

**RWTH Aachen University
Media Computing Group
Prof. Dr. Jan Borchers**

**Mensch-Maschine-Interaktion
SS2007**

User Support

Han Xiao, Bin Tan

07.2007

Tutor: Daniel Spelmezan

Inhaltsverzeichnis

1	Einleitung	3
2	Anforderung von User Support	3
2.1	Verwendbarkeit	4
2.2	Genauigkeit und Ganzheit.....	4
2.3	Vereinbarkeit.....	4
2.4	Robustheit	4
2.5	Flexibilität	5
2.6	Unimperativkeit.....	5
3	Arten von User Support.....	5
3.1	Befehl-Assistenz.....	5
3.2	Befehl-Hinweis.....	6
3.3	Die kontextsensitive Hilfe.....	6
3.4	Online-Anleitung.....	6
3.5	Online-Dokumente	7
3.6	Guide und Assistant.....	7
4	Adaptive Hilfe-Systeme (Adaptive Help Systems).....	8
4.1	Repräsentation der Kenntnisse: Modellierung von Benutzer	8
4.2	Repräsentation der Kenntnisse: Modellierung von Domänen und Aufgaben ..	9
4.3	Repräsentation der Kenntnisse: beratende Strategie der Modellierung	10
4.4	Techniken für Repräsentation der Kenntnisse	10
4.5	Probleme mit Repräsentation der Kenntnisse und Modellierung	11
4.6	Initiative, Wirkung und Bereich.....	12
5	Design von User-Support-System	12
5.1	Präsentationen von Hilfe.....	12
5.2	Ausführen von Hilfe	13
6	Fazit.....	14
	Literatur	15

1 Einleitung

Es gibt immer eine implizite Annahme, wenn man ein richtiges Interaktionssystem entwerfen kann, so dass der Gebrauch des Systems vollständig intuitiv ist, dann braucht der Benutzer sehr wenige oder keine Hilfe oder Training. Dieser Gedanke ist sehr gut, aber kann jetzt das beste System – diese Ziel auch nicht erreichen. Der Benutzer braucht verschiedene Hilfe in der verschiedenen Zeit, man möchte diese Hilfe-Funktion entwerfen.

Die Hilfe, die der Benutzer braucht, besteht aus vier Haupttypen:

- schnelle Hinweisse
- Hilfe für die spezielle Aufgabe
- volle Erklärung
- Demonstration

Schnelle Hinweisse kann die Erklärung von einem Werkzeug, das der Benutzer im allgemeinen versteht und schon benutzt hat, dem Benutzer aufweisen. Hilfe für die spezielle Aufgabe kann man helfen, wenn der Benutzer in der Arbeit von einer speziellen Aufgabe auf ein Problem getroffen hat, oder weiß nicht, wie kann man ein Werkzeug in einer speziellen Aufgabe benutzen. Diese Hilfe ist direkt mit der Aufgabe bezüglich. Einige erfahrene oder neugierige Benutzer brauchen volle Erklärung von Werkzeugen und Befehle, um vollständiger zu verstehen. Der vierte Support-Typ ist Demonstration. Dieser Typ für den neuen Benutzer von einem Werkzeug ist besonders brauchbar, der erklärt Schritt für Schritt, wie kann man den Werkzeug benutzen.

Der Benutzer in verschiedener Zeit von einem System braucht verschiedene Hilfe, das bedeutet, dass man verschiedene Hilfe-Funktionen braucht. In diesem Text können wir viele verschiedene User-Support-Systeme verstehen, und feststellen, wie kann man ein gutes User-Support-System entwerfen.

Zwischen Hilfe-Funktion und Dokument gibt es im allgemeinen Unterschied. Die Hilfe-Funktion ist auf das Problem orientiert und spezifisch. Das Dokument ist auf das System orientiert und allgemein. Die gleichen Prinzipien treffen auf die beiden zu, deshalb sollten wir die beiden nicht trennen. Zwar ist die meiste Inhalte, die wir in diesem Text lernen, für das Papierdokument und die Anleitung nützlich, aber diskutieren wir hauptsächlich Online-Support. Bevor wir die verschiedene Methode vom User-Support lernen, haben wir zunächst verstanden, welche Anforderung braucht die ideale Hilfe-Funktion.

2 Anforderung von User Support

Welche Hilfe-Funktion ist ideal? Diese Frage ist schwer. Aber wir können zuerst

wissen, welches wichtige Attribut hat die ideale Hilfe-Funktion.

2.1 Verwendbarkeit

Während der Benutzer mit dem System interagiert, kann er irgendwann die Hilfe bekommen. Das Hilfe-Programm sollte am besten mit anderen beliebigen Programmen gleichzeitig laufen. Wenn das Hilfe-Programm vom laufenden Programm nicht abhängt, gibt es in dem kein Fenster-System Problem. Aber im Fenster-System sollte man immer sofort das Hilfe-Werkzeug durch eine Taste bekommen.

2.2 Genauigkeit und Ganzheit

Die Hilfe muss genau und komplett, aber während die Programme häufig aktualisiert werden und die unterschiedlichen Versionen gleichzeitig benutzt werden, ist das Problem nicht leicht. Wenn die bereitgestellte Hilfe mit dem tatsächlichen Verhalten der Funktion nicht gleich ist, kann der Benutzer keine richtige oder genaue Lösung des Problems bekommen. Die Ganzheit ist auch sehr wichtig. Wenn die Hilfe nicht komplett ist, u.U. bekommt der Benutzer für das interessante Thema keine Hilfe. Bei diesen Situationen ist der Benutzer bestimmt unbefriedigt.

2.3 Vereinbarkeit

Der Benutzer benötigt unterschiedliche Hilfe für unterschiedliche Zwecke. Das bedeutet, dass ein Hilfe-System aus viele Teile besteht. Alle Hilfe, die jede Teile zur Verfügung stellt, müssen gleich sind. Online-Hilfe sollte mit Papierdokument gleich bleibend auch sein. Und im Programm gibt es auch User Support, diese sollte über dem ganzen System gleich bleibend sein. Wenn nicht, ist diese Hilfe-Funktion unnützlich.

2.4 Robustheit

Das ist sehr wichtig, dass die Hilfe-Funktion robust ist. Während das System sich unerwartet benimmt oder gänzlich ausgefallen ist, sollte die Hilfe-Funktion mit der korrekten Fehlerbehandlung und dem vorhersagbaren Verhalten versorgen. Wenn der

Benutzer braucht, kann er Hilfe bekommen. Deshalb ist Robustheit wichtiger als andere Attribute.

2.5 Flexibilität

Viele Hilfe-Funktionen sind sehr steif, dass sie immer die gleichen Hilfeinformationen geben und unabhängig davon sind, ob der Benutzer die Fachkenntnisse hat, oder in welchem Kontext arbeiten sie. Eine flexible Hilfe-Funktion erlaubt dem Benutzer, dass er mit dem System in einer Weise interagieren kann, die zu seinen Notwendigkeiten passt.

2.6 Unimperativkeit

Das letzte Prinzip vom Entwurf der Hilfe-Funktion ist die Unimperativkeit. Die Hilfe-Funktion sollte die Arbeit und auch das Programm vom Benutzer nicht verhindern. Und eine Hilfe-Funktion soll nicht selbst aktiv die Hilfe geben, sondern auf die Bitten vom Benutzer.

3 Arten von User Support

In diesem Abschnitt konzentrieren wir uns auf die Arten der bereitgestellten Hilfe anstatt jede spezifische Hilfe-Funktion. Dann setzen wir die adaptive Hilfe ausführlicher auseinander.

3.1 Befehl-Assistenz

Diese Art ist die grundlegende Art von User-Support. Bei UNIX man, Dos oder Windows benutzt man immer.

Wenn der Benutzer den Hinweis oder die genaue Information des Befehls versteht, ist dieser Art leicht und leistungsfähig. Jedoch, der Benutzer versteht für einige Befehle vielleicht sehr genau, aber für andere versteht vielleicht sehr wenig; oder weißt er einige Befehle nicht, aber braucht er benutzen; und einige Befehle denkt er bestehend, aber ist keine. Für diese Situationen kann Befehl-Assistenz dem Benutzer sehr wenig oder nicht helfen.

3.2 Befehl-Hinweis

Wenn der Benutzer in der (command-line-Benutzeroberfläche?) eine Störung trifft, hilft Befehl-Hinweis ihm, normalerweise den richtigen Gebrauch zur Verfügung zu stellen. Wenn die Störung leicht ist (z.B. eine falsche Syntax), ist die Art sehr nützlich.

Andere Form von Befehl-Hinweis sind die Anwendungsmenüs und auswählbare Icons. Bei diese Art möchte der Benutzer auch die Befehle verstehen, so wird die zusätzliche Unterstützung noch angefordert.

3.3 Die kontextsensitive Hilfe

Einige Hilfe-Funktionen sind kontextsensitiv. Diese könnte sowohl dem spezifischen Benutzer das spezifische Wissen geben, als auch mit einer leichte Hilfetaste oder Funktion versorgen, die Funktion deutet aufgrund des Kontextes und liefert entsprechende Hilfe.

3.4 Online-Anleitung

Online-Anleitung erlaubt dem Benutzer, der die grundlegende Arbeit des Programms in einer (Testumgebung?) zu machen. Der Benutzer kann auf seinen Bedarf und seine Geschwindigkeit weitermachen.

Die meiste Online-Anleitung haben keine Intelligenz: sie wissen weder über den Benutzer und seine vorhergehende Erfahrung, noch über das Gebiet und die unterrichtende Art. Deshalb ist Online-Anleitung unbiegsam und unbefriedigend.

Eine andere ungleiche Idee ist "*Training Wheels*", die von Carroll und seinen Kollegen an IBM vorgeschlagen wird. "*Training Wheels*" erlaubt dem Benutzer, in einem Online-Anleitung-System zu erforschen und experimentieren, das nur begrenzte Funktionen hat. Durch diese Version kann der neue Benutzer den Betrieb vom System besser verstehen.

3.5 Online-Dokumente

Online-Dokumente kann auf Computer die vorhandenen Papierdokumenten effektiv liefern. Und die Materialien können im laufenden Medium vom Benutzer fortwährend bekommen werden und von vielen Benutzern erhalten werden. Die Papierdokumente können durch Index, Inhaltsverzeichnis, Seitennummer oder andere Anhalte dem Benutzer helfen, besser durchzusehen. Diesen Attribute haben die meiste Dokumentsysteme keine. Aber die Papierhandbücher sind leicht verloren, können nur auf einem physikalischen Platz gelegt sein. Online-Dokumente können diese Probleme lösen.

Durch die Hypertextkenntnis kann der unerfahrene Benutzer Online-Dokumente leicht durchsehen. Die Kenntnis speichert Texte im Netzwerk, und verbindet alle Teile von Texte durch die auswählbare Verbindung.

Jedoch haben die Hypertext-Systeme ein gleiches Problem-----Navigation. Der Benutzer könnte im Hypertext verirren, das bedeutet, dass er noch lange Zeit verbringt, wieder zu suchen.

Ein minimales Handbuch, in dem nur die Grundgehalte stehen bleiben, kann dem Benutzer helfen, das System schneller zu verstehen.

3.6 Guide und Assistant

Guide ist ein die spezifische Aufgabe ausfüllendes Werkzeug, es kann den Benutzer Schritt für Schritt führen, die Aufgabe zu erledigen. In diesem Verlauf braucht Guide die Informationen der Antworten von dem Benutzer verwenden. Wenn der Benutzer einen formatierten Lebenslauf bekommen möchte, möchte er eine Reihe Fragen, die der Microsoft Word Lebenslauf-Guide fragt, beantworten und einige grundlegende persönliche Daten eingeben. Guide verursacht dann aufgrund der eingereichten Daten einen Lebenslauf.

Der Benutzer kann durch die Anwendung von Guide die komplexe Aufgabe sicher, schnell und genau ausführen. Für die seltene Aufgabe ist Guide besonders nützlich. Jedoch könnte Guide mit den Wahlen, die der Benutzer wollte, nicht versorgen oder um die Information, die der Benutzer keine hat, bitten. Ein guter Guide kann dem Benutzer erlauben, einen Schritt sowie vorwärts zurückzuschieben.

Ein anderes neues Werkzeug von User Support ist Assistant. Der überwacht das Verhalten des Benutzers, und gibt die Vorschläge oder Tips.

Ein unterlegenes Beispiel ist "Clippy", ein Assistant von Microsoft Office 97. Warum ist "Clippy" unterlegen? Ein wichtiges Attribut von Hilfe-Funktion ist unimperativ. Jedoch "Clippy" hat eine lange Liste der Vorschläge und ein unaufhaltsamer Animationsfilm. Zusätzlich sind seine Vorschläge häufig nicht angebracht. Also ist dieser Entwurf ganz unterlegen.

“Clippy” stellt eine wichtige Lektion für User Support zur Verfügung. Die Hilfe-Funktion muss unter der Steuerung des Benutzers sein, und diese Hilfe soll unimperativ sein. Eine vertretende Funktion von Microsoft XP ist “smart tags”, die erlernt schon die Lektion.

4 Adaptive Hilfe-Systeme (Adaptive Help Systems)

Es gibt viele verschiedenen Funktionen und komplexen Anwendungen in einem großen Computer System. Dabei ist es möglich, dass nicht alle Benutzer genaue Erfahrungen mit den Funktionen besitzen und die Benutzer haben auch verschiedenen Bedarf. Adaptive Hilfe Systeme versuchen dieses Problem dadurch zu beheben, dass sich verschiedenen Bedarfs von Benutzer anpassen und ihnen entsprechenden Hilfen geben.

Wie ein intelligentes System, adaptive Hilfe System braucht hauptsächlich Kenntnisse Repräsentation (knowledge representation). Diese Kenntnisse, die adaptive Hilfe System benutzte, schließen ein Modell von eines Benutzers Kenntnissen, Vorlieben, und Fehlern ein. Mit den Kenntnissen von der Domäne, indem der Benutzer arbeitet, kann adaptive Hilfe System die Hilfe geben, die der aktuellen Aufgabe von dem Benutzer entsprechen.

Aber es ist tatsächlich schwierig, um ein adaptive Hilfe System zu konstruieren. Das Problem steht auf Kenntnisse Repräsentation, die Wirkung vom Modell, die Initiative und der Bereich vom System.

4.1 Repräsentation der Kenntnisse: Modellierung von Benutzer

Ein wichtiges und auch unerlässliches Modell von einem adaptiven Hilfe System ist ein Modell des Benutzers. Ein Benutzer Modell kann einfach als einziger typischer Benutzer sein, wenn alle Benutzer genaue Erfordernisse haben.

Aber bei vielen Systemen, die sich *anpassungsfähig(adaptable)* nennen, das Benutzer Modell wird erlauben, dass die Benutzer selbst einige Attribute vom System benutzerdefiniert. Aber die Anfänger können nicht gut benutzerdefinieren weil sie noch nicht wissen, wie diese Benutzerdefinieren zu machen.

Es gibt auch ein anderes adaptives Modell, nicht wie das Benutzerdefinieren Modell, das die Verantwortung von dem Benutzerdefinieren übernimmt und sich auch verändern können lässt. Mit die Informationen von die Überwachung von der Benutzers Interaktion kann das System nach dem Zeitpunkt eines Standard Benutzer Modell darstellen, und dann diese Modell für den aktuellen Benutzer aktualisieren.

