

TASK ANALYSIS

Sebastian Burger

Sebastian.Burger@rwth-aachen.de

Christoph Will

Christoph.Will@rwth-aachen.de

RWTH Aachen

Media Computing Group

ÜBERSICHT

Die “Task Analysis” ist eine Sammlung von Techniken zur Analyse und Modellierung von Aufgaben, die im Bereich der Mensch-Maschine-Interaktion (*HCI*) eingesetzt werden. In dieser Ausarbeitung wird dem Leser ein Überblick über dieses Gebiet gegeben. Dazu werden zuerst, nach einer kurzen allgemeinen Einführung, einige der vielen Techniken zur Modellierung von Aufgaben vorgestellt. Diese sind zur besseren Veranschaulichung den verschiedenen grundlegenden Ansätzen entnommen:

- Zerlegung in Unteraufgaben (*task decomposition*)
- wissensbasierte Techniken (*knowledge-based techniques*)
- Objekt- und Bezugsbasierte Techniken (*entity-relationship-based techniques*).

Im Anschluss wird ein Einblick gegeben in die verschiedenen Methoden, mit denen die der Analyse zu Grunde liegenden Daten gesammelt werden können. Zuletzt wird beispielhaft die Anwendung dieser Techniken auf konkrete Probleme gezeigt.

SCHLÜSSELWÖRTER

Aufgabenmodellierung, Zerlegung in Unteraufgaben, wissensbasierte Techniken, Objekt- und Bezugsbasierte Techniken, Informationsquellen

EINFÜHRUNG

Allgemein

“Task Analysis” beschreibt die verschiedenen Methoden und Techniken, mit denen man analysieren und modellieren kann, wie Menschen Aufgaben bearbeiten. Hierbei lassen sich die konkreten Handlungen durch Techniken beschreiben, die auf dem Ansatz der Zerlegung in Unteraufgaben (*task decomposition*), basieren. Die Auslöser für bestimmte Aktionen werden dabei jedoch weniger beachtet. Hierfür eignet sich eine objektorientierte Analyse (*entity-relationship-based analysis*), besser, da sie besonderen Wert auf die Beziehungen zwischen Agierenden und Objekten legt – dabei allerdings die Detailtiefe der Zerlegung in Unteraufgaben vermissen lässt. Das zur Bewältigung der Aufgabe nötige Vorwissen wird bei beiden Methoden allerdings nicht berücksichtigt, weshalb sich zur Ermittlung desselbigen besonders die wissensbasierten Modelle eignen.

Welchen der drei grundlegenden Ansätze man verfolgt, hängt insbesondere von dem Ziel ab, dass der Analyse zu Grunde liegt: Möchte man z.B. Ein Benutzerhandbuch entwerfen, ist natürlich das nötige Wissen zur Ausführung eines Arbeitsschrittes von besonderer Bedeutung. Dementsprechend bietet sich in diesem Fall eine wissensbasierte Analyse an.

Unterschiede zu anderen Techniken

Was unterscheidet die “task analysis” von anderen Techniken?

Kognitive Modelle z.B. können auch vor der eigentlichen Implementierung einer Software benutzt werden, da man sie auch auf rein theoretische Vorgänge anwenden kann. Methoden wie die objektbasierten Analysen hingegen setzen eine detaillierte Kenntnis der Aufgabenstellung, der zur Bewältigung nötigen Abläufe und der beteiligten Personen und Objekte voraus. Da dieses Wissen hauptsächlich durch Observation gewonnen werden kann, lassen sich also solche Techniken besser auf bereits existierende Systeme anwenden.

Betrachtet man nun zielorientierte Analysemethoden, wie GOMS, die auf den ersten Blick der Zerlegung in Unteraufgaben relativ ähnlich sind, lässt sich ein weiterer Unterschied feststellen: das Ziel der Analyse. Modelle wie GOMS versuchen eher die internen kognitiven Prozesse des Nutzers zu verstehen, also das “Warum” hinter einer Aktion. “Task analysis” auf der anderen Seite, interessiert sich eher für das äußerlich beobachtbare Verhalten der Nutzer und somit mehr für das “Was” dieser tut. Hier lässt sich allerdings keine klare Grenze ziehen, da selbst eine sehr objektive Analyse immer noch einige Rückschlüsse auf die Absichten des Nutzers ziehen wird. Mit der zunehmenden Bedeutung der HCI und Anwenderorientierung der Softwareentwicklung, zeigt sich auch der entscheidende Unterschied zwischen der “task analysis” und ihren sehr ähnlichen Vorfahren: Diese hatten als Hauptziel, das System aus Entwicklersicht zu modellieren, während der Schwerpunkt heute auf der Sichtweise des Nutzers liegt.

