

Konfiguration und Präsentation komplexer Anwendungen

Udo Arend und Edmund Eberleh

SAP AG, Usability Group

Zusammenfassung

In diesem Beitrag wird ein Ansatz zur modellbasierten Konfiguration eines Endbenutzersystems vorgestellt, der bestimmte Nachteile von adaptierbaren Systemen, adaptiven Systemen oder auch intelligenten Agenten vermeidet. Aufgrund von Metadaten zur betriebswirtschaftlichen Struktur einer Organisation als auch aufgrund von Aufgabenbeschreibungen können Systemeinstellungen explizit vorgenommen werden. Der Endbenutzer wählt diejenige Sicht auf den Anwendungsumfang der Software, der für ihn relevant ist, und erhält dadurch eine einfach zu bedienende Benutzungsoberfläche. Die in diesem Beitrag beschriebene neuartige Präsentationskomponente, der sogenannte „Session Manager“, bietet neben dem einfachen Einstieg in Anwendungen noch eine intuitive, weitergehende Individualisierungsmöglichkeit. Die Vorteile des gewählten Ansatzes, die Software an die Arbeitsabläufe innerhalb von Organisationen anzupassen, werden dargestellt.

1 Problemstellung

Die Zahl der verfügbaren Anwendungsprogramme und Utilities nimmt ständig zu, ebenso wie die Funktionen der Anwendungsprogramme und Betriebssysteme selbst. Gleichzeitig steigt die Zahl der Benutzer von weltweit angebotener Software in den dreistelligen Millionenbereich. Die Softwarehersteller stehen vor folgenden Problemen: der Benutzer soll „seine“ Programme schnell innerhalb der großen Menge anderer ebenfalls vorhandener finden, er soll innerhalb dieser Programme „seine“ Funktionen schnell auslösen können, und er soll Aussehen und Organisation der Programme und Funktionen an seine ästhetischen und kognitiven Strukturen anpassen können. Der software-ergonomischen Lösung dieser Probleme kommt beim Neuentwurf eines Produktes mittlerweile hohe Bedeutung zu [18]. Es lassen sich zur Zeit mindestens zwei gegensätzliche Lösungsstrategien hierfür identifizieren: Einstellung durch direkten Eingriff seitens des Benutzers und Einstellung durch „intelligente“ Systemagenten. Wir argumentieren, daß beide Strategien aus einem arbeitswissenschaftlichen Blickwinkel betrachtet wesentliche Aspekte software-ergonomischer Systemgestaltung vernachlässigen. Nach einer Diskussion dieser Ansätze werden wir am Beispiel des SAP R/3-Systems einen dritten Weg zur Konfiguration und Präsentation komplexer Anwendungssysteme aufzeigen, der auf Aufgabenbeschreibungen basiert.

Diese Arbeit knüpft bzgl. des Präsentationsaspekts an Überlegungen von Eberleh & Meinke [4] an. Die nachfolgend dargestellte Arbeit verändert und konkretisiert die dort beschriebenen Überlegungen allerdings in wesentlichen Teilen und verbindet sie mit einem Konfigurationsverfahren, welches bisher noch nicht thematisiert wurde. Im folgenden wird daher nur eine Übersicht über bestehende Ansätze und Probleme bei der Konfiguration komplexer Anwendungssysteme gegeben. Für eine Übersicht über die Präsentationsprobleme und Lösungen unterschiedlicher Betriebssysteme sei auf Eberleh & Meinke [4] verwiesen.

2 Bestehende Ansätze zur Konfiguration komplexer Anwendungssysteme

2.1 Put the User in Control !?

Aus der psychologischen Forschung, z.B. [17], ist seit längerem bekannt, daß subjektiv wahrgenommene fehlende Kontrolle über die Umwelt irritierend, belastend und motivationsmindernd wirkt. Diese Erkenntnis liegt letztlich den software-ergonomischen Forderungen nach Steuerbarkeit, Transparenz und Erwartungskonformität von Softwaresystemen zugrunde [5, 19] und hat im Prinzip der „Direkten Manipulation“ und im „WYSIWYG-Prinzip“ Einfluß auf die Gestaltung heutiger Softwaresysteme genommen [1]. Die Verletzung dieser Prinzipien oder „usability heuristics“ trägt nach einer Analyse von Nielsen [14] wesentlich zur Erklärung von Benutzbarkeitsproblemen bei. Die erklärungsstärksten ersten vier Faktoren hierfür lauten: (1) Visibility of system status, (2) Match between system and real world, (3) User control and freedom, (4) Consistency and standards.

Nach diesen Prinzipien gestaltete Systeme bieten dem Benutzer grafisch eine möglichst vertraute Abbildung seiner bekannten Umwelt und deren Objekte an und erlauben ihm eine möglichst direkte und sofort sichtbare Änderung dieser Abbildung. Programmsymbole können in diesem Paradigma direkt an den gewünschten Ort verschoben werden und auf diese Weise sichtbarer und zugreifbarer gemacht werden oder versteckt werden, weil weniger wichtig oder häufig genutzt. Da die direkte Manipulation in vielen Fällen an Grenzen stößt, wird das Sichtbar- bzw. Unsichtbarmachen von Programmen oder Funktionen oft zusätzlich oder ausschließlich indirekt mittels Menüs und Funktionstasten angeboten. Neben dieser Wahl der Anordnung und Sichtbarseins von Arbeitsobjekten kann der Benutzer in der Regeln noch die grafische Repräsentation vieler Objekte beeinflussen. Zentrales Merkmal eines derartigen Systems ist, daß das System solange unverändert bleibt, bis der Benutzer mittels der genannten Möglichkeiten explizit die Systemkonfiguration ändert, es somit ein adaptierbares System ist [15]. Dieses Prinzip der Adaptierbarkeit führt aus mehreren Gründen zu Problemen, wie am Beispiel von Windows 95 deutlich wird (Abb. 1):

1. Es werden redundante Möglichkeiten für die gleiche Aufgabe angeboten: Programme können gleichzeitig im Startmenü, in der Office-Shortcutleiste, in der Task-Bar, im Benutzermenü des Startmenüs, auf dem Desktop und im Ordnerfenster abgelegt werden, ermöglicht durch Objektkopien. Es ist fraglich, ob Benutzer mit diesem Konzept der Objektreferenzen tatsächlich eine Kontrolle der Umwelt bekommen.
2. Viele Einstellungsprobleme werden auf den Benutzer verlagert, in der Annahme, daß der Benutzer selbst am besten weiß, was für ihn optimal geeignet ist. Dieses ist unserer Meinung nach bei der Art und Fülle heute angebotener Einstellungsmöglichkeiten zu hinterfragen. In den Prüfgrundsätzen der Unfallverhütungsvorschrift [7] geht man folgerichtig nur noch davon aus, daß ein System in *einer* Einstellung der Vorschrift genügen muß, um das Prüfsiegel zu bekommen. Was dann vor Ort der Benutzer am System verändert, entzieht sich völlig der Kontrolle des Herstellers, so daß ein ergonomie-geprüftes System in der Praxis völlig zum Ungünstigen hin verstellt werden kann.
3. Die Einstellungsdialoge erhöhen zumindest initial den Aufwand im Bereich des Interaktionsproblems. Inwieweit längerfristig dadurch tatsächlich die beabsichtigte Optimierung im Arbeiten mit dem Softwaresystem eintritt, ist gerade unter Berücksichtigung von Punkt 2 fraglich.

4. Die Einstellungen müssen von jedem Benutzer individuell neu vorgenommen werden. In manchen Fällen werden zwar Voreinstellungen oder Paletten mitgeliefert. Diese bieten eine gewisse Erleichterung bei der Einzelauswahl, geben aber trotzdem kaum einen Hinweis auf die Angemessenheit der Wahl.

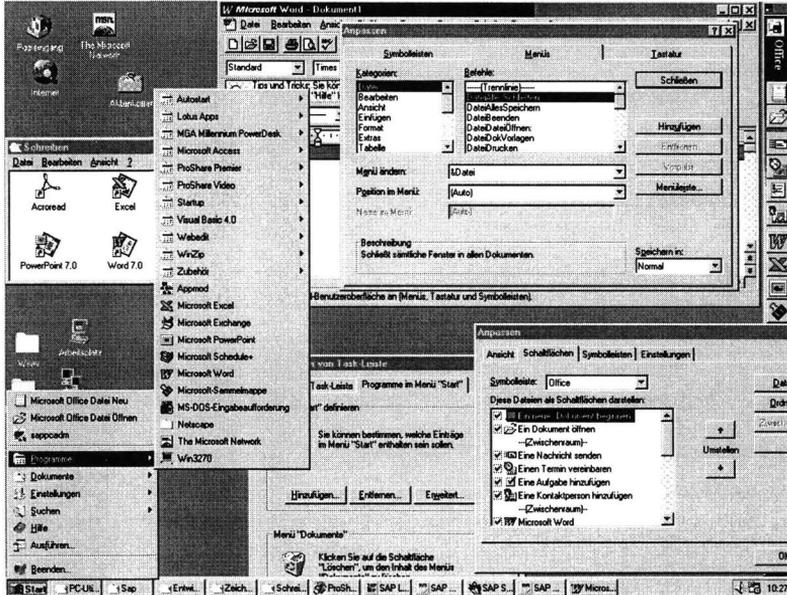


Abb. 1: Konfigurationsmöglichkeiten von Anwendungen in Windows 95

Insgesamt stehen den positiven Möglichkeiten einer adaptierbaren Benutzungsschnittstelle somit vielfältige Kosten gegenüber, die ein exaktes Bewerten dieses Gestaltungsprinzips auch bei empirischen Untersuchungen erschweren und bisher unklar lassen [11].

2.2 Don't bother the User with all Tasks !?

Eine weitere, bereits seit längerem bestehende arbeitspsychologische Forderung besteht darin, die Aufgaben- und Funktionsverteilung zwischen Mensch und Rechner ausdrücklich im Gestaltungsprozeß eines Mensch-Rechner-Systems mit zu berücksichtigen [8, 10]. Ein Teil dieser Forderung ist in das Prinzip der Aufgabenangemessenheit eingeflossen, welches u.a. aussagt, daß der Benutzer bei seiner Aufgabenerledigung durch das System adäquat unterstützt wird, indem es ihm bestimmte Routinetätigkeiten oder für die Sachaufgabe nicht notwendige Tätigkeiten abnimmt [5]. Um dieses zu erreichen, ist ein Modell über den Aufgabenbereich und -ablauf und den Benutzer erforderlich. Derartige Modelle können zur Erreichung von Aufgabenangemessenheit in zwei Richtungen eingesetzt werden: (1) zur Verringerung des Interaktionsproblems, indem der Computer die Präsentation und Interaktion vollständig an den Benutzer optimal anpaßt (adaptives System), und (2) zur Unterstützung bei der Sachaufgabe, indem der Computer dem Benutzer schwierige oder langweilige Tätigkeiten selbständig abnimmt (intelligente Agenten). In beiden Fällen verliert der Benutzer mehr oder

weniger die Kontrolle über das Verhalten des Computersystems. Es liegt somit ein Konflikt bezüglich verschiedener software-ergonomischer Gestaltungskriterien vor. Ein Vorschlag zur Lösung des Konfliktes zwischen Flexibilität und Transparenz besteht im Anbieten einer „Schalttafel“, wo sämtliche Modifikationsmöglichkeiten mit dem aktuellen Zustand ersichtlich sind und auf Wunsch geändert werden können [6]. Eine ausschließliche, automatische Anpassung und Einstellung der Benutzungsoberfläche durch das System selbst wird denn auch als kritisch eingeschätzt [16], ist jedoch trotzdem Gegenstand aktueller Arbeiten zur Softwareergonomie und kontextsensitiven Hilfesystemen [2]. Kontrovers verhält es sich ebenfalls bezüglich der Arbeiten zu intelligenten Agenten. In einer Reihe von Studien werden Agenten als Lösung zur Bewältigung der Informationsflut verwendet [12] und werden auch zur systemgesteuerten Auswahl der für den Benutzer sinnvollen Funktionalität eingesetzt [3]. Der Benutzer gibt somit als Preis einer effektiven Aufgabenerledigung einen wesentlichen Teil seiner Autonomie an das System ab, was von anderen Autoren als äußerst kritisch gesehen wird [9].

2.3 The User wants to get his Job done !

Im folgenden möchten wir einen dritten Weg zur Unterstützung des Benutzers beim Einstellen seines Systems aufzeigen. Diese Alternative basiert auf der Annahme, daß ein Benutzer in einer Arbeitssituation (nur dieser Anwendungskontext liegt den weiteren Überlegungen zugrunde) in der Regel nicht die Zeit und Lust hat, sich mit den vielfältigen Individualisierungsmöglichkeiten seines Rechners intensiv zu beschäftigen, da diese Tätigkeiten nichts mit seiner eigentlichen Aufgabe zu tun haben. Idealerweise sollte der Benutzer eine Software vorfinden, welche für seine persönliche Arbeitstätigkeit bereits so gut wie möglich angepaßt ist, so daß er nur noch geringfügige und aktiv kontrollierte Anpassungen vornehmen muß. Die Vorkonfiguration des Systems sollte nicht in kaum nachvollziehbarer Weise dynamisch durch einen „intelligenten“ Agenten geschehen, sondern auf der Grundlage einer expliziten und damit diskutierbaren Aufgabenbeschreibung des Benutzers. Die „Intelligenz“ eines Softwaresystems sollte unserer Meinung nach ausschließlich dazu dienen, die von den Benutzern als sinnvoll definierte Organisations- und Aufgabenstruktur in ein leicht benutzbares Softwarewerkzeug zur Unterstützung dieser Inhalte umzusetzen. Wie eine derartige Lösung konkret aussehen kann, stellen wir nachfolgend am Beispiel des SAP R/3-Systems dar.

3 Kurze Charakterisierung des R/3-Systems und Probleme

Das R/3-System der SAP AG ist ein Softwarepaket für integrierte betriebswirtschaftliche Standardanwendungen. Es ist modular aufgebaut und deckt den gesamten Bereich betrieblicher Funktionen ab. Während das R/3-System nach außen als eine abgeschlossene Einheit im Desktop des jeweiligen Front-End-Geräts erscheint, stellt es nach innen ein sehr komplexes Paket vieler miteinander vernetzter Anwendungen und Daten dar. In der Vergangenheit (bis Release 3.0) wählte ein Benutzer seine Funktionen folgendermaßen aus: wenn er den Namen (Transaktionscode) der Anwendung kennt, konnte er ihn direkt in einem entsprechenden Feld eingeben (Gedächtnisproblem!). Der normale Weg war jedoch, die Anwendung durch Klicken mit der Maus aus einem Menübaum herauszusuchen (vgl. Abb. 2) wie beim Startmenü von Windows 95.

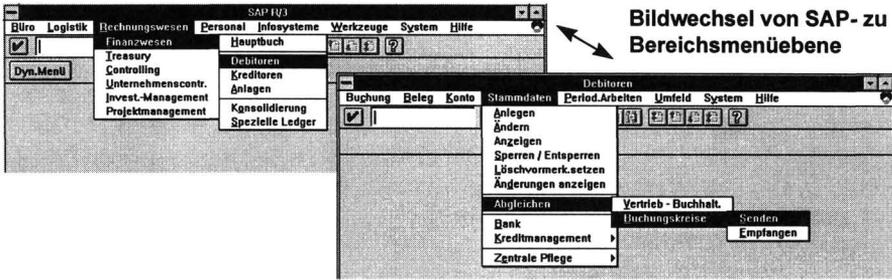


Abb. 2: Herkömmlicher Menüestieg in R/3-Anwendungen

Bei einer maximalen Zahl von ca. zehntausend (!) Anwendungs- und Berichtsvarianten war dies für den Endbenutzer nicht immer eine einfache Aufgabe, selbst wenn die Menüebenen bereits aufgabenangemessen strukturiert waren [4]. Ein durchschnittlicher Benutzer verwendet nur 10 bis 50 Anwendungen aus der Gesamtmenge, die allerdings in der Regel sehr häufig (Sachbearbeiter). Allerdings gibt es noch eine zweite Gruppe von Benutzern, die sehr viel mehr Anwendungen nutzen, diese jedoch eher unregelmäßig und seltener (Führungskräfte) [13]. Mit dem nachfolgend dargestellten Ansatz wollten wir drei Probleme lösen:

1. wie können Benutzer die von ihnen genutzten Funktionen aus einem umfangreichen Gesamtsystem auswählen,
2. wie kann das System sie dabei unterstützen, so daß nicht jeder Benutzer diese Auswahl alleine und immer wieder neu treffen muß,
3. wie können Benutzer eine einmal getroffene Auswahl individuell und schnell anpassen

4 Ein Ansatz zur modellbasierten Konfiguration

Das Spektrum der Anwender einer komplexen betriebswirtschaftlichen Software reicht vom mittelständischen Betrieb bis zum multinationalen Konzern. Ebenso sind alle Branchen vertreten, die das R/3-System einsetzen, seien es z.B. traditionelle Maschinenbauunternehmen, die Chemiebranche, die Autoindustrie, Hitech-Unternehmen, Energieversorger, Versandhandel, Medienkonzerne oder auch öffentliche Unternehmen. Entsprechend unterschiedlich sind die Nutzungsbedürfnisse der Anwender. Daraus ergibt sich die Notwendigkeit, die Standardsoftware auf die jeweiligen Bedürfnisse des Kunden einzurichten. Um dies zu ermöglichen, können die Einstelltransaktionen des Systems topdown aufgrund von einer sogenannten „Anwendungskomponentenhierarchie“ ausgewählt werden. Dieser Komponentenbaum bildet eine generalisierte betriebswirtschaftliche Organisationsstruktur ab (s. Abb. 3). Die Menge der angezeigten Anwendungen ändert sich jedoch nicht.

Die Idee für die im folgenden beschriebene neuartige Präsentationskomponente des R/3-Systems, den sogenannten „Session Manager“, ist es, diese Metadatenstruktur der Komponentenhierarchie für die Auswahl der betriebswirtschaftlichen Anwendungen auszunutzen. Die angestrebte Lösung soll die Komplexität der bisherigen Auswahl von Anwendungen für den Endbenutzer (vgl. Kapitel 3) dadurch reduzieren, daß die Gesamtmenge der R/3-Anwendungen zweifach reduziert wird:

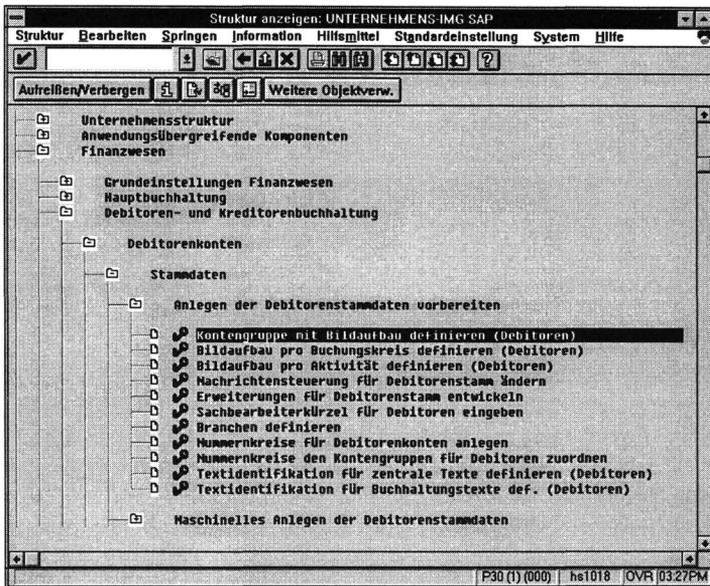


Abb. 3: Ein kleiner Ausschnitt der Anwendungskomponentenhierarchie

1. Ziel: Bereitstellung eines *Unternehmensmenüs* durch eine modellbasierte *Reduktion* der Gesamtzahl von Anwendungen im R/3-System auf diejenigen, die der Kunde tatsächlich nutzen möchte, und durch *Konstruktion* eines sichtbaren Menübaumes, der die Anwendungen tätigkeitsorientiert darstellt. Ansatz: Nutzung der Darstellungsstruktur des „alten“ R/3-Menübaumes (vgl. Abb. 2), an dem alle Anwendungen hängen, und zusätzliche Zuordnung aller R/3-Anwendungen zu den Anwendungskomponenten der Komponentenhierarchie. Bei der Systemeinführung werden diejenigen Zweige aus der Komponentenhierarchie ausgewählt, die aktiv werden sollen. Dadurch bestimmt sich die Menge der Anwendungen, die das „unternehmensspezifische Menü“ bilden. Aus der reduzierten Menge der Anwendungen wird der sichtbare Baum konstruiert.
2. Ziel: Bereitsstellung eines *Benutzermenüs* durch eine modellbasierte *Reduktion* der im Unternehmen eingesetzten Anwendungen, auf diejenigen, die ein bestimmter Endbenutzer ausführen darf und soll und ebenfalls durch *Konstruktion* eines sichtbaren Menübaumes wie unter 1. beschrieben. Ansatz: Organisationen lassen sich dadurch charakterisieren, daß Mitarbeiter auf ihnen zugeordneten Stellen bestimmte Aufgaben ausführen. Die Idee ist, im R/3-System *Stellen* mit zugehörigen *Aufgaben* zu definieren, d.h. einer Stelle R/3-Anwendungen zuzuordnen und diese einem Stelleninhaber mitzugeben. Aus der Menge der Aufgaben einer Stelle (=R/3-Anwendungen) läßt sich dann ein benutzerspezifisches Menü generieren.

Abbildung 4 verdeutlicht den Ansatz: R/3-Anwendungen werden einmal der Komponentenhierarchie zugeordnet und zum anderen vordefinierten Stellen. Aufgrund der Auswahl bestimmter Bereiche aus der Komponentenhierarchie können die benötigten Anwendungen bestimmt werden, und mit Hilfe der Struktur des alten Menübaumes kann das Unternehmensmenü konstruiert werden. Mit Hilfe einer Stelle werden die Anwendungen pro Benutzer bestimmt, und daraus wird aufgrund der Menübaumstruktur das Benutzermenü konstruiert.

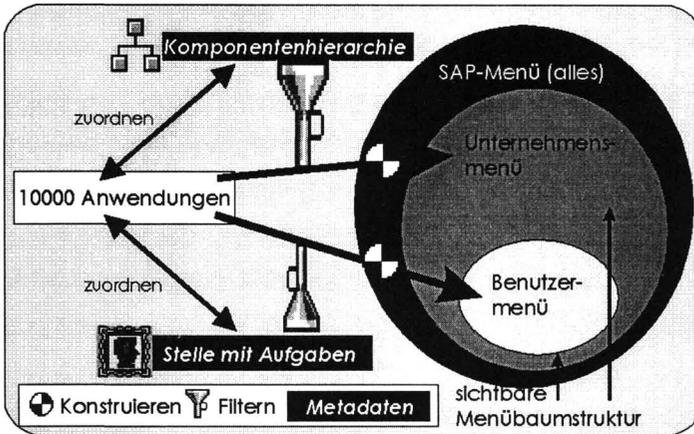


Abb. 4: Struktur zur modellbasierten Konfiguration

Das folgende Kapitel erläutert die von uns gewählte Lösung im Detail.

5 Die Konstruktionskomponenten des Session Managers

5.1 Die sichtbare Menübaumstruktur und die zugrundeliegende Struktur

Die Komponentenhierarchie bildet eine betriebswirtschaftliche Gliederung eines Unternehmens. Der Endbenutzer dagegen muß bestimmte Aufgaben an seinem Arbeitsplatz bearbeiten, die eine zeitliche aber auch inhaltliche Gliederung besitzen. Welche Aufgaben der Endbenutzer bearbeiten darf, hängt zusätzlich von seiner Stellung im Unternehmen ab. Die Struktur des eigentlichen Menübaums mit allen Anwendungen, den der Endbenutzer sieht, soll aber seinen Arbeitsbereich möglichst adäquat abbilden. Diese Information liegt im System bereits in Form der „alten“ Menübäume vor. Die Menübäume wurden so definiert, daß sie möglichst ganzheitlich bestimmte Arbeitsbereiche (Bereichsmenü) abbilden [4], d.h. die Menübäume eignen sich zur Darstellung des „unternehmensspezifischen“ als auch „benutzerspezifischen“ Menüs. Die Verbindung zwischen den sichtbaren Menübäumen und der Komponentenhierarchie wurde von uns durch Zuordnung aller Anwendungen zur Komponentenhierarchie geschaffen. Damit sind alle Daten abgelegt, um aus einer vom Kunden reduzierten Komponentenhierarchie ein sichtbares Unternehmensmenü zu erzeugen.

5.2 Stellen, Aufgaben und Tätigkeitsklassen

Für die Erzeugung des Benutzermenüs sind weitere Daten im System abzulegen. Dabei muß der Prozeß der Definition von Stellen mit Aufgaben sinnvoll unterstützt werden. Eine einfache Auswahl von Anwendungen aus der Gesamtmenge von 10000 Anwendungen würde bei dem Stellendefinierer voraussetzen, daß er genau weiß, welche Anwendungen für welche Stelle benötigt werden. Man kann nicht davon ausgehen, daß dieses Wissen vorhanden ist. Was allerdings vorhanden ist, ist ein Wissen über die Struktur der Komponentenhierarchie und über die hierarchische Position der zu definierenden Stelle innerhalb der Organisation,

z.B. Führungskraft. Mit Hilfe von Anwendungskomponenten der Komponentenhierarchie läßt sich eine Stelle bzgl. ihrer organisatorischen Einbettung beschreiben, z.B. Debitorenbuchhaltung. Durch Auswahl der hierarchischen Position innerhalb der Organisation dagegen kann das Spektrum der Aufgaben beschrieben werden. Mit diesen beiden Dimensionen lassen sich die Aufgaben einer Stelle genau beschreiben. Zur Festlegung der hierarchischen Position ist eine Zuordnung der Anwendungen zu sogenannten „*Tätigkeitsklassen*“ erforderlich.

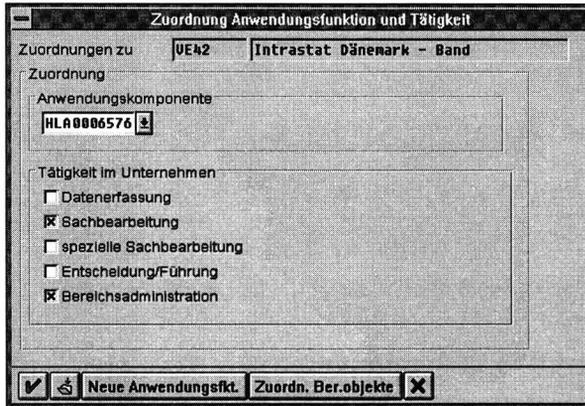


Abb. 5: Definition von Tätigkeitsklassen

Fünf Tätigkeitsklassen reichen nach unserem Verständnis dazu aus (vgl. Abb. 5). Alle Anwendungen wurden gezielt diesen Tätigkeitsklassen zugeordnet. Beide Dimensionen sind zueinander orthogonal. Mit dem gewählten Modell ist z.B. die Situation abbildbar, daß ein Sachbearbeiter im Bereich Controlling anspruchsvollere Anwendungen ausführen kann als ein Sachbearbeiter im Rechnungswesen.

5.3 Filterung und Konstruktion

Durch die Zuordnung aller Anwendungen zu Anwendungskomponenten und Tätigkeitsklassen können sowohl das Unternehmensmenü und aufgrund einer Stellenbeschreibung das Benutzermenü automatisch erzeugt werden. Dazu wird ein zweistufiger Prozeß benutzt: ein *Analyseprozeß* und ein nachfolgender *Konstruktionsprozeß*.

Der Analyseprozeß für das Unternehmensmenü: der Kunde wählt aus, welche Zweige der Komponentenhierarchie von ihm gewünscht sind. Anhand der Tabelle, die die Zuordnungen Anwendungen zu Anwendungskomponenten enthält, werden diejenigen Anwendungen vermerkt, die den ausgewählten Anwendungskomponenten zugeordnet wurden.

Der Analyseprozeß für das Benutzermenü: in einer Stellenbeschreibung werden Tupel aus Anwendungskomponenten und Tätigkeitsklassen festgelegt. Aufgrund dieser Tupel werden anhand der Zuordnungstabelle die Anwendungen vermerkt, die den entsprechenden Tupeln entsprechen.

Der Konstruktionsprozeß: aufgrund der übriggebliebenen Anwendungen werden alle die Zweige aus dem zur Präsentation anstehenden Gesamtmenübaum eliminiert, die an ihren

Blättern keine Anwendungen mehr enthalten. Es entsteht ein Menübaum, der nur noch die gewünschten Anwendungen enthält.

Der entstandene Menübaum, sei es das Unternehmensmenü oder das Benutzermenü, wird von der Präsentationskomponente automatisch angefordert und auf dem PC gepuffert.

6 Die Präsentationskomponente des Session Managers

Der Session Manager präsentiert sich dem Endbenutzer als eigenes paralleles Fenster. Wenn man sich am R/3-System anmelden möchte, startet man den Session Manager und gibt Benutzernamen, Kennwort, Mandant und Sprache an und wählt eine Systemkennung aus.

Nach erfolgter Anmeldung kommt jeweils pro Systemanmeldung ein eigenes Register hoch, welches die eigentlichen Komponenten zur zentralen Steuerung der Anwendungen enthält (vgl. Abb. 6).

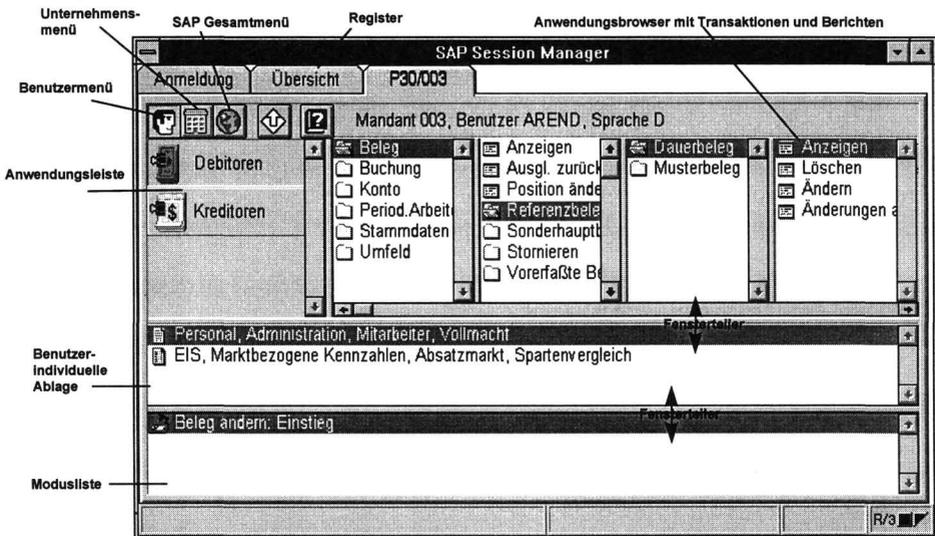


Abb. 6: Aufbau der Systemregister des Session Managers

Die oberste Reihe wird durch die Systemregister gebildet. Hier kann der Benutzer zwischen Anmeldebild, Übersichtsbild und den einzelnen Systemen wechseln. Darunter erscheint eine Kontrolleiste mit den Funktionen zur Auswahl einer bestimmten Sicht auf die R/3-Anwendungen (Benutzermenü, Unternehmensmenü und SAP-Gesamtmenü), der Abmeldenfunktion und der Funktion zum Aufruf der Hilfe zum Session Manager. Die Anwendungsleiste enthält die Einstiegsbereiche in das R/3-System. Im Falle des SAP-Menüs sind dies die Bereiche „Finanzwesen“, „Personal“, „Informationssysteme-Katalog“, „Logistik“, „Büroanwendungen“ und „Werkzeuge“ (Workbench, Administration). Der Anwendungsbrowser neben der Anwendungsleiste enthält dann die jeweiligen Äste der Anwendungen zu dem gewählten Anwendungsbereich.

Ein Beispiel möge die Funktionsweise verdeutlichen: wählt man das Gebäude-Icon aus der Kontrolleiste aus, wird das aufgrund des Einführungsprozesses erzeugte Unternehmensmenü geladen und angezeigt. Es erscheinen in der Liste links und im Browser nur noch die Äste, die aufgrund der Auswahl von Komponenten übriggeblieben sind. Der Benutzer klickt in der Anwendungsleiste einen Bereich an. Von links nach rechts wählt er sich bis zur gewünschten Anwendung und startet diese durch Doppelklick mit der Maus oder durch Ziehen in den Bereich der Modusliste. Die browserartige Darstellung eines Baums hat gegenüber der baumartigen Darstellung im Microsoft Explorer den Vorteil, daß der jeweilige Kontext zu einer getroffenen Wahl auf der nächsthöheren Hierarchieebene des Baumes erhalten bleibt.

Der mittlere Bereich des Session Managers beinhaltet die „benutzerspezifische Ablage“. Hierin kann der Benutzer mittels Drag und Drop häufig genutzte Anwendungen hineinziehen. Die Anwendungen werden genauso gestartet wie im Browser. Der untere Bereich, die „Modusliste“, zeigt die aktiven Anwendungen. Mit Hilfe der Fensterteiler lassen sich die Bereiche Browser, Ablage und Modusliste in der Größe den Wünschen des Benutzers anpassen.

6.1 Das benutzerspezifische Menü

Das Benutzermenü wird geladen, wenn man auf das Icon mit dem „Kopf“ klickt. Links erscheinen in der Anwendungsleiste als Einstieg nun die aufgrund der Stellendefinition erzeugten Bereichsmenüs (vgl. Abb. 6). Wählt man eines aus, wie in der Abbildung „Debitoren“, erscheinen rechts im Anwendungsbrowser die zugehörigen Hierarchiestufen. Man klickt sich von links nach rechts durch und wählt eine Anwendung aus.

Der entscheidende Vorteil des Benutzermenüs ist:

- der Baum enthält nur die Anwendungen, die dem jeweiligen Benutzer zugeordnet wurden und für die er auch eine Berechtigung hat
- die Zahl der Hierarchiestufen ist klein, da bereits auf Bereichsmenüebene eingestiegen wird (Breite vor Tiefe)
- das Benutzermenü ist PC-unabhängig.

7 Resümee und geplante Weiterentwicklungen

Grundlage zur Entwicklung des Session Managers waren zwei Forderungen:

- die Konfiguration eines komplexen Systems soll transparent und topdown durch den Kunden erfolgen und zwar auf Grundlage des Arbeitsablaufs bei der Einrichtung [10]
- der Konfigurationsprozeß soll durch die Nutzung von Metadaten unterstützt werden.

Die erste Forderung wurde umgesetzt, indem zum einen eine Sicht der Organisation geschaffen wurde (betriebliche Komponentenhierarchie und Stellenbeschreibungen) und zum anderen davon losgelöst eine Sicht des Benutzers auf das System (Benutzermenü und Unternehmensmenü im Session Manager). Die zweite Forderung wurde umgesetzt, indem alle Anwendungen durch Experten (SAP-Berater mit guten Praxiskenntnissen) klassifiziert wurden, d.h. Anwendungskomponenten und Tätigkeitsklassen zugeordnet wurden.

Der Nutzen ergibt sich für alle Beteiligten: das Projektteam, daß ein R/3-System einführt, kann transparent alle Komponenten abschalten, die nicht eingesetzt werden. Die Bereichsad-

ministratoren können mit einem Minimum an Aufwand Stellen für ihre Benutzer definieren und haben dabei gleichzeitig eine Kontrolle darüber, was der Benutzer soll und darf. Der Endbenutzer selbst enthält genau die Anwendungen in einem stark reduzierten Menübaum, die für ihn relevant sind. Mit einer einfach strukturierten Benutzungsoberfläche erhält er nicht nur die Kontrolle über gestartete Prozesse, sondern kann noch weitere Individualisierungen durch einfaches Drag und Drop vornehmen. Damit ist eine sinnvolle Kombination von direkter Manipulation und modellbasierter Steuerung gefunden worden. Die Resonanz von Beratern und Nutzern ist entsprechend positiv.

Aufwand entsteht bei dem gewählten Ansatz an einer Stelle - darauf soll abschließend hingewiesen werden: die erzeugten Metadaten müssen ständig aktuell gehalten werden (wenn z.B. neue Anwendungen ergänzt werden) und die Klassifikation einer Anwendung muß valide sein. Um hierbei auf der sicheren Seite zu sein, wird bei dem zur Zeit ausgelieferten Session Manager auf die Auswertung der Zuordnung zu Tätigkeitsklassen verzichtet, um von den Kunden Daten über den Prozeß der Stellendefinition zu sammeln. Wenn Erfahrungen vorliegen, kann eine entsprechende Erweiterung vorgenommen werden.

Naheliegend ist auch, daß vom Session Manager nicht nur SAP-Anwendungen aufgerufen werden können, sondern auch beliebige andere Desktop-Applikationen. Dem Benutzer stünde dann ein vollständig integrierter Desktop zur Verfügung. Entsprechende Anforderungen von SAP-Kunden liegen bereits vor.

Literaturverzeichnis

- [1] Apple Computer. Macintosh Human Interface Guidelines. Addison-Wesley, Reading, MA, 1992.
- [2] D. Benyon, T. Kühme, U. Malinowski, S. Piyawadee: Computer-aided adaptation of user interfaces. An INTERCHI '93 Workshop Report, SIGCHI Bulletin, Vol. 26, Jan. 1994, p. 25.
- [3] A.E. Blandford: An agent-theoretic approach to computer participation in dialogue. Int. J. Man-Machine Studies, 39, 1993, pp. 965-998.
- [4] E. Eberleh, F. Meinke: OASE: Eine Arbeitsplatzumgebung für komplexe Anwendungssysteme. In: H.D. Böcker (Hg.): Software-Ergonomie '95. Teubner, Stuttgart, 1995, 125-141.
- [5] DIN 66234, Teil 8: Bildschirmarbeitsplätze - Grundsätze ergonomischer Dialoggestaltung, Berlin, Beuth, 1988.
- [6] T. Greutmann, D. Ackermann: Zielkonflikte bei Software-Gestaltungskriterien. In: S. Maaß, H. Oberquelle (Hg.): Software-Ergonomie '89. Teubner, Stuttgart, 1989, 144-152.
- [7] Grundsätze für die Prüfung der Arbeitssicherheit von Elektronischen Datenverarbeitungs- und Bildschirmgeräten, Entwurf 04/96. Fachausschuß Verwaltung der Berufsgenossenschaften, Hamburg, 1996.
- [8] W. Hacker: Arbeits- und organisationspsychologische Grundlagen der Software-Ergonomie. In: E. Eberleh et al. (Hg.): Einführung in die Software-Ergonomie. DeGruyter, Berlin, 1994, 53-94.
- [9] G. Hellbardt: Die Ethik von Agenten. In: Informatik-Spektrum, 19, 1996, 87-90.
- [10] ISO 13407. Human centred design processes for interactive systems. Committee Draft International Standard, 1996-02.
- [11] C. Karger, R. Oppermann: Empirische Nutzungsuntersuchungen adaptierbarer Schnittstelleneigenschaften. In: D. Ackermann & E. Ulich (Hg.): Software-Ergonomie '91. Teubner, Stuttgart, 1991, 273-280.
- [12] P. Maes, A. Wexelblatt: Interface Agents. Workshop at CHI '96 in Vancouver, BC, Canada, April 13-18, 1996.
- [13] C. Neugebauer: Arbeitsorganisation & mentale Repräsentation betriebswirtschaftlicher Software. - Wissensstrukturen als Grundlage benutzerorientierten Interface-Designs. Unveröffentlichte Diplomarbeit, RWTH Aachen, 1991.

- [14] J. Nielsen: Enhancing the explanatory power of usability heuristics. In: Proceedings of the CHI, 1994 (Boston, MA, USA, April 24-28, 1996) ACM, New York, 1994, pp. 152-158.
- [15] R. Oppermann: Individualisierung von Benutzungsschnittstellen. In: E. Eberleh et al. (Hg.): Einführung in die Software-Ergonomie. DeGruyter, Berlin, 1994, 235-269.
- [16] R. Oppermann, C. Thomas: Auf dem Weg zur benutzerspezifischen Systemanpassung. In: W. Coy, P. Gorny, I. Koop, C. Skarpelis (Hg.): Menschengerichte Software als Wettbewerbsfaktor. Teubner, Stuttgart, 1993, 442-462.
- [17] M.E.P. Seligman: Erlernte Hilflosigkeit. München, Urban & Schwarzenberg, 1983.
- [18] K. Sullivan: The Windows 95 Interface: A Case Study in Usability Engineering. In: Proceedings of the CHI '96 (Vancouver, BC, Canada, April 13-18, 1996) ACM, New York, 1996, pp. 473-480.
- [19] E. Ulich: Von der Benutzungsoberfläche zur Arbeitsgestaltung. In: K.H. Rödiger (Hg.): Software-Ergonomie '93. Teubner, Stuttgart, 1993, 19-30.

Adressen der Autoren

Dr. Udo Arend
Usability Group
SAP AG
Neurottstr. 16
69190 Walldorf/Baden
Email: udo.arend@sap-ag.de

Dr. Edmund Eberleh
Usability Group
SAP AG
Neurottstr. 16
69190 Walldorf/Baden
Email: edmund.eberleh@sap-ag.de