Benutzer Modelle könne durch drei wegen konstruieren werden: Quantelung, Stereotyp und Overlay Modell.

Quantelung

In Quantelung-Modell-System verteilen sich die Sachverständigen Begutachtungen von Benutzer über einige verschiedene Stufen. Der Benutzer steht auf einer dieser Stufe, die ist beweglich und abhängig von der Quantelung von seiner Sachverständigen Begutachtung. Diese Sachverständigen Begutachtung können die Tätigkeiten von Benutzer, z.B. die Zahl des Benutzens von einem Befehl, oder die anderen Attribute vom Benutzer sein. Dieses Modell kann nur ungefähr die Sachverständigen Begutachtung eines Benutzens zeigen, trotzdem es ein einfaches und schnelles System ist und wenige Information von dem System Protokoll braucht.

Stereotyp

Ein anderes Modell nennt sich Stereotyp. Dabei werden einige Kategorie oder Stereotype des Benutzers vorher gebaut, dann wird der Benutzer in eine Kategorie klassifiziert. Das Stereotyp Modell ist besonderes hilfreich wenn die meisten Benutzer dargestellt werden, aber nicht zu hoch entwickelt.

Overlay Modell

Overlay-Modell ist auch häufig benutzt. In diesem System wird ein idealisiertes Modell zuerst dargestellt. Dann lässt sich dieses idealisierte Modell mit einem individuellen Benutzer vergleichen. Endlich mit dem Ergebnis, das entweder Gemeinsamkeit oder Verscheiden sind, wird das Benutzer Modell gebaut. Dieses Ergebnis kann auch durch den Vergleichen zwischen der bekannten Fehler und der Benutzer gemachten Fehler erhalten. Mit diesem Modell kann das System dem Benutzer helfen, um die Fehler zu vermeiden.

4.2 Repräsentation der Kenntnisse: Modellierung von Domänen und Aufgaben

Außer der Kenntnisse der Benutzer brauchen alle adaptive Hilfe Systeme die Kenntnisse des eigenen System, die normalerweise aus Benutzen des Befehls, gemeinsame Fehler und Aufgaben stehen. Manchmal verwenden die adaptiven Hilfe Systems auch die aktuelle Aufgabe des Benutzers. Wenn der Benutzer sich mit einer Aufgaben beschäftigt, die ein Problem lösen sollt, sind nur die an die Vorliebe des Benutzers anpassende Hilfe nicht genug. Um den Benutzer die Hilfe geben, die besonders für eine Aufgabe, muss das System einem Modell von Aufgaben des Benutzers darstellen. Dabei wird die Folge der Befehl des Benutzers verwendet. Durch die Vergleiche mit der Folge der Befehl des Benutzers zu der vorher gespeicherten Folge der Aufgabe, kann die aktuelle Aufgabe des Benutzers manchmal erkannt werden.

Aber was problematisch ist die Methode der Suchen nach dem Ziel des Benutzers manchmal nicht verfügbar oder kann nicht vertreten werden, besonderes in einer großen Domäne. Außerdem ist es nicht immer einfach, die Absicht des Benutzers zu schließen, weil der Benutzer vielleicht das Problem durch einen ungewöhnlichen Weg

behebt. Die Quelle der Informationen des Benutzers, das Protokoll des Systems, behalten auch nur begrenzte Informationen.

4.3 Repräsentation der Kenntnisse: beratende Strategie der Modellierung

Es ist schon gewiss, dass es verschiedene Wege gibt, die Hilfen für den Benutzer zu geben. Und die Kenntnisse und die Situation der Benutzer sind nicht gleich. Dabei wirken die verschiedenen Hilfen also auch nicht gleich. Deshalb haben wir das Problem, nach der Kenntnisse und dem Ziel des Benutzers, welche Hilfetypen zu geben: Erinnerung, aufgabenspezifische Hilfe, und Tutorial-Hilfe.

Beratende Strategie der Modellierung wird aber in wenigen adaptiven Hilfe-Systemen verwendet, weil die Wahlen, die Strategie geben kann, sehr begrenzt sind. Trotzdem ist es wichtig, dass die Designer immer denken, mehrere Wahlen der Hilfen für den Benutzer zu bieten.

4.4 Techniken für Repräsentation der Kenntnisse

Die häufige benutzte Technik für Repräsentation der Kenntnisse sind Rule-based Technik, Frame-based Technik, network-based Technik und Example-based Technik, die allen auch zu der künstlichen Intelligenz gehört sind.

Rule-based Technik

Kenntnisse können als Regeln und Tatsachen dargestellt werden. Deshalb ist Mechanismus der Folgerung hierbei verfügbar. Mit der Regeln der Produktion werden zu tunende Aktionen und Kenntnisse gezeigt. Wenn die Tatsachen der bekannten Kenntnisse die Zustände der Regeln füllen, werden die Regeln benutzt und dann bekommen die Kenntnisse aus Folgerung. Rule-based Technik hat eine einfache Struktur der Folgerung, die als IF-THEN Form gezeigt wird. Die Technik ist auch nützlich in großen Domänen.

IF

Tatsachen der bekannten Kenntnisse

THEN

Folgerung

Frame-based Technik

In der kleinen Domänen, z.B. Modellierung des Benutzers, wird Frame-based Technik gut verwendet. Die Struktur des Systems ist von den Rahmen, aber nicht von den Regeln wie Rule-based System. Hierbei wird ein Rahmen mit Steckplätzen von Labels dargestellt, die charakteristischen Merkmale repräsentieren, ähnlich ein Objekt zu bauen. Diese Merkmale haben verschiedene Type: numerischer Wert oder logischer Wert, etc. Wenn der Input des Benutzers gleich zu dem Wert des Rahmen, dann wird einige Aktionen getreten.

Objekt des Modells

```
{
    Merkmal #1,
    Merkmal #2,
    ...
    Merkmal #n
}
```

Network-based Technik

Die Netzstruktur kann gut benutzt, wenn es viele komplexe Zusammenhänge zwischen die Tatsachen von Benutzer und System gibt. Damit können diese Zusammenhänge von der Tatsachen als Eltern oder Kindern gezeigt. Deshalb ist diese Technik besonders an Schema oder Verbindung der Informationen gut benutzt.

Example-based Technik

Eine Struktur von Entscheidung, die ein Baum oder ein Netz sein kann, funktioniert in Example-based System. Die Folgerung bekommen nicht aus den Regeln sondern aus den Beispielen der Merkmale, häufig die Aktivitäten des Benutzers. Also braucht dieses System normalerweise nicht viele Informationen der Kenntnisse.

4.5 Probleme mit Repräsentation der Kenntnisse und Modellierung

Kenntnisse spielt eine sehr wichtige Rolle in adaptiver Hilfe System. Aber es ist manchmal schwer, sie zu halten, besonders wenn es keinen Fachmann in der Domäne gibt. Um die Kenntnisse richtig zu vertreten, eine große Menge davon werden häufig braucht. Außerdem sind die Informationen in einem laufenden System sehr schwer zu bekommen. Und diese aus Systems Logs Informationen können die Kenntnisse auch nur ungefähr darstellen.

4.6 Initiative, Wirkung und Bereich

Außer der Repräsentation der Kenntnisse gibt es auch andere Punkte, über dem wir beim Design von adaptiver Hilfe System nachdenken sollen.

Zuerst ist die Initiative des Systems. Dabei sollen wir darüber nachdenken, ob der Benutzer die vollständige Kontrolle von System haben. Die Aktivität von System beeinflusst die Interaktion und auch den Benutzer. Zu häufige Mahnung und überflüssige Hilfe können Benutzer gegen die Hilfen des Systems Abneigung empfinden lassen. Manchmal ist Initiative gemischt: System sollt die ganze Zeit dem Benutzer Hilfe geben können und die Hilfe sollt die Aktion des Benutzers nicht verhindern.

Ein anderer ist die Wirkung von der Modellierung und der Adaptation. Was sollt adaptiert werden und wie sollt das gemacht werden ist das Problem. Bei Modellierung benutzt man nur es, dass was sollt braucht werden.

Der letzte Punk, der kann nicht ignoriert werden, ist der Bereich der Hilfe. Man muss sich darüber entscheiden, ob die Hilfe nur für eine Anwendung oder für ganz System gebietet ist. Natürlich ist die Letztere viel mehr komplex.

5 Design von User-Support-System

Es gibt zwei wichtige Punkte, die sollt in dem Design von Benutzer Stütze System beobachtet werden. Zuerst, das Hilfe System sollt nicht als ein Add-on im ganz System sein, sonder mit anderen Systems gleichzeitig entworfen werden, so dass die Hilfe System relevant ist und mit allen übrigen Systems übereinstimmt. Zweite, der Gehalt und der Zusammenhang von Hilfe sind wichtiger als die Technologie, die in der Hilfe System genützt werden sollt. Beim Design sollt es zuerst über den früher nachdenken.

5.1 Präsentationen von Hilfe

Anfordern von Hilfe

Es gibt verschiedene Methode, dadurch das System den Benutzer die Hilfe bieten kann.

	Starke Punkte	Schwache Punkte
Befehl	übereinstimmt zu übrigen Benutzeroberfläche	Thema Input, Kenntnisse Bedarf
Button	einfach, selbstständig	begrenzte Information

Funktion	flexibel	
Verwendung	flexibel, mehrfache Hilfe Stile	Störung zu aktuelle Verwendung

Display von Hilfe

Abhängig von der verschiedenen Systems, die Hilfe wird in einem neuen Window oder ganzem Bildschirm oder gespaltetem Bildschirm gezeigt. Nach dem Niveau der Hilfe kann die Hilfe auch in einem pop-up Box oder als ein markiertes Icon gezeigt werden.

wirksame Präsentation von Hilfe

Zu Schreiben eine gute Hilfe schließt viele Faktoren ein.

gute Hilfe	schlecht Hilfe
klare, gewohnte Sprache, übereinstimmte Terminologie, richtungweisende Sprache, übereinstimmte Format, etc.	Fachsprache, große Blöcke

5.2 Ausführen von Hilfe

Wie die Hilfe von System ausführt ist noch eine zutreffende Entscheidung.

wo	Betrieb System Command, Meta-Command, Verwendung
System Ressource	Raum von Bildschirm, Raum von Speicher, Geschwindigkeit
Struktur von Daten der Hilfe	einzige Datei, Datei Hierarchie, Datenbank

Flexibilität und Erweiterbarkeit	Hard Kopie, Benutzer selbst hinschreiben
----------------------------------	---

6 Fazit

Es gibt kein interaktives System, dessen Benutzer nie Hilfe brauchen. Um Benutzer zu helfen, können wir verschiedene Arten der Hilfe verwenden. Verschiedene Hilfen unterstützen auch verschiedene Anforderungen und verschiedene Typen der Benutzer. Also ist es wichtig, dass genau am Anfang der Design von System schon eine gut anpassende Art der Hilfe zu wählen.

Literatur

[1] Craig Marion's links to Performance Support resources
(<http://mysite.verizon.net/resnx4g7/Perform/PerfSupt.html>)

[2] Gordon Meyer's Usable Help - examining documentation and help systems for software and consumer products (<http://www.g2meyer.com/usablehelp/index.html>)

[3] opentag.com - tools and technologies used in the localization of software, on-line help and documentation (<http://www.opentag.com/index.htm>)

[4] links to help authoring tools from Columbus State Community College
(<http://www.huskiesrus.com/classes/TCO215/tco215helpauthoringtools.htm>)

[5] Help Technology Centre - resources and techniques for help systems
(<http://frogleg.mvps.org/helptechnologies/>)