Version war direkte Aktivierbarkeit in der oben beschriebenen Form implementiert, in der anderen Version war dieses Konzept nicht realisiert. Die Versuchsteilnehmer mußten mit ihrer jeweiligen Version der Groupware zwei Anpassungsaufgaben durchführen. Die Aufgabendurchführung wurde protokolliert und anschließend ausgewertet.

Als Ergebnis dieser Untersuchung läßt sich feststellen, daß die in den Hypothesen formulierten Annahmen über die positiven Effekte direkter Aktivierbarkeit in signifikanter Weise nachgewiesen werden konnten. Insofern hat das Konzept direkte Aktivierbarkeit durch die Ergebnisse dieser Evaluationsstudie eine Bestätigung erfahren. Es verbleibt aber zu untersuchen, ob die im Laborsetting festgestellten positiven Effekte direkter Aktivierbarkeit sich auch in der täglichen Arbeitspraxis der Nutzer einstellen werden.

## **5.2 Das Konzept der explorativen Ausführbarkeit**

### *Beiträge:*

Exploration Environments: Supporting Users to Learn Groupware Functions, submitted to: *Interacting with Computers*

„Why did that happen?“ – Building Appropriate Mental Models on Groupware Functions, in: Bullinger, H.-J.; Ziegler, (Hrsg.) J.: *Human-Computer Interaction: Communication, Cooperation and Application Design*, Lawrence Erlbaum, Mahwah, 1999b, S. 338 - 342

Design for Self-directed Learning – Evaluating the Concepts of Direct Activation and Explorative Execution, Arbeitsberichte, ProSEC - Institut für Informatik III, Universität of Bonn, Oktober 1999

Das Auffinden der funktionalen Interaktionspunkte ist nur eine Voraussetzung für das selbstgesteuerte Lernen anpaßbarer Funktionen. Nutzer müssen darüber hinaus unterstützt werden, den aktuellen Zustand einer anpaßbaren Funktion und die Wirkung der Ausführung von Anpassungsfunktionen auszuprobieren. Bei der Exploration von Groupware entstehen aufgrund der Verteiltheit der verschiedenen Benutzungsschnittstellen und des Zusammenwirkens der Aktivitäten unterschiedlicher Nutzer neue Probleme. Die bisher in der Literatur zu findenden Konzepte zur Unterstützung der Exploration von Einzelplatzanwendungen erwiesen sich als unzureichend, um diese Problemstellungen zu lösen (vgl. Paul 1994). Deshalb habe ich mich verstärkt mit dem Explorationsverhalten von Nutzern und den Möglichkeiten einer technischen Unterstützung der Exploration bei Groupware beschäftigt. Diese Fragestellung war bis dato in der einschlägigen Literatur kaum erforscht.

Der Aufsatz "*Why did that happen? Building Appropriate Mental Models on Groupware Functions*" ist im Frühjahr 1999 entstanden. Er untersucht das Lern- und Explorationsverhalten von Nutzern beim Erlernen einer Groupware und diskutiert dabei festgestellte Probleme. In einer qualitativen Langzeituntersuchung in mehreren Ministerien einer Landesregierung zeigte sich, daß Benutzer einer marktgängigen Groupware erhebliche Probleme hatten, die verschiedenen in vielfältigen oder anpaßbaren Funktionen realisierten Alternativen zu unterscheiden. Dies führte zu erheblichen Problemen bei der Nutzung der Groupware. Obwohl einführende Schulungsmaßnahmen angeboten wurden, spielte Exploration eine wichtige Rolle beim Erlernen der Groupware. Exploratives Lernen von Groupware erfordert den Zugang zur Benutzungsschnittstelle anderer Nutzer, um die von der Ausführung einer Funktion ausgelösten Zustandsübergänge wahrnehmen zu können. Der Zugriff auf die Be-

nutzungsschnittstelle anderer Nutzer kann aber zu Störungen dieser Nutzer führen. Diese empirischen Befunde deuten darauf hin, daß das Erlernen von Groupware neue, über die in der einzelplatzorientierten software-ergonomischen Literatur bisher diskutierten Konzepte hinausgehende Ansätze erfordert.

Einen solchen Ansatz habe ich in meinem Aufsatz "*Exploration Environments: Supporting Users to Learn Groupware Functions*" mit dem Konzept der explorativen Ausführbarkeit entwickelt. Ich ging bei der Konzeptentwicklung von folgenden, teilweise durch die Empirie motivierten Überlegungen aus: Eine Funktion kann nur dann exploriert werden, wenn Nutzer die durch ihre Ausführung ausgelösten Zustandsübergänge an der Ausgabeschnittstelle wahrnehmen können. Das Erlernen anpaßbarer Funktionen ist dadurch erschwert, daß die bei ihrer Ausführung feststellbaren Zustandsübergänge durch die erneute Ausführung von Anpassungsfunktionen veränderbar sind. Sind Anpassungsobjekte nur für einen bestimmten Geltungsbereich definiert, dann müssen die Nutzer ein Verständnis für das in den verschiedenen Nutzungssituationen unterschiedliche Systemverhalten erwerben. Dieser Verständniserwerb ist dann zusätzlich erschwert, wenn für bestimmte Nutzungssituationen inkonsistente Anpassungsobjekte vorliegen, die automatische Resolutionsverfahren erfordern (vgl. Kap. 4.2). Bei gestuften Anpassungsfunktionen ist darüber hinaus der Einfluß der Ausführung einer Anpassungsfunktion zweiter Ordnung auf die Anpassungsfunktion erster Ordnung und die anpaßbare Nutzungsfunktion schwer nachvollziehbar.