TECHNIKEN ZUR ANALYSE UND MODELLIERUNG

Task decomposition

Offensichtlich macht es in den meisten Fällen wenig Sinn, nur die übergeordnete Zielsetzung zu betrachten, da diese

oft zu allgemein gefasst ist. Es ist also notwendig, Aufgaben in einzelne Teilaufgaben aufzubrechen. Dadurch wird das Modell realistischer und entspricht somit eher der Sicht des Nutzers.

Viele Analysetechniken nutzen diese Vorgehensweise in unterschiedlichem Ausmaß. Im Folgenden wird die "hierarchical task analysis" (HTA) als typischer Vertreter dieses Ansatzes vorgestellt. Dieses Verfahren hat, wie der Name suggeriert, als Ergebnis eine hierarchische Anordnung der Aufgaben und Unteraufgaben. Weiterhin lassen sich mit ihr sogenannte "Pläne" erstellen, die beschreiben, unter welchen Bedingungen und in welcher Reihenfolge Teilaufgaben ausgeführt werden. Zur Veranschaulichung betrachten wir eine alltägliche Situation: die Zubereitung eines Spiegeleis. Was sind nun offensichtliche Teilaufgaben dieses Vorgangs? Zuerst wären da die Vorbereitung der Pfanne, danach das eigentliche Zubereiten des Eis sowie die Nachbereitung. Somit ergibt sich die in Tabelle 1 gezeigte Einteilung. Plan 0 bezeichnet hierbei, in welcher Reihenfolge welche Teilaufgaben der Aufgabe abgearbeitet werden sollen. Für einen unerfahrenen Nutzer wäre diese Aufteilung jedoch zu grob. Daher wird eine weitere Ebene eingeführt, wie in Tabelle 2 zu sehen ist. Diese dient der weiteren Konkretisierung der Unteraufgaben, in unserem Beispiel der Aufgaben 1 und 2. Es fällt auf, dass der lineare Aufbau nicht dem realen Ablauf der Aktionen entspricht: Man könnte z.B. das Ei aus dem Kühlschrank holen, während die Pfanne heiß wird. Offensichtlich gibt es mehrere Möglichkeiten zu "warten": Einerseits der "busy wait", währenddessen man eine andere Handlung ausführen kann, andererseits einen "idle wait", welcher gleichzeitig keine weiteren Aktionen zulässt. In beiden Fällen wird meistens auf das Eintreten eines bestimmten Ereignisses gewartet und nicht auf das Ende einer festen Zeitspanne. Überhaupt spielen Bedingungen eine große Rolle beim Aufstellen der "Pläne", wie im Beispiel an Plan 2 zu sehen ist, da der Punkt 2.4 nur ausgeführt wird, wenn es vom Nutzer gewünscht ist. Es ist auch nicht immer notwendig, im Gegensatz zu diesem Beispiel, Teilaufgaben in einer bestimmten Reihenfolge auszuführen.

Das aufgeführte Beispiel verdeutlicht, dass "task analysis" oftmals keine exakte Wissenschaft ist, denn das Ergebnis

- | |
|--|
| 0. Spiegelei zubereiten
1. Pfanne und Herd vorbereiten
2. Spiegelei braten
3. Ei auf den Teller legen
4. Herd ausschalten
Plan 0: führe 1 – 2 – 3 – 4 in dieser Reihenfolge durch |
|--|

Tabelle 1: Erster Schritt der HTA

0. Spiegelei zubereiten

1. Pfanne und Herd vorbereiten

1.1 Pfanne aus dem Schrank holen

1.2 Pfanne auf den Herd stellen

1.3 Herd anschalten

1.4 Warten bis die Pfanne heiß ist

2. Spiegelei braten

2.1 Ei aus dem Kühlschrank holen

2.2 Ei in die Pfanne schlagen

2.3 Warten bis das Ei durch ist

2.4 Zweite Seite des Eis

2.4.1 Ei wenden

2.4.2 Warten bis zweite Seite auch durch ist

3. Ei auf den Teller legen

4. Herd ausschalten

Plan 0: führe 1 – 2 – 3 – 4 in dieser Abfolge durch

Plan 1: führe 1.1 – 1.2 – 1.3 – 1.4 in dieser Abfolge durch

Plan 2: führe 2.1 – 2.2 – 2.3 – falls gewünscht 2.4

in dieser Abfolge durch

Plan 2.4: führe 2.4.1 – 2.4.2 in dieser Abfolge durch

Tabelle 2: Zweiter Schritt der HTA

wird von der Sichtweise des Analysierenden, sowie der Zielsetzung der Analyse beeinflusst. Man hätte problemlos in diesem Beispiel einzelne Schritte noch viel detaillierter beschreiben können, um z.B. eine präzise Anleitung für einen "Anfänger" zu entwerfen. Der erste Schritt der HTA (Tabelle 1) hingegen gibt schon einen guten Überblick über den gesamten Vorgang, was für einen "fortgeschrittenen" Nutzer schon ausreichend sein könnte.

Da die HTA iterativ ist, weil Teilaufgaben Schritt für Schritt zerlegt werden, stellt sich die Frage, wann die Iteration zu stoppen ist. Hier lässt sich, auf Grund der vielfältigen Anwendungsmöglichkeiten, keine eindeutige Regel festlegen. Die sogenannte "P * C – Regel" bietet dabei einen ersten Ansatz: Ist das Produkt der Wahrscheinlichkeit, dass der Nutzer einen Fehler begeht (P) und den Kosten, die dieser Fehler verursachen würde (C), unterhalb einer bestimmten Grenze, so ist es nicht notwendig diesen Schritt detaillierter zu beschreiben. Weiterhin wäre es auch nicht sinnvoll, motorische Funktionen zu erläutern, wie z.B. Mausbewegungen, da diese weder nützlich, noch besonders genau wären. Ebenso lassen sich komplexere "interne" Entscheidungen, die über simple "Ja/Nein-Fragen" hinausgehen, offensichtlich

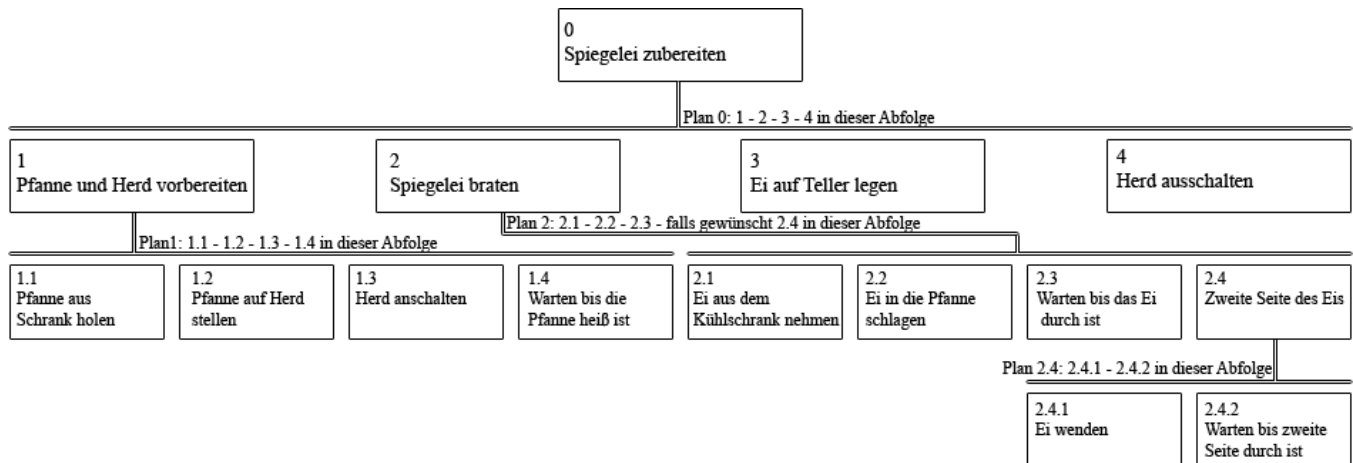


Abbildung 1: Ergebnis einer HTA als Baumdiagramm

schwer in der HTA modellieren. Das Ergebnis einer HTA lässt sich leicht von einer Textdarstellung in ein anschauliches Baumdiagramm überführen, wie in Abbildung 1 dargestellt. Ob der Analysierende die Informationen in einer formalen Notation, als Tabelle oder als Diagramm präsentiert, ist seiner persönlichen Präferenz überlassen. Natürlich macht es oft wenig Sinn, bereits zu Beginn eines Designprozesses mit einer sehr formalen und daher auch komplexen Darstellung zu arbeiten. Der gesamte Prozess ist alles andere als geradlinig, es kann sogar vorkommen, dass auf Grund von Zwischenergebnissen das ursprüngliche Hauptziel abgeändert werden muss. Die Qualität des Ergebnisses hängt somit von den Fähigkeiten des Analysierenden ab. Weiterhin werden verschiedene Personen unterschiedliche Ergebnisse produzieren, die aber alle "richtig" sein können, abhängig von Zielsetzung und Durchführung der Analyse.

Knowledge-based techniques

Das Ziel wissensbasierter Analysetechniken ist es, ein Verständnis für das zur Lösung einer Aufgabe benötigte Wissen zu schaffen, sowie Aufgaben diesbezüglich miteinander vergleichen zu können. Als Ergebnