

# Hilfreiche Hilfe? - Adaptive Learning-on-Demand

Jörg Desel, Dorothea Iglezakis

Lehrstuhl für Angewandte Informatik  
Katholische Universität Eichstätt-Ingolstadt  
joerg.desel@ku-eichstaett.de

**Abstract:** Das in dieser Arbeit vorgestellte Hilfesystem für webbasierte Systeme soll dem Benutzer genau die Informationen liefern, die in diesem Moment für ihn passend, hilfreich und interessant sind. Es passt die Hilfe-Inhalte an das jeweilige angenommene Wissen des Benutzers über die Funktionalitäten der Ziel-Software an. Die Annahmen über das jeweilige Wissen basieren auf Daten, die ohne Störung des Benutzers erfasst werden. Die Benutzermodellierung verwendet kognitionspsychologische Erkenntnisse. Die vorgenommene Art der Adaption erleichtert die Erstellung und die Wartung von Hilfeinhalten und entlastet die Autoren von der Aufgabe, unterschiedliche Hilfeversionen für verschiedene Benutzergruppen zu schreiben. Das Hilfesystem wurde implementiert und jeder Adaptionsschritt mit empirischen Studien evaluiert.

## 1 Einleitung

Die Einführung neuer Technologien in Unternehmen scheitert häufig an mangelnder Akzeptanz durch die Endbenutzer. Zwei Theorien aus der Sozialpsychologie prognostizieren die Akzeptanz solcher Innovationen: Das Technology Acceptance Model [Dav89] nennt als entscheidende Einflussfaktoren die wahrgenommene Nützlichkeit und die wahrgenommene Benutzerfreundlichkeit. Nach [MPC01] ist bei Software eine Dokumentation bzw. Hilfe dabei die zweitwichtigste relevante Ressource, nach der verfügbaren Zeit. Die Theorie of Planned Behavior nennt Einstellung, subjektive Norm und wahrgenommene Verhaltenskontrolle als die entscheidenden Faktoren zur Vorhersage der Nutzung. Nach [BR96] sind Wissen über die Nutzung sowie Unterstützung hochsignifikante Faktoren für wahrgenommene Verhaltenskontrolle.

Lern- und Hilfesysteme können Wissen vermitteln und dadurch sowohl die Einschätzung der Benutzbarkeit als auch den Grad an wahrgenommener Kontrolle erhöhen. Nach der ersten Akzeptanz einer Technologie verändert sich aber die Einstellung zu ihr [XL00], und bei Software verändert sich das Wissen über die Funktionalitäten der Software bei den Benutzern ständig. Um die kritischen Phasen der Softwareeinführung wirklich zu unterstützen, müssen Lern- und Hilfesysteme daher die individuellen Lernprozesse begleiten. Tatsächlich orientieren sie sich aber meist nur an einem Benutzertypus. Laut [Nie00] war dies in den 70ern der Experte, in den 80ern der durchschnittliche Benutzer, in den Anfangsphasen des WWW der Novize und seit der steigenden Bedeutung komplexer webbasierter Systeme wieder der Experte.

Als Konsequenz sind Benutzer mit derartig statischen Lern- und Hilfesystemen recht unzufrieden. Sie lassen sich lieber durch menschliche Ansprechpartner unterstützen. Dies wird in der Studie [WMO91] deutlich dokumentiert. Dieselbe Studie berichtet auch über den Wunsch nach einer verbesserten rechnergestützten Hilfe. Ein helfender Kollege kann das aktuelle Wissen eines Benutzers einschätzen und seine Hilfe entsprechend anpassen. Genau dies soll das in diesem Beitrag vorgestellte Hilfesystem ebenfalls leisten. Allerdings erhöhen sich bei einer derartigen Anpassung und den damit verbundenen Varianten von Hilfetexten der Aufwand für die Erstellung und Wartung von Hilfesystemen erheblich. Dies ist wohl die Hauptursache dafür, dass sich adaptive Hilfesysteme bislang nicht haben durchsetzen können. Das hier vorgestellte Hilfesystem soll adaptiv sein, aber ohne zusätzlichen Editier- und Wartungsaufwand auskommen.

Adaption setzt voraus, dass dem System Information über den aktuellen Wissensstand des jeweiligen Benutzers vorliegt. Diese Information könnte durch den Benutzer selbst eingegeben werden, entweder bei der Nutzung der Ziel-Software (dessen Funktionalität gelernt werden soll) oder auch erst bei Aufruf der Hilfe. Allerdings scheitert dies oft daran, dass der Benutzer bei der Arbeit mit der Zielsoftware gestört und deutlich verlangsamt wird. Die Erfassung der für die Anpassung relevanten Informationen geschieht in unserem Ansatz daher ohne aktive Beteiligung des Benutzers.

Der zweite Abschnitt beschreibt die Idee unseres Ansatzes. Der dritte Abschnitt geht auf Vorarbeiten zu adaptiven Systemen und zu Hilfesystemen kurz ein. Im vierten Abschnitt stellen wir unser Hilfesystem vor. Der fünfte Abschnitt beschreibt die Evaluation des Hilfesystems in Zusammenhang mit einer Ziel-Software. Zusammenfassung, Verallgemeinerungsmöglichkeiten und abschließende Bemerkungen schließen den Beitrag ab.

## **2 Idee und Architektur**

Die Grundidee des hier vorgestellten adaptiven Hilfesystems ist auf verschiedene Arten von Software übertragbar. Die konkrete Umsetzung konzentriert sich aber auf ein bestimmtes Anwendungsszenario. Ziel-Software sind hier webbasierte Systeme, also Programme, die auf einem Web-Server ausgeführt und von Benutzern auf einem Client-Rechner mit einem Web-Browser bedient werden. Mehrere Benutzer können gleichzeitig zugreifen. Die Systeme bieten – im Gegensatz zu Web-Seiten – eine Reihe von Funktionalitäten an. Benutzer melden sich zu einer Nutzung (einer Session) mit einer Nutzerkennung an und sind so über mehrere Sessions hinweg identifizierbar.

Das Einsatzszenario sieht wie folgt aus. Zunächst startet der Benutzer über seinen Browser die Ziel-Software und identifiziert sich auf dem Server. Dort wird eine Session gestartet. Während der Benutzung der Ziel-Software werden die Aktivitäten auf dem Server protokolliert. Fordert der Benutzer Hilfe an, ruft die Ziel-Software das adaptive Hilfesystem unter Angabe der Benutzerkennung auf. Das Hilfesystem erschließt mit Hilfe der Protokolldatei das angenommene Wissen des Benutzers, passt die Hilfeinhalte entsprechend an und liefert sie über die Ziel-Software dem Benutzer. Die Session endet durch Abmeldung des Benutzers.

Web-Server bieten stets eine Protokollierung der Seitenaufrufe an. So entsteht für die Protokollierung der Benutzeraktivitäten innerhalb der Ziel-Software kein zusätzlicher Aufwand, die Nutzungsgeschwindigkeit wird nicht verlangsamt und der Benutzer nicht

behindert. Das Protokoll beinhaltet aber naturgemäß nur Server-seitige und keine Client-seitigen Aktivitäten des Benutzers (z.B. Anklicken nicht aktiver Elemente).

Die Kernidee des Ansatzes ist, das Wissen eines Benutzers aufgrund seiner protokollierten Handlungen zu erschließen, um die Hilfeinhalte an dieses Wissen anzupassen. Konkret wird aus der Protokolldatei erschlossen, welche Funktionalität ein Benutzer zu welchem Zeitpunkt aufgerufen hat. Es wird angenommen, dass er mit jedem Aufruf einer Funktionalität sein Wissen darüber vertieft, während er mit fortschreitender Zeit Wissen über nicht verwendete Funktionalitäten wieder vergisst. Dadurch erhält jeder Benutzer zu jeder Funktionalität der Ziel-Software und zu jedem Zeitpunkt einen Wert, der sein aktuelles Wissen über diese Funktionalität beschreibt. Die Kombination all dieser Werte für alle Benutzer stellt das Benutzermodell dar. Bei Aufruf der Hilfe erfolgt die Adaption von Hilfeseiten aufgrund der Werte im Benutzermodell. Die Komponenten des Gesamtsystems sind in folgender Graphik dargestellt.

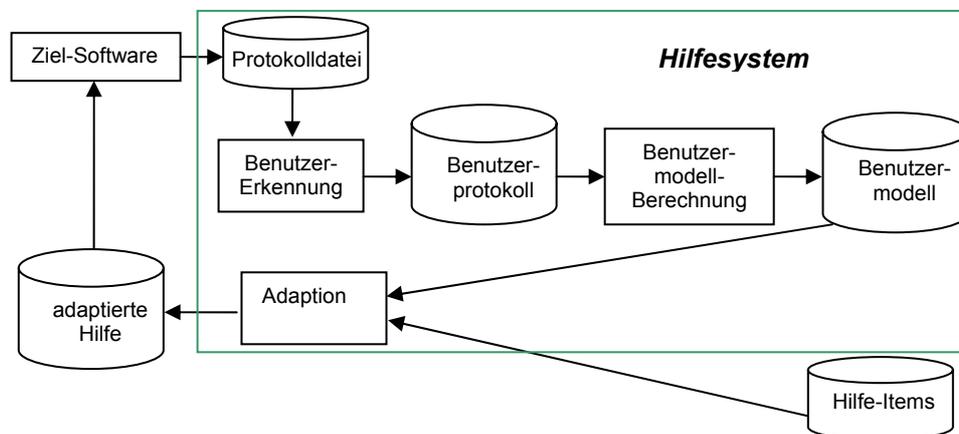


Abbildung 1: Architektur des Hilfesystems

### 3 Definitionen und Vorarbeiten

#### 3.1 Hilfesysteme

Nach [Gel98] verstehen wir unter einem Hilfesystem ein System, das sich auf eine Ziel-Software bezieht und den Benutzer bei der Ausführung seiner Funktionalitäten unterstützt. Zudem soll der Benutzer die Ziel-Software und deren Umgebung besser verstehen. Ein Hilfesystem liefert Informationen in kleinen Einheiten, den Hilfe-Items.

Zur Effektivität von Hilfesystemen findet man in der Literatur Qualitätsmaße, die im Wesentlichen angeben, wie viele Hilfe-Items zu einer Zeitersparnis bei der Benutzung der Ziel-Software führen, wenn man den Aufwand des Lesen eines Hilfe-Items mit berücksichtigt. Weitere Anforderungen an Hilfesysteme sind Korrektheit, Konsistenz Relevanz und Verständlichkeit. Für Relevanz und Verständlichkeit ist ein Bezug auf die jeweilige Nutzergruppe sinnvoll. Dies bedingt aber das mehrfache Vorliegen von Hilfe-Items zu denselben Funktionalitäten der Ziel-Software, so dass Korrektheit und

Konsistenz gefährdet sind. Hilfe-Items können in Form von Text vorliegen, aber auch Bilder und multimediale Elemente enthalten.

### **3.2 Benutzeradaptive Systeme**

Ein System kann sich nur an einen Benutzer anpassen, wenn dazu Informationen über diesen vorliegen, meist in Form von Merkmalen des Benutzers. Die Merkmale konstituieren das Benutzermodell. Das Modell wird bei der Interaktion mit dem Benutzer aktualisiert, indem Merkmalsausprägungen durch einen Inferenzmechanismus abgeleitet werden. Das Benutzermodell ist umgekehrt Grundlage für Adaptionentscheidungen. Das Ziel der Adaption ist primär eine bessere Unterstützung des Benutzers, kann aber z.B. auch zielgerichtete Werbung sein, bei der der Nutzen eher fraglich ist. Typische Anwendungsbereiche sind Lernsysteme, Vorschlagsysteme und technische Systeme.

Benutzermodelle sollen nur Informationen über den Benutzer speichern, die ableitbar und für die Adaption relevant sind. Die Informationen können kurzfristig oder langfristig gespeichert werden, sie können sich auf einzelne Benutzer oder auf Gruppen ähnlicher Benutzer beziehen. Im Initialzustand stehen noch keine Informationen zur Verfügung, bei Lern- und Hilfesystemen geht man in diesem Fall oft von einem Standardmodell aus, das mit der Zeit angepasst wird. Benutzermodelle sind oft für den Nutzer nicht unmittelbar einseh- und verstehbar, was neben juristischen Problemen Misstrauen auslösen kann. In unserem Ansatz ist das Benutzermodell anonymisiert und nicht einsehbar.

Die Datenerhebung bei der Benutzermodellierung kann von der aktiven Mitarbeit des Benutzers abhängen (z.B. Fragebögen) oder aber verborgen geschehen, durch Beobachtung des Benutzerverhaltens. Im Kontext von adaptiven Hilfesystemen gibt es Vorschläge, das Verhalten bezüglich der Ziel-Software wie auch das Verhalten bezüglich der Hilfe einzubeziehen. So könnte nach dem Aufrufen eines Hilfe-Items angenommen werden, dass der Benutzer anschließend über das dort vermittelte Wissen verfügt, wenigstens wenn er die entsprechende Seite hinreichend lange geöffnet hatte. Wegen der offensichtlichen Fehlerquellen bei diesem Ansatz berücksichtigen wir ausschließlich die tatsächliche Durchführung von Funktionalitäten der Ziel-Software.

Für die Inferenz von Merkmalsausprägungen gibt es verschiedenste Vorschläge, wie die Verwendung probabilistischer Modelle (Bayes'sche Netze, Hidden Markov Models) oder deterministischer Berechnungen. Auch bei der Adaption hat sich kein Verfahren für alle Bereiche endgültig durchsetzen können. Hier wurde insbesondere mit der Verwendung unterschiedlicher Medien, je nach Benutzermerkmalen, experimentiert.

### **3.3 Existierende benutzeradaptive Hilfesysteme**

Aus Platzgründen kann hier keine ausführliche Diskussion der in der Literatur beschriebenen Systeme erfolgen. Stattdessen sei auf [Igl04] verwiesen. Dort werden die Systeme UNIX Consultant, MetaDoc, OHRIMUS und EPIAM verglichen, die sich am Wissen des Benutzers orientieren, sowie PUSH und PLUS, die Ziele und Pläne des Benutzers berücksichtigen, sowie das Lumiere-System und das Adaptive Intelligente Hilfesystem (AIHS), in denen sowohl langfristiges Wissen als auch kurzfristige Pläne modelliert werden.

### 3.4 ACT-Theorie

Die ACT-Theorie (Atomic Components of Thought) [AL98, And02] ist eine umfassende kognitionspsychologische Theorie, die menschliche Kognition quantitativ modelliert. Umfangreiche empirische Ergebnisse stützen die ACT-Theorie und entwickeln sie weiter. Sie geht davon aus, dass die menschliche Kognition aus einfachen Bestandteilen besteht, die entweder dem Faktenwissen oder dem prozeduralen Wissen (Regeln) zugeordnet werden können. Diese Bestandteile werden durch Beobachtung gewonnen und abstrahiert. Durch Wiederholung werden Spuren im Gedächtnis aktiviert bzw. verstärkt. Wissensbestandteile sind miteinander vernetzt, und so verbundene Bestandteile werden ebenfalls aktiviert. Ab einer gewissen Aktivierungsschwelle ist das zugehörige Wissen bewusst abrufbar. Häufige Verwendung eines Wissensbestandteils erhöht also seine Zugänglichkeit. Die Aktivierung nimmt mit der Zeit ab, wodurch Vergessen modelliert wird.

Die Stärke einer Gedächtnisspur wird bestimmt durch  $strength = \sum_j t_j^{-d}$ , wobei  $t_j$  die Zeit seit dem j-ten Kontakt mit dem Gedächtnisinhalt angibt. Für den Parameter  $d$  ist nach verschiedenen Studien die Wahl 0,5 ein für die Modellierung der Gedächtnisleistung geeigneter Wert. Die Menge der Kontakte nimmt mit der Zeit zu, so dass eine Neuberechnung der Stärke immer mehr Summanden umfassen würde. Glücklicherweise kann man aber den bereits zuvor berechneten Wert wieder verwenden (und mit  $(\Delta t)^{-d}$  multiplizieren, wobei  $\Delta t$  die seither verstrichene Zeit angibt) und muss nur die danach neu aufgetretene Kontakte durch Aufsummieren berücksichtigen. Wir nennen die Stärke auch ACT-Wert.

In unserem Ansatz wird der ACT-Wert verwendet, um das Wissen eines Benutzers über die einzelnen Funktionalitäten der Ziel-Software zu beschreiben. Kontakte sind die Durchführung der Funktionalität. Die Benutzer werden durch die Benutzererkennung anhand ihrer Kennung identifiziert. Da die Protokolldatei zusätzlich Zeitpunkt und Seite (und damit Funktionalität) enthält, stehen alle Parameter für die Aktualisierung des Benutzermodells zur Verfügung.

In [Igl05] wurde eine empirische Evaluation vorgestellt, die die Signifikanz des ACT-Werts als Schätzwert für das jeweilige Wissen eines Benutzers über Systemfunktionalitäten beweist. In dieser Evaluation ging es um die Eignung dieses Wertes als Grundlage für den Inferenzmechanismus eines adaptiven Hilfesystems, also genau um den auch in der vorliegenden Arbeit verwendeten Kontext.

## 4 Das adaptive Hilfesystem CHEetah

CHEetah (Extensible adaptive help system) ist ein modulares adaptives Hilfesystem für Web-basierte Systeme. Die Module sind so aufgebaut, dass Zielsystem-abhängige Teile unkompliziert ausgetauscht werden können, um die Anpassung an verschiedene Umgebungen zu erleichtern. CHEetah wurde in C++ und PHP implementiert. Es kann mehrere Systeme zugleich als Ziel-Software verwalten und ermöglicht daher ein systemübergreifendes Benutzermodell. Dabei können sich die jeweiligen Protokolldateien unterscheiden. Die grundsätzliche Funktionsweise ist in Abbildung 1 dargestellt. Aufgrund der Spezifika unseres Anwendungskontextes gibt es aber einige weitere Komponenten.

#### 4.1 Prozesserkennung

Wie zuvor beschrieben, wird in unserem Benutzermodell der ACT-Wert für das aktuelle Wissen jedes Benutzers und jeder Systemfunktionalität gespeichert. Ein Aufruf einer Seite in einem webbasierten System ist aber nicht gleichzusetzen mit der Verwendung einer Funktionalität. Vielmehr sind für eine Funktionalität mehrere aufeinander folgende Seitenaufrufe (Einzelaktionen) notwendig und es kann alternative Wege zur Ausführung der Funktionalität geben. Diese Möglichkeiten werden in sog. Prozessen beschrieben.

Das System CHEETAH verfügt über einen *ProcessDefiner* zur Definition und Eingabe der Prozesse in eine Prozessdatenbank. Der *ProcessDefiner* gehört nicht zum Laufzeitsystem, denn Prozesse werden nur bei Initialisierung definiert. Er arbeitet dabei als eine Art Makro-Recorder und versteht eine aufgerufene Seite mit ihren Parametern als Einzelaktion. Der (prozessdefinierende) Benutzer kann durch Eingabe einer Anfangs- und einer Endmarke sowie dazwischen einer beispielhaften Aufrufsequenz einen Prozess definieren.

Das Modul *ProcessRecognizer* erkennt aus den in der Protokolldatei für einen Benutzer gespeicherten Daten, welche der vordefinierten Prozesse der Benutzer zu welchem Zeitpunkt vollständig ausgeführt hat. Eine gewisse – definierbare - Anzahl von nicht-relevanten Seitenaufrufen (Fehlclicks oder parallele Ausführung einer anderen Funktionalität) innerhalb einer Funktionalität kann ignoriert werden. Start- und Endzeitpunkte eines vollständig durchgeführten Prozesses werden in einer Prozessprotokolldatei gespeichert. Da dieses Modul stets aktiv und zeitkritisch ist, handelt es sich um eine aufwendige Implementierung, auf die aus Platzgründen hier nicht näher eingegangen wird.

Um zu erkennen, welche Funktionalitäten der Benutzer vor Aufruf der Hilfe begonnen, aber nicht erfolgreich abgeschlossen hat, ermittelt das Modul *ContextRecognizer* Prozessanfangsstücke aus den vom *ProcessRecognizer* zurückgelieferten Protokolldaten.

#### 4.2 Benutzermodell

Der jeweilige ACT-Wert eines Benutzers für eine Funktionalität wird in einem Modul *ComputeUserModel* berechnet, wobei die Prozessprotokolle als Eingabe dienen und der frühere ACT-Wert wieder verwendet wird. Das Benutzermodell selbst besteht somit aus einer zweidimensionalen Tabelle, deren Einträge eine reelle Zahl für jeden Benutzer und jeden Prozess sind. Diese Werte werden bei der Auswahl von Hilfe-Items verwendet, also bei der Adaptionkomponente.

#### 4.3 Adaption

Abhängig vom angenommenen Vorwissen eines Benutzers zu einer Funktionalität soll ihm Hilfe in unterschiedlicher Form angeboten werden. Dabei ist nicht offensichtlich, welche Form der Hilfe für Benutzer mit wenig, mittelmäßig oder viel Wissen geeignet ist. Dieser Aspekt wird im folgenden Kapitel im Zusammenhang mit der Evaluation betrachtet. Wie in der Einleitung angekündigt, sollen trotz vorgesehener Adaption Hilfe-Items nicht in mehreren Versionen erstellt werden müssen, um Arbeit zu sparen und um Aktualität, Konsistenz und Korrektheit der Hilfeinhalte zu unterstützen. Ein möglicher Ausweg aus diesem scheinbaren Widerspruch ist das adaptive Ein- bzw. Ausblenden

einzelner Hilfe-Bestandteile, wie für wiederverwendbare Lehrmodule z.B. in [DKS99] vorgeschlagen.

Ein Hilfe-Item besteht daher aus mehreren Bestandteilen, die jeweils je nach Adaption ein- oder ausgeblendet werden. Um die Bestandteile unterscheiden zu können, liegen sie im XML-Format vor. Die folgende DTD beschreibt das Hilfeformat und zugleich mögliche Typen eines Hilfe-Items, aus denen bei der Adaption ausgewählt werden kann.

```
<!ELEMENT item (purpose?,way?,interaction?,process*,example*,tip*)>
<!ATTLIST item id ID REQUIRED>
<!ELEMENT purpose (#PCDATA)>
<!ELEMENT way (#PCDATA)>
<!ELEMENT interaction (intro?,element*)>
<!ELEMENT process (intro?,step*)>
<!ATTLIST process type ("short"|"long") REQUIRED "short">
<!ELEMENT example (intro?,step*)>
<!ELEMENT tip (#PCDATA)>
<!ELEMENT intro (#PCDATA)>
<!ELEMENT element (#PCDATA)>
<!ATTLIST element name CDATA REQUIRED
                type (text|textarea|button|checkbox|select|icon|link) IMPLIED
                desc CDATA IMPLIED>
```

Die tatsächliche Auswahl aus diesen Bestandteilen erfolgt mit Hilfe von XSL-Style-sheets, die die angepassten Hilfe-Items in HTML-Form ausgeben. Eine Adaptionsregel weist jeder einzelnen Komponente eines Hilfe-Items den Status „eingebildet“, „einblendbar“ oder „ausgeblendet“ zu. Grundlage der Adaptionentscheidungen ist eine explorative Studie, die in Abschnitt 5.3. beschrieben wird.

Schließlich wird die generierte Hilfeseite durch das Modul *HelpViewer* angezeigt. Einblendbare Informationsbestandteile werden durch Klicken auf die Überschrift sichtbar. Ein Button „weitere Informationen“ versetzt per JavaScript-Funktion alle ausgeblendeten Bestandteile in den Status „einblendbar“. Das Aussehen der Hilfebestandteile kann mit Hilfe von CSS an das gewünschte Design angepasst werden.

## 5 Evaluation

Ohne aussagekräftige Evaluation ist der Wert eines adaptiven Hilfesystems nicht feststellbar, auch wenn viele Entwurfsentscheidungen plausibel sein mögen. So werden in der einschlägigen Literatur kaum noch Systeme ohne Evaluation vorgestellt. Entwurfsentscheidungen für das in dieser Arbeit behandelte System CHEetah wurden in mehreren Studien evaluiert, von denen eine bereits veröffentlicht wurde. Eine ausführliche Dokumentation der Studien in diesem Beitrag scheitert aus Platzgründen. Wir werden deshalb nur die jeweils untersuchte Frage, das jeweilige Design und den Ablauf einer Studie sowie eine Zusammenfassung der Ergebnisse darstellen. Details findet man in [Igl06].

Eine Evaluation soll nicht nur aussagekräftige Antworten auf die Frage liefern, ob ein adaptives System dem Benutzer Vorteile bringt, sondern auch, welche Vorteile dies sind und wie sie zustande kommen. Wir orientieren uns an der Dissertationsschrift [Wei03], in der ein allgemeines Framework für die Evaluation benutzeradaptiver Systeme vorgeschlagen wird, und an [PW05]. Danach sind zu unterscheiden die Evaluation der Zuverlässigkeit und Gültigkeit der **Eingabedatenerfassung**, des **Inferenzmechanismus**, der **Adaptionentscheidungen** und der gesamten **Interaktion** zwischen System und

Benutzer. Jede dieser Evaluationen ist Voraussetzung für die nächste. Die Studien können explorativ, kontrolliert mit simulierten oder mit realen Nutzern oder aber in der realen Praxis durchgeführt werden.

### 5.1 Ziel-Software und Eingabedatenerfassung

Für eine Evaluation des implementierten Hilfesystems ist eine geeignete Ziel-Software notwendig, also ein webbasiertes System mit mehreren Funktionalitäten bzw. Prozessen, zu denen Hilfe sich in irgendeiner Weise anbietet. Dieses System ist in unserem Fall WebTime, ein webbasierter Gruppenkalender, der im Rahmen eines Projektes der Katholischen Universität Eichstätt-Ingolstadt mit einer mittelständischen Software-Firma entwickelt wurde. Er hat die folgenden Funktionalitäten:

<b>Terminkalender</b>	Termine für einzelne Gruppenmitglieder anlegen, löschen, suchen, gemeinsamen Termin finden, Termine mit Checklisten verknüpfen, Terminarten anlegen, Rechtevergabe
<b>ToDo-Liste</b>	ToDo-Einträge anlegen, bearbeiten, in Kalenderübersicht und ToDo-Übersicht anzeigen
<b>Datenaustausch</b>	Anzeigen der Termine von verschiedenen Webservern per SOAP, Daten-export und Datenimport per vCal
<b>Protokollierung</b>	von Aktivitäten, Aufzeichnen von Prozessen
<b>Urlaubsverwaltung</b>	Beantragen von Urlaub, Erstellen und Kontrollieren von Urlaubsregeln, An- und Abwesenheitsübersicht für verschiedene Arbeitsgruppen, Anzeigen der Urlaubskonten

WebTime zeichnet die Aktivitäten (jeden Server-seitigen Zugriff) des Benutzers in einer XML-basierten Protokolldatei auf. Diese Datei wird vom *ProcessRecognizer* gelesen. Sie enthält die folgenden Angaben: **Benutzerkennung** (verschlüsselt aus Datenschutzgründen), **Zeitstempel** (der letzte Aufruf einer Benutzerhandlung definiert den Zeitpunkt der gesamten Handlung), **Seitenname** (zur Erkennung einer Handlung), **Referer** (von wo aus wurde die Seite aufgerufen?, wird genutzt für die Prozesserkennung), **POST-, GET-, COOKIE und SESSION-Variablen** (werden bei der Prozesserkennung zur Unterscheidung von Einzelaktionen verwendet), **Sessionkennung** (identifiziert einen zusammenhängenden Besuch eines Benutzers).

Die Korrektheit der Aufzeichnungen wurde durch stichprobenhafte Kontrollen überprüft. Die Reliabilität und Validität der Daten kann angenommen werden. Eine Evaluation der Eingabedatenerfassung ist daher nicht notwendig.

### 5.2 Evaluation des Inferenzmechanismus

Eine entsprechende Studie, veröffentlicht in [Igl05], zeigt, dass der ACT-Wert für Benutzer mit geringen Erfahrungen recht zuverlässig die subjektive Expertise vorhersagt.

Die 19 Teilnehmer der Studie wurden über 12 Monate bei der Benutzung von WebTime beobachtet und ihre Aktivitäten protokolliert. Die berechneten ACT-Werte wurden anschließend auf Zusammenhänge mit der subjektiven und der objektiven Expertise der zugehörigen Funktionalität überprüft. Zusätzliche Vergleichsvariable war die Selbsteinschätzung zur allgemeinen Computer-Erfahrung. Die subjektive Expertise wurde

durch Fragebogen, die objektive Expertise durch ein Online-Experiment erhoben, in dem neben der Korrektheit der Ausführung auch die Ausführungszeit gemessen wurde. Es wurden zwei Funktionalitäten von WebTime betrachtet (**neuen Termin anlegen**, **neues ToDo anlegen**).

Das Ergebnis der Studie ist eine signifikante Korrelation zwischen der subjektiven Expertise und dem ACT-Wert bei der Funktionalität **Neues ToDo anlegen**, das die Versuchspersonen zuvor wenig benutzt hatten. Auch für die andere Funktionalität sowie für die objektive Expertise konnten positive Korrelationen ermittelt werden, die allerdings nicht signifikant sind. Weder die Ausführungszeit noch die Computer-Erfahrung korrelieren mit dem ACT-Wert nennenswert.

### 5.3 Evaluation der Adaptionentscheidungen

Hilfeb Bestandteile sollen in Abhängigkeit vom jeweiligen Wissen ein- oder ausgeblendet werden. Das Wissen wird durch den entsprechenden ACT-Wert repräsentiert. Nach [Wei03] handelt es sich hier weniger um eine Evaluation im eigentlichen Sinn, als um eine explorative Studie zur Ermittlung einer empirischen Grundlage für die Adaptionentscheidungen.

Nach subjektiver Einschätzung der Erfahrung mit WebTime und der allgemeinen Computer-Erfahrung beurteilten die Versuchspersonen Bestandteile von Hilfe-Items bezgl. dreier Funktionalitäten von WebTime auf den Dimensionen Interessantheit und Nützlichkeit.

Die betrachteten Funktionalitäten waren **Termin suchen** (nach verschiedenen Kriterien, Ausgabe als PDF-Datei), **ToDo anlegen** (neue Aufgabe in persönliche ToDo-Liste) und **Sachbearbeiter Reihenfolge ändern** (Reihenfolge in Terminübersicht). Die Prozesse wurden so ausgewählt, dass sie von den Versuchspersonen unterschiedlich häufig ausgeführt wurden, sich aber in Komplexität und Aufbau glichen. Die folgenden Hilfeb Bestandteile wurden erstellt: **Zweck des Prozesses**, **Weg zur Funktion**, **Beschreibung der Interaktionselemente**, **Prozessbeschreibung kurz**, **Prozessbeschreibung lang**, **weiterführende Tipps**, **umfassendes Beispiel**.

Ein Pre-Test in zwei Runden (jeweils mit sieben Mitarbeitern des Lehrstuhls) stellte sicher, dass vorgeschlagene Hilfeb Bestandteile klar diesen Kriterien entsprechen.

Die eigentliche Studie untersuchte, wie sich die Einschätzung von Hilfeb Bestandteilen bei verschiedenen Informationszielen (wie kann ich eine Funktionalität ausführen / was gibt es an neuen Funktionalitäten / was kann ich in meiner Situation machen?), bei verschiedenen Ausprägungen von Wissen (gegeben durch den jeweiligen ACT-Wert) und allgemeiner Computer-Erfahrung verhalten. Eine Online-Befragung enthielt pro Seite einen Hilfeb Bestandteil, der nach Interessantheit und Nützlichkeit beurteilt werden sollte. Beide Merkmale wurden über eine sechsstufige Likertskala (trifft gar nicht zu, ..., trifft sehr zu) erfasst. Ihre allgemeine Computer-Erfahrung sollten die Versuchspersonen selbst einschätzen (Neuling / Anfänger / Fortgeschrittener / Experte). Die Informationsziele wurden dadurch simuliert, dass die Versuchspersonen sich in eine entsprechende Situation hineinversetzen sollten, also einen entsprechenden Text vorgelegt bekamen.