Bei der Exploration von Funktionen, die der technischen Organisationsschnittstelle zuzuordnen sind, besteht das zusätzliche Problem, daß die durch ihre Ausführung ausgelösten Zustandsübergänge an der Ausgabeschnittstelle des Aktivators nicht vollständig wahrgenommen werden können. Darüber hinaus können Funktionen, die der technischen Organisationsschnittstelle zuzurechnen sind, in Wechselwirkung zueinander stehen, d. h. die Ausführung bestimmter Funktionen an der Benutzungsschnittstelle eines Nutzers triggert die Ausführung von Funktionen anderer Nutzer. Aus diesem Grunde ist der durch die Ausführung bestimmter Funktionen an der Ausgabeschnittstelle eines anderen Nutzers eintretende Zustandsübergang nur unter Berücksichtigung der Wirkungsweise dort existenter getriggelter Funktionen möglich.

Vor diesem Hintergrund scheint die Implementierung spezieller Funktionen der Dialogschnittstelle wünschenswert, die es erlauben, die Wirkungsweise anpaßbarer Nutzungsfunktionen und (gestufter) Anpassungsfunktionen aus der Perspektive verschiedener betroffener Nutzer auszuprobieren. Dazu muß eine *Explorationsumgebung* implementiert werden, in der die Daten und Anpassungsobjekte der technischen Organisationsschnittstelle enthalten sind, die zur Exploration der durch die Ausführung der Funktion ausgelösten Zustandsübergänge notwendig sind. Außerdem gehören weitere Funktionen zur Explorationsumgebung. Bei anpaßbaren Funktionen können zum einen Anpassungsfunktionen in der Explorationsumgebung enthalten sein. Zum anderen kann es sich dabei um andere Nutzungsfunktionen handeln, deren Ausführung dazu beiträgt, die Wirkungsweise der zu explorierenden Funktion zu verdeutlichen.

Eine Explorationsumgebung für eine Funktion der technischen Organisationsschnittstelle bietet dem Aktivator die Möglichkeit, neben seiner eigenen Benutzungsschnittstelle das Systemverhalten an der Schnittstelle anderer Nutzer wahrzunehmen. Zur Darstellung des Systemverhaltens an der Schnittstelle anderer Nutzer lassen sich ein *deskriptiver* und ein *simulierender* Ansatz unterscheiden. Beim deskriptiven Ansatz wird zur Laufzeit entsprechend den Spezifikationen der zu explorierenden Funktion Freitext erzeugt und dem explorierenden Nutzer angezeigt. Dieser Text beschreibt das

Systemverhalten an der Schnittstelle anderer Nutzer. Beruht die Explorationsumgebung dagegen auf Simulation, so wird die Benutzungsschnittstelle anderer Nutzer an der des Aktivators dargestellt. Durch Wechseln der Perspektiven zwischen verschiedenen möglichen Nutzern und die Aktivierung anderer in der Explorationsumgebung implementierter Funktionen kann der Aktivator die Effekte der Ausführung der zu explorierenden Funktion auf die Schnittstelle anderer Nutzer erkunden.

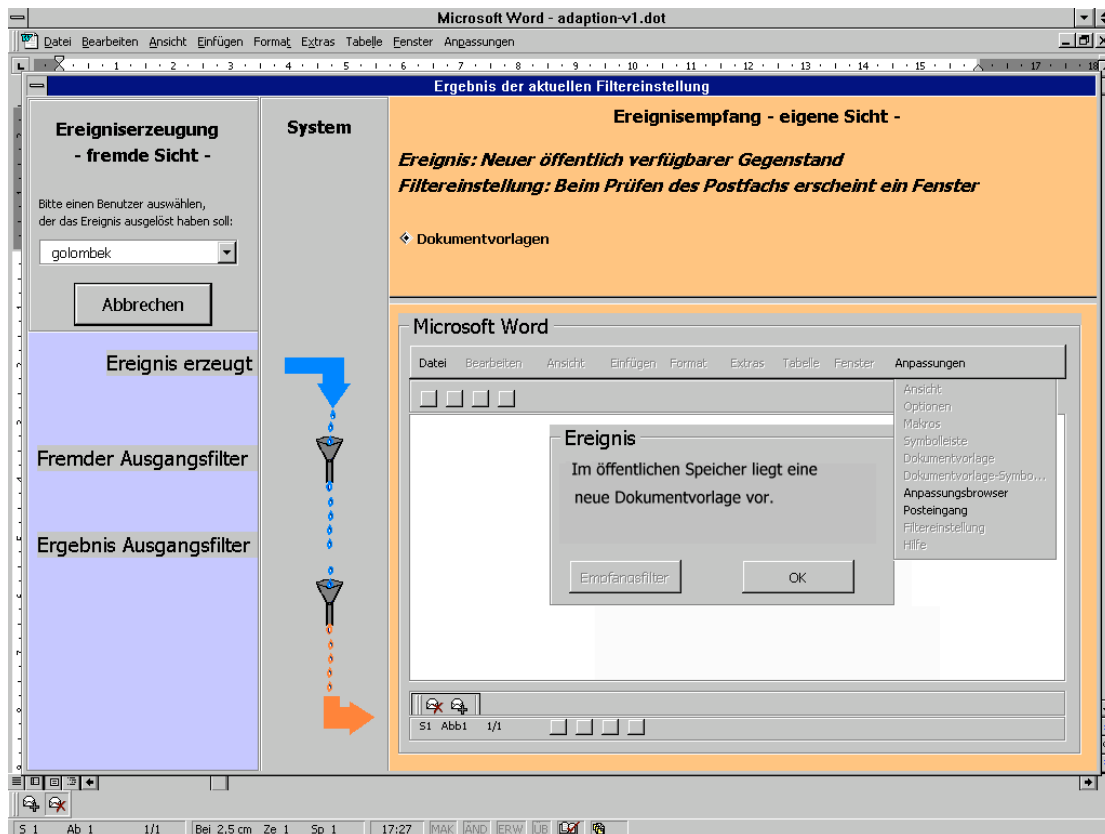


Abb. 6: Explorationsumgebung für den Empfangsfilter

Um bei der Simulation den Zustand der im System befindlichen Daten nicht zu verändern, müssen Spieldaten in der Explorationsumgebung erzeugt werden. Die in der Explorationsumgebung realisierten Funktionen führen auf den Spieldaten Zustandsübergänge aus, die analog zu denen der realen Funktionen sind. Auch die in der Explorationsumgebung nachgebauten Dialog- und Ein- und Ausgabeschnittstellen der ursprünglichen Funktionen sollten ihrem realen Vorbild so ähnlich wie möglich sein. Allerdings können aus didaktischen Gründen und aufgrund der für die gleichzeitige Anzeige zweier simulierter Benutzungsschnittstellen (der des Aktivators und der eines anderen Nutzers) begrenzten Ein- und Ausgabemöglichkeiten Vereinfachungen vorgenommen werden.

Es wurden zwei Beispiele einer auf Simulation beruhenden Explorationsumgebung entwickelt. Beim komponentenbasierten Suchtool für Groupware ist lediglich eine Explorationsumgebung für die Suchfunktionalität und die verschiedenen Stufen der Anpassungsfunktionen realisiert. Wird das ursprüngliche Suchtool während der Exploration angepaßt, so wird der Aktivator am Ende der Exploration befragt, ob er das neu entstandene Suchtool zusätzlich speichern will. Außerdem ist die Zu-

griffskontrollfunktion in den Explorationsumgebungen enthalten, so daß der Aktivator die Effekte der Zugriffsrechte auf das Verhalten des Suchtools untersuchen kann. Beim anpaßbaren Ereignisdienst sind für jeden der Filter und die jeweiligen Anpassungsfunktionen eigene Explorationsumgebungen realisiert. Da es sich bei den Filtern um getriggerte Funktionen handelt, können auch die ihre Ausführung auslösenden Zustandsübergänge in der Explorationsumgebung erzeugt werden. Abb. 6 stellt den Explorationsmodus des Ereignisfilters dar. Der Nutzer erzeugt ein bestimmtes Ereignis und kann dann ausprobieren, bei welcher Filtereinstellung welche Form der Ereignisdarstellung erfolgt.

Am Beispiel der regelbasierten Zugriffskontrolle wird gezeigt, wie Explorationsumgebungen auf deskriptive Weise realisiert werden können. Zu jedem Zeitpunkt der Spezifikation einer auf ein bestimmtes Objekt bezogenen Regel kann sich der Aktivator durch Abfrage der Explorationsumgebung textuell anzeigen lassen, über welche Zugriffsrechte bestimmte andere Nutzer augenblicklich verfügen.

Das Konzept explorativer Ausführbarkeit ist auf qualitative und quantitative Weise evaluiert worden. Die Explorationsumgebung des komponentenbasierten Suchtools wurde mit vier Groupwarenutzern auf einem Workshop evaluiert. Die Nutzer wiesen unterschiedliche Qualifikationsniveaus im Umgang mit Computern auf. Die besser qualifizierten Nutzer fanden die Explorationsumgebung nützlicher, insbesondere um die Wirkungsweise von vordefinierten Suchtools mit mehreren Anzeigefenstern zu verstehen und um selbst erstellte Suchtools zu testen. Allerdings zeigten sich auch Probleme, die sich aus der zusätzlichen Handhabungskomplexität der simulierten Explorationsumgebung ergaben. Die weniger qualifizierten Nutzer fühlten sich deshalb teilweise überfordert. Insofern weisen diese Ergebnisse darauf hin, daß dieses Konzept sich eher als Lernunterstützung für besser qualifizierte Computernutzer eignet als für Anfänger.

Im Aufsatz „*Design for Self - directed Learning - Evaluating the Concepts of Direct Activation and Explorative Execution*“ werden die Ergebnisse einer quantitativen Evaluationsstudie des Konzepts explorativer Ausführbarkeit dargestellt. Im Gegensatz zur qualitativen Evaluationsstudie wurden hierbei die Explorationsumgebung des anpaßbaren Ereignisdienstes untersucht.<sup>8</sup> In der teststatistischen Untersuchung bin ich von der Hypothese ausgegangen, daß die Nutzung von Explorationsumgebungen zur Bildung eines besseren Verständnisses über die Wirkungsweise anpaßbarer Funktionen und dadurch zu erfolgreicherem und schnellerem Anpassungsaktivitäten führen würde. Außerdem vermutete ich, daß Explorationsumgebungen das subjektive Sicherheitsgefühl der Nutzer hinsichtlich der von ihnen durchgeführten Anpassungsaktivitäten fördern würden.

Um diese Hypothese zu testen, wurde im Rahmen des Experiments die Nutzung zweier Versionen des anpaßbaren Ereignisdienstes untersucht. In der ersten Version waren die Explorationsumgebungen implementiert, in der anderen Version waren diese nicht realisiert. Die Versuchsteilnehmer mußten mit ihrer jeweiligen Version der Groupware zwei Anpassungs- und drei Verständnisaufgaben bearbeiten. Als Ergebnis dieser Untersuchung läßt sich feststellen, daß die in den Hypothesen formulierten Annahmen über die positiven Effekte von Explorationsumgebungen nachgewiesen werden konnten. Insofern haben die qualitativen und quantitativen Evaluationsstudien das Konzept explorativer Ausführungen bestätigt. Allerdings deuten die Ergebnisse der

---

<sup>8</sup> In dieser Studie wurde auch das Konzept direkter Aktivierbarkeit evaluiert (vgl. Kap. 5.1).

qualitativen Evaluationsstudie daraufhin, daß die Nützlichkeit von Explorationsumgebungen vom Qualifikationsniveau der Nutzer abzuhängen scheint.

## 6 Diskussion

Die Gestaltung anpaßbarer Groupware ist ein vielschichtiges Problem. Die in dieser Habilitationsschrift zusammengefaßten Aufsätze verbinden Untersuchungen über den Gestaltungs- und Einführungsprozeß von Groupware mit solchen zur software-ergonomischen Gestaltung. Das erweiterte IFIP-Modell bietet die zur Einordnung der Beiträge notwendige Terminologie. Aufgrund des selbstorganisierten Charakters von Organisationen und daraus resultierenden, nicht vollständig antizipierbaren Nutzungsmöglichkeiten von Groupware wird für deren Einwicklung und Einsatz ein integriertes, evolutionäres und partizipatives Prozeßmodell vorgeschlagen. Dabei spielt Anpaßbarkeit als Eigenschaft einer Groupware eine entscheidende Rolle. Am Beispiel dreier Funktionalitäten der technischen Organisationsschnittstelle wurden differenzierte Anforderungen an eine anpaßbare Systemgestaltung erhoben, Konzepte anpaßbarer Gestaltung dieser Funktionalitäten entwickelt und diese beispielhaft implementiert. Um Nutzer beim Umgang mit anpaßbar gestalteter Groupware zu unterstützen, wurden basierend auf empirischen Voruntersuchungen zwei Konzepte für die Gestaltung der Dialogschnittstelle entwickelt und beispielhaft implementiert.

Bei der in dieser Arbeit angewandten Forschungsmethodik basieren wissenschaftliche Konzeptbildung und innovative Implementierungen immer auf der Beteiligung von Nutzern bzw. potentiellen Nutzern. Die wissenschaftliche Leistung besteht darin, geleitet durch geistes- und sozialwissenschaftliche Theorien, aus den empirisch feststellbaren Einzelfällen allgemeiner gültige Konzepte zu entwickeln und im Hinblick auf ihre Wirksamkeit zu evaluieren. Die in dieser Arbeit entwickelten Gestaltungskonzepte sind so dargestellt, daß sie in einer Vielzahl von Anwendungen umgesetzt werden können. Eine solche Umsetzung erfordert allerdings deren kontextbezogene Spezifizierung. In dieser Arbeit ist dies beispielhaft im Rahmen der verschiedenen Implementierungen erfolgt.

Der in dieser Schrift niedergelegte Stand der Arbeiten weist gewisse Defizite hinsichtlich der Evaluation der entwickelten Konzepte auf. Das Prozeßmodell integrierter Organisations- und Technikentwicklung wurde außerhalb dieser Arbeit in einem Projekt evaluiert, bei dem es galt, räumlich verteilte Konstruktionsprozesse zwischen einem Hüttenwerk und zwei externen Ingenieurbüros zu verbessern (vgl. Wulf u. a. 1999). Hinsichtlich der in Kap. 4 dargestellten Konzepte die für Gestaltung anpaßbarer Groupware wurden bisher lediglich das anpaßbare Suchtool für Groupware einem Feldtest unterzogen. Die erweiterte Zugriffskontrolle wurde neben den in dieser Arbeit dargestellten Evaluationsschritten in mehreren Workshops mit Nutzern evaluiert (vgl. Wulf, Pfeifer und Pipek 1998). Die Konzepte zur Gestaltung der Dialogschnittstelle anpaßbarer Groupware sind dagegen relativ gut evaluiert. Das Konzept explorativer Ausführbarkeit ist sowohl qualitativ als auch quantitativ untersucht worden. Das Konzept direkter Ausführbarkeit ist zumindestens durch eine teststatistische Untersuchung evaluiert.

Hinsichtlich der Verallgemeinerbarkeit solcher Evaluationsergebnisse besteht das prinzipielle Problem, daß übertragbare Gestaltungskonzepte immer nur in Form einer konkreten Umsetzung in speziellen Nutzungskontexten untersucht werden können. Bei der Interpretation der Evaluationsergebnisse besteht dann das Problem zu unterscheiden, ob diese sich auf das Gestaltungskonzept, dessen konkrete Implementie-

rung, die Besonderheiten der evaluierenden Nutzergruppe oder anderer Gegebenheiten des Anwendungskontextes zurückführen lassen. Insofern sind die erzielten Evaluationsergebnisse kontextbezogen zu interpretieren.

Trotz dieser methodischen Einschränkungen scheint es zu dem hier gewählten Weg einer evolutionären und partizipativen Softwareentwicklung und der darauf aufbauenden theoriegeleiteten Konzeptentwicklung keine Alternative zu geben. Bei interaktiven Anwendungen, die in hohem Maße mit den Arbeitsprozessen ihrer Nutzer interagieren, können nur auf diese Weise problemangemessene und innovative Gestaltungskonzepte entwickelt werden. Wie in dieser Arbeit gezeigt wurde, gilt dies insbesondere für die anpaßbare Gestaltung von Groupware.

## Literatur

- Ackermann, D.; Ulich, E.: The chances of individualization in human-computer interaction and its consequences. In: Frese, M.; Ulich, E.; Dzida, W. (eds), Psychological issues of human computer interaction in the work place, North Holland, Amsterdam 1987, S. 131 – 146
- Akscyn, R.; Yoder, E.; McCracken, D.: The data modell is the heard of the interface design, in: Proceedings of CHI'88, Ellen Frisbie, Chicago 1988, S. 115 - 120
- Banavar, G.; Doddapaneni, S.; Miller, K.; Mukherjee, B.: Rapidly Building Synchronous Collaborative Applications By Direct Manipulation, in: Proceedings of CSCW '98, ACM-Press, New York, 1998, S. 139 – 148
- Bannon, L.; Schmidt, K.: CSCW: Four Characters in Search of a Context. In: Bowers, J.M.; Benford, S.D. (eds): Studies in Computer Supported Cooperative Work - Theory, Design, Practice, North Holland, Amsterdam 1991, S. 3 –16
- Belotti, V.; Sellen, A.: Design for Privacy in Ubiquitous Computing Environment. In: de Michelis, G.; Simone, C.; Schmidt, K. (eds): Proceedings of the Fourth Conference on Computer Supported Cooperative Work - ECSCW '93, Kluwer, Dordrecht, 1995, S. 77 - 92
- Bentley, R.; Dourish, P.: Medium versus Mechanism. Supporting Collaboration Through Customisation, in: Marmolin, H.; Sundblad, Y.; Schmidt, K. (Hrsg.), Proceedings of the Fourth European Conference on Computer Supported Cooperative Work - ECSCW '95, Kluwer, Dordrecht 1995, S. 133-148
- Carroll, J. M.; Mack, R. L.: Actively learning to use a word processor, in: Cooper, W. E. (ed.): Cognitive aspects of skilled typewriting, Springer: New York, 1983, S. 259 – 281
- Carroll, J. M.; Mack, R. L.: Learning to use a word processor by doing, by thinking, by knowing, in: Thomas, J. C.; Schneider, M. L. (eds): Human factors in computer systems, Ablex: Norwood, NJ 1984, S. 13 – 51
- Darlington, J.; Dzida, W.; Herda, S.: The role of excursions in interactive systems, International Journal of Man-Maschine Studies, Vol. 18, 1983, S. 101 – 112
- Dutke, S.: Mentale Modelle: Konstrukte des Wissens und Verstehens, VAP, Göttingen 1994
- Dzida, W.: Das IFIP-Modell der Benutzerschnittstelle, in: Office Management, Sonderheft 31, 1983, S. 6 – 8
- Dzida, W.: Modellierung und Bewertung von Benutzerschnittstellen. In: Software Kurier für Mediziner und Psychologen, Nr. 1, 1988, S. 13 - 28
- Dzida, W.; Herda, S.; Itzfeld, W.-D.: User-perceived quality of interactive systems, in: IEEE Transactions on Software-Engineering, Vol. 4, No. 4, 1978, S. 270 - 276
- Ellis, C. A.; Wainer, J.: A Conceptual Model of Groupware, in: Proceedings of CSCW '94, ACM-Press, New York 1994, S. 79 – 88
- Fertig, S.; Freeman, E.; Gelernter, D.: „Finding and Reminding“ Reconsidered, in: SIGCHI Bulletin, Vol. 28, No. 1, 1996, S. 66 - 69
- Fischer, G.; Girgensohn, A.: End-User Modifiability in Design Environments. In: Proceedings of the Conference on Computer Human Interaction (CHI '90), April 1 - 5, 1990, Seattle, Washington, ACM-Press, New York 1990, S. 183 - 191
- Floyd, Ch. 1994: Evolutionäre Systementwicklung und Wandel in Organisationen, in: GMD-Spiegel, Vol. 24, 3, S. 36 – 40
- Floyd, Ch; Reisin, F.-M.; Schmidt, G.: STEPS to software development with users. In: Ghezzi, C.; McDermid, J.A. (eds.): ESEC '89 - 2nd European Software Engineering Conference, University of Warwick, Coventry. Lecture Notes in Computer Science No. 387, Springer, Heidelberg u. a. 1989, S.

48 – 64

- Frese, M.; Albrecht, K.; Altmann, A.; Lang, J.; Papstein, P. V.; Peyerl, R.; Prümper, J.; Schulte-Göcking, H.; Wankmüller, I.; Wendel, R.: The effects of an active development of the mental model in the training process: experimental results in a word processing system, in: Behaviour and Information Technology, Vol. 7, No. 3, 1988, S. 295-304
- Friedrich, J.: Adaptivität und Adaptierbarkeit informationstechnischer Systeme in der Arbeitswelt - Zur Sozialverträglichkeit zweier Paradigmen. In: Reuter, A. (Hrsg.): Proceedings der 20 GI-Jahrestagung, Springer, Heidelberg u. a. 1990, S. 178 - 191
- Fuchs, L.: Situationsorientierte Unterstützung von Gruppenwahrnehmung in CSCW-Systemen, PhD-Thesis, Department of Computer Science, University of Essen, 1997
- Fuchs, L.; Sohlenkamp, M.; Genau, A.; Kahler, H.; Pfeifer, A.; Wulf, V.: Transparenz in kooperativen Prozessen: Der Ereignisdienst in POLITeam. In: Krcmar, H.; Lewe, H.; Schwabe, G. (Hrsg.): Herausforderung Telekooperation (Proceedings of the DCSCW '96), Springer, S. 3 - 16
- Greenberg, S.: Computer-supported Cooperative work and Groupware. In: Greenberg, Saul (ed.): Computer-supported Cooperative Work and Groupware, Academic Press, London 1991a, S. 1 - 8
- Haaks, D.: Anpaßbare Informationssysteme - Auf dem Weg zu aufgaben- und benutzerorientierter Systemgestaltung und Funktionalität, Göttingen u.a. 1992
- Hacker, W.: Software-Ergonomie – Gestalten rechnergestützter geistiger Arbeit?! In: Schönplflug, W.; Wittstock, M. (Hrsg.): Software-Ergonomie '87, Teubner, Stuttgart 1987, S. 31 - 45
- Henderson, A.; Kyng M. (1991): There's No Place Like Home. Continuing Design in Use. in: Design at Work, Lawrence Erlbaum Associates, Publishers, S. 219-240
- Henderson, A.: A Conversation with Austin Henderson. Interviewed by Kate Ehrlich, interactions, No. 6, 1998, S. 36 – 47
- Herrmann, Th.: Zur Gestaltung der Mensch-Computer-Interaktion: Systemerklärung als kommunikatives Problem, Niemeyer, Tübingen 1986
- Herrmann, Th.: Grundsätze ergonomischer Gestaltung von Groupware. In: Hartmann, A.; Herrmann, Th.; Rohde, M.; Wulf, V. (Hrsg.): Menschengerechte Groupware - Software-ergonomische Gestaltung und partizipative Umsetzung, Teubner, Stuttgart 1994, S. 65 -108
- Herrmann, Th.; Just, K.: Anpaßbarkeit und Aushandelbarkeit als Brücke von der Software-Ergonomie zur Organisationsentwicklung. In: Hasenkamp, U. (Hrsg.): Einführung von CSCW-Systemen in Organisationen, Vieweg, Braunschweig u.a. 1994, S. 89 - 109
- Höller, H.: Kommunikationssysteme - Normung und soziale Akzeptanz, Vieweg: Braunschweig u.a. 1993
- ISO 9241: Ergonomic requirements for office work with visual display terminals (VDTs) Part 10: Dialogue Principles
- Johnson-Lenz, P.; Johnson-Lenz, T.: Post-mechanistic groupware primitives: rhythms, boundaries and containers, in Greenberg, Saul (eds): Computer-supported Cooperative Work ae., Academic Press, London, 1991, S.271-293
- Kahler, H.; Stiemerling, O.; Wulf, V.; Höpfner, G.: Gemeinsame Anpassung von Einzelplatzanwendungen, in: Proceedings of Software-Ergonomie '99, Teubner, Stuttgart, 1999b, S. 183 - 194
- Kamouri, A. L.; Kamouri, J.; Smith, K. H.: Training by exploration: facilitating the transfer of procedural knowledge through analogical reasoning, in: Journal of Man-Machine Studies, Vol. 24, 1986, S. 171-192
- Karsten, H.; Jones, M.: "The long and winding road: Collaborative IT and organisational change"; in: Int. Conference on Computer Supported Work (CSCW'98); ACM Press; Seattle, WA, USA; 1998, S. 29-38
- Koch, M.; Reiterer, H.; Tjoa, A M.: Software-Ergonomie. Gestaltung von EDV-Systemen – Kriterien, Methoden und Werkzeuge, Springer, Wien u.a. 1991
- Lefrancois, G. R.: Psychological Theories and Human Learning: Kongors Report, Wadsworth: Belmont, CA 1972
- Luhmann, N. 1984: Soziale Systeme: Grundriß einer allgemeinen Theorie, Suhrkamp: Frankfurt
- Maaß, S.: Strömungen, Leitbilder und Begrifflichkeit der Software-Ergonomie – Beiträge zur Entwicklung und Systematisierung eines Fachgebietes, Habilitationsschrift, Universität Hamburg, 1995
- Mackay, W. E.: Users and customizable Software: A Co-Adaptive Phenomenon, PhD-Theses, MIT, Boston (MA) 1990
- MacLean, A.; Carter, K.; Lövfstrand, L.; Moran, T: User-tailorable Systems: Pressing the Issue with Buttons, in: Proceedings of the Conference on Computer Human Interaction (CHI '90), April 1-5, 1990, Seattle, Washington, ACM-Press, New York 1990, S. 175 –182
- Mark, G.: Emerging Multiple Perspectives in Groupware Use: Intra and Intergroup Conventions, in:

- Proceedings of GROUP '97, ACM-Press, New York, 1997, S. 19 - 28
- Mark, G.; Wulf, V.: Changing Interpersonal Communication through Groupware Use, accepted by: Behaviour & Information Technology, Vol. 18, No. 5, 1999, S. 385 - 395
- Mark, G., Fuchs, L., and Sohlenkamp, M. (1997). Supporting Groupware Conventions through Contextual Awareness. *Proceedings of ECSCW'97*, Kluwer Academic Publishers, Dordrecht 1997, S. 253-268
- Maturana, H.; Varela, F., 1980: Autopoiesis and Cognition, Reidel, Dordrecht
- Maturana, H. R.; Varela, F. J.: Der Baum der Erkenntnis, Die biologische Wurzeln menschlichen Erkennens, Scherz, Bern und München 1987 (im spanischen Original: El árbol del conocimiento, 1984)
- Mørch, A.: Method and Tools for Tailoring of Object-oriented Applications: An Evolving Artifacts Approach, PhD-Thesis, University of Oslo, Department of Computer Science, Research Report 241, Oslo 1997
- Nardi, B.; Barreau, D.: Finding and Reminding: File Organisation from the desktop, in: SIGCHI Bulletin, Vol. 27, No. 3, 1995, S. 39 - 43
- Nardi, B.; Barreau, D.: „Finding and Reminding“ Revisited: Appropriate Metaphors for File Organisation at the Desktop, in: SIGCHI-Bulletin, Vol. 29, No. 1, 1997, S. 76 - 78
- Oberquelle, H.: Kooperative Arbeit und menschengerechte Groupware als Herausforderung für die Software-Ergonomie. In: Oberquelle, H. (Hrsg.): Kooperative Arbeit und Computerunterstützung, VAP, Göttingen 1991, S. 1 - 10
- Oberquelle, H.: Anpaßbarkeit von Groupware als Basis für die dynamische Gestaltung von computergestützter Gruppenarbeit. In: Konradt, U.; Drisis, L.: Benutzeroberflächen in der teilautonomen Arbeit, Köln 1993, S. 37 - 54
- Oberquelle, H.: Situationsbedingte und benutzerorientierte Anpaßbarkeit von Groupware, in: Hartmann, A. u. a. (Hrsg.), Menschengerechte Groupware, Stuttgart, 1994, S. 31-50
- Oberquelle, H. u.a.: Mensch & Computer 2000 – Information, Interaktion, Kooperation, in: Informatik Spektrum, 32. Jhg., Nr. 3, 1999, S. 212 – 214
- Oppermann, R.: Individualisierte Systemnutzung. In: Paul, M. (Hrsg.): Proceedings der 19. GI-Jahrestagung, Band 1, Springer, Heidelberg u. a. 1989, S. 131 - 145
- Oppermann, R.; Murchner, B.; Reiterer, H.; Koch, M.: Software-ergonomische Evaluation. EVADIS II. Walter de Gruyter, Berlin 1992
- Oppermann, R.; Simm, H.: Adaptability: User-Initiated Individualization, In: Oppermann, R. (ed.): Adaptive User Support – Ergonomic Design of Manually and Automatically Adaptable Software, LEA, Hillsdale, NJ 1994
- Orlikowski, W. J. (1996): Evolving with Notes: Organizational change around groupware technology, in: Ciborra, C.: *Groupware & Teamwork*, Chichester: J. Wiley, S. 23 - 60.
- Paetau, M.: Konfigurative Technik und die Dynamik sozialer Systeme - ein Überblick über die Gestaltungsproblematik und ein Lösungsvorschlag. In: Müller, W.; Senghaas-Knobloch, E. (Hrsg.): Arbeitsorientierte Technikbewertung und Softwaregestaltung - Leitbilder, Methoden und Werkzeuge. Münster u. a. 1993, S. 149 - 180
- Paul, H.: Exploratives Agieren, Peter Lang, Frankfurt/M 1994
- Rauterberg, M.: Ein Konzept zur Quantifizierung software-ergonomischer Richtlinien, Dissertationsschrift, Universität Zürich, Zürich 1995
- Rödiger, K.-H. (Hrsg.): Software-Ergonomie '93, Teubner, Stuttgart 1993
- Stiemerling, O.; Cremers, A. B.: Tailorable Component Architectures for CSCW-Systems, in: Proceedings of the 6th Euromicro Workshop on Parallel and Distributed Programming, Jan 21-24, 1998, Madrid, Spain, IEEE Press, S. 302-308
- Stiemerling, O.; Kahler, H. and Wulf, V.: How to Make Software Softer - Designing Tailorable Applications. In Proceedings of 2<sup>nd</sup> Conference on the Design of Interactive Systems, Amsterdam (NL), ACM Press 1997, S. 365-376
- Syri, A.: Entwurfsmuster für CSCW-Systeme, Shaker, Aachen 1999 (zugl.: Dissertationsschrift Universität Linz 1999)
- Teege, G.: Individuelle Groupware - Gestaltung durch Endbenutzer, DUV, Wiesbaden 1998 (zugl.: Habilitationsschrift, TU München, 1998)
- Trigg, R.H.; Moran, T.P.; Halasz, F.G.: Adaptability and tailorability in Note Cards. In: Bullinger, H.J., Shackel, B. (eds): Herman-Computer Interaction - INTERACT '87, Elsevier, Amsterdam, 1987, S. 723 - 728
- Ulich, E.: Über das Prinzip der differentiellen Arbeitsgestaltung. In: Industrielle Organisation, 47. Jg., Nr. 12, 1978, S. 566 - 568
- Winograd, T.; Flores, F.: Understanding Computers and Cognition, 3. Aufl., Addison - Wesley,

- Norwood, NJ 1988
- Wulf, V.: *Konfliktmanagement bei Groupware*, Vieweg, Braunschweig und Wiesbaden 1997
- Wulf, V.; Hartmann, A.: The Ambivalence of Network's Visibility in an Organizational Context. In: Clement, A.; Kolm, P.; Wagner, I. (eds): *NetWORKing: Connecting Workers in and between Organizations*, North Holland, Amsterdam 1994, S. 143 – 152
- Wulf, V.; Mark, G.: The Emergence of Conventions within Processes of Organizational and Technological Change, in: *Proceedings of the 7th International Conference on Human-Computer Interaction*, Elsevier, Amsterdam 1997, S. 293 – 296
- Wulf, V.; Krings, M.; Stiernerling, O.; Iacucci, G.; Fuchs-Fronhofen, P.; Hinrichs, J.; Maidhof, M.; Nett, B.; Peters, R.: Improving Inter-Organizational Processes with Integrated Organization and Technology Development, in: *JUCS - Journal of Universal Computer Science* 1999, Vol. 5, No. 6, 1999
- Wulf, V.; Pfeifer, A.; Pipek, V.: *What you want and what you get – resolving conflicts in groupware*, Arbeitspapier, Institut für Informatik, Universität Bonn, 1998