Die Aktivitäten von acht Versuchspersonen (zwischen 23 und 46 Jahre alt, drei Männer, fünf Frauen) innerhalb der Ziel-Software wurden etwa ein Jahr lang protokolliert. Die jeweiligen ACT-Werte lagen zwischen 0 und 16,15. Auf dieser Grundlage wurden die Versuchspersonen für jede Funktionalität in eine der folgenden Gruppen eingeordnet:

ACT = 0	wurde noch nie ausgeführt
ACT < 0,7	wurde selten oder lange zurückliegend ausgeführt
ACT < 1,0	wurde gelegentlich ausgeführt
ACT < 4,0	wird regelmäßig verwendet
ACT ≥ 4,0	wird häufig ausgeführt

Zwei Teilnehmer schätzten sich als fortgeschrittene Computer-Nutzer ein, fünf Teilnehmer als Experten. Diese Verteilung war unabhängig von den ACT-Werten.

Die Auswertung der Studie zeigte, dass das Informationsziel des Benutzers geringe Auswirkungen auf die Beurteilung von Hilfebestandteilen hat. Dies steht im Widerspruch zu den Ergebnissen in [Höö96] im Zusammenhang mit dem adaptiven Hilfesystem PUSH, das sich mit Erfolg an diese Variable adaptiert. Der Unterschied mag von der gewählten Ziel-Software abhängig sein.

Benutzer mit mehr Computer-Erfahrung haben laut der Studie allgemein einen geringeren Bedarf an Hilfe. Da allerdings keine der Versuchspersonen angab, geringe oder keine allgemeine Computer-Erfahrung zu besitzen, kann die Studie keine Aussagen über die Beurteilung von Hilfebestandteilen durch Computer-Novizen machen.

Der Zusammenhang zwischen ACT-Wert und Bewertung der Hilfebestandteile ist deutlich und teilweise überraschend. Verschieden erfahrene Benutzer bevorzugten unterschiedliche Hilfebestandteile. Dabei ist die Beurteilung nicht immer linear, sondern kann im mittleren Expertisestadium die niedrigsten Werte aufweisen. Die Ergebnisse führten zu folgenden Adaptionentscheidungen des Hilfesystems (aus = ausgeblendet, ein = einblendet, aA = auf Anforderung einblendbar):

<b>Bestandteil</b>	<b>ACT</b>	<b>= 0</b>	<b>&lt; 0,7</b>	<b>&lt; 1</b>	<b>&lt; 4</b>	<b>≥ 4</b>
Zweck		aA	ein	aus	aA	aA
Weg		aA	ein	aA	aA	aA
Interaktion		aA	ein	aus	aA	aA
Prozess kurz		aus	ein	aus	aus	aA
Prozess lang		ein	aus	ein	ein	aA
Beispiel		aA	aA	ein	ein	aA
Tipps		ein	aA	aA	ein	ein

#### 5.4 Evaluation der Gesamtinteraktion

Mit Hilfe dieser Studie sollte der Erfolg des Hilfesystems in der Gesamtinteraktion überprüft werden. Sie wurde in Form einer kontrollierten Studie mit 61 neuen Versuchspersonen (Studierende, Akademiker) durchgeführt, deren Interaktion mit WebTime zwei Wochen lang protokolliert wurde. Es wurde entweder keine Hilfe, nicht-adaptive Hilfe oder adaptive Hilfe durch CHEetah angeboten. Zusammenfassend sollen hier nur Hypothesen und die entsprechenden Ergebnisse der Evaluation genannt werden:

**H1: Größere Korrektheit der Ausführung bei angebotener Hilfe im Vergleich zu keiner Hilfe**  
Die Hypothese wurde durch die Studie deutlich bestätigt. Sie gilt sowohl für objektiven als auch für subjektiven Erfolg.

**H2: Weniger irrelevante Aktionen bei angebotener Hilfe im Vergleich zu keiner Hilfe**

Auch diese Hypothese wurde durch die Studie bestätigt. Zudem zeigt sich ein deutlicher Vorteil der adaptiven Hilfe.

**H3: Schnellere Ausführung bei adaptiver Hilfe im Vergleich zu statischer Hilfe bei wenigstens gleicher Korrektheit**

Wurde nicht bestätigt.

**H4: Geringere Varianz in der Einschätzung von Interessantheit und Nützlichkeit bei adaptiver Hilfe als bei statischer Hilfe**

Wurde durch die Studie deutlich bestätigt.

**H5: Größere Nützlichkeit von adaptiver Hilfe im Vergleich zu statischer Hilfe**

Wurde durch die Studie deutlich bestätigt.

**H6: Größere Interessantheit von adaptiver Hilfe im Vergleich zu statischer Hilfe**

Es ist zwar eine Korrelation in der erwarteten Richtung festzustellen, diese ist jedoch nicht signifikant.

**H7: Geringere Annahme von Hilfe bei mehr Wissen oder mehr Computer-Erfahrung**

Wurde deutlich bestätigt.

## 5 Abschluss

Wir haben das adaptive Hilfesystem CHEetah vorgestellt, seine Funktionsweise und Architektur erläutert und Evaluationsergebnisse dargestellt. Bei der Eingabedatenerfassung haben wir die Möglichkeit der Server-seitigen Protokollierung relevanter Daten web-basierter Systeme genutzt. Die Inferenz des Benutzermodells basiert auf der ACT-Theorie, die insbesondere auch Vergessen modelliert. Die Adaptionentscheidungen wurden explorativ ermittelt. Adaption geschieht durch Ein- oder Ausblenden von Hilfe-Bestandteilen, so dass Hilfetexte nicht in mehreren Versionen erzeugt und verwaltet werden müssen. Das System ist im Zusammenhang mit der Ziel-Software WebTime vollständig implementiert, kann aber zusammen mit der Ziel-Software aus Lizenzgründen nicht frei zur Verfügung gestellt werden. Aufgrund seines modularen Aufbaus kann es leicht an andere webbasierte Systeme angepasst werden.

Die Grundidee unseres Ansatzes ist auf jede Art von Benutzer-Software anwendbar, bei der die Hilfe Unterstützung bei verschiedenen Funktionalitäten darstellt. Die spezifische Umsetzung verwendet dagegen Spezifika webbasierter Systeme. Sie lässt sich von der betrachteten Ziel-Software auf andere webbasierte Systeme leicht übertragen, sofern Benutzer sich mit einer Kennung identifizieren, ihre Aktionen beobachtbar und damit protokollierbar sind und die Funktionalitäten der Ziel-Software durch diese Aktionen erkannt werden können. Die Verwendung der Inferenz- und der Adaptionmethode ist grundsätzlich auf andere adaptive Lehr-/Lernsysteme übertragbar, denn auch dort geht es um Wissensbestandteile, die vergessen werden und um die geeignete Darstellung gerade des Wissens, das der Lernende benötigt. Die Aussagekraft der Evaluation ist naturgemäß immer etwas eingeschränkt durch spezielle Umstände bei den Studien. Die grundsätzlichen Ergebnisse der Studien halten wir zwar für belastbar. Gewisse Einschränkungen dadurch, dass die Versuchspersonen und ihre Situation nicht wirklich die Gesamtheit der potentiellen Benutzer repräsentieren, müssen aber hingenommen werden.

Auch wenn die Anpassung von CHEetah an andere webbasierte Software unproblematisch ist, gibt es stets Unterschiede, die z.B. neue Hilfe-Items sinnvoll erscheinen lassen oder neue Varianten der Eingabedatenerfassung. Dies gilt insbesondere für allgemeine Lehr-/Lernsysteme, bei denen der Wissenserwerb nicht mit einer Funktionsausführung gleichgesetzt werden kann. Im Rahmen einer spin-off Unternehmung werden derartige Anpassungen und Erweiterungen als Dienstleistung angeboten. Zudem überlegen wir, die Berechnung des ACT-Wertes aufgrund der nicht ganz erwartungsgemäßen Ergebnisse zu verfeinern. Das zentrale Ziel wurde aber mit unserer Arbeit erreicht: Ein Hilfesystem zu erstellen, das den Benutzer in seinem individuellen Lernprozess begleitet ohne gleichzeitig den Arbeitsaufwand für Hilfeautoren übermäßig zu erhöhen.

## Literaturverzeichnis

- [And02] Anderson, J.R.: ACT: A simple theory of complex cognition. In: Cognitive Modelling, MIT Press, 2002, S. 48-68
- [AL98] Anderson, J.R.; Lebiere, C.: The Atomic Components of Thought. Lawrence Erlbaum Associates, 1998
- [BR96] Benham, H.C.; Raymond, B.C.; Information technology adoption: evidence from a voice mail introduction. SIGCPR Comput. Pers. 17 (1996) 1, S. 3-25
- [Dav89] Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13 (1989) 3, S. 319-339
- [DKS99] Desel, J.; Klein, M; Stucky, W.: Virtuelle Kurse durch Wiederverwendung didaktischer Lehrmodule. Bericht Nr. 395, Institut AIFB, Universität Karlsruhe (1999)
- [Gel98] Gelernter, B.: Help design challenges in network computing. Proc. Computer Documentation 1998, ACM Press, S. 184-193
- [Hö096] Höök, K.: A Glass Box Approach to Adaptive Hypermedia. Dissertation, Stockholm University, 1996
- [Igl04] Iglezakis, D: Adaptive help for web based applications. Adaptive Hypermedia and Adaptive Web-based Systems, Springer LNCS 3137 (2004), S. 304-307
- [Igl05] Iglezakis, D: is the ACT-value a valid estimate for knowledge? An empirical evaluation of the inference mechanism of an adaptive help system. Workshop on the Evaluation of Adaptive Systems, User Modelling 2005, Springer LNCS 3538 (2005)
- [Igl06] Iglezakis, D: Hilfreiche Hilfe – Methode, Implementierung und Evaluation eines Verfahrens zur Adaption an das prozedurale Wissen, Dissertation, Katholische Universität Eichstätt-Ingolstadt (2006) und Shaker Verlag (2007)
- [MPC01] Mathieson, K; Peacock, E., Chin, W.W.: Extending the technology acceptance model: the influence of perceived user resources. The DATA BASE for Advances in Information Systems 32 (2001) 3, S. 86-112
- [Nie00] Nielsen, J.: Novice vs. expert users. Version: Feb 2000. [www.useit.com/alertbox/20000206.html](http://www.useit.com/alertbox/20000206.html) (Abruf 17.12.2006)
- [PW05] Paramythis, A.; Weibelzahl, S.: A decomposition model for the layered evaluation of interactive adaptive systems. User Modeling 2005, Springer LNAI 3538 (2005), S. 438-432
- [Wei03] Weibelzahl, S.: Evaluation of Adaptive Systems. Dissertation, Universität Freiburg (2000)
- [WMO91] Waern, Y.; Malmsten, N.; Oestreicher, L.; Hjalmarsson, A.; Gidlöf-Gunnarsson, A.: Office automation and user's need for support. Behaviour & Information Technology 10 (1991) 6, S. 501-514
- [XL00] Xia, W.; Lee, G.: The influence of persuasion, training and experience on user perceptions and acceptance of IT innovation. Proc. of ICIS'00, Ass. for Information Systems, 2000, S. 371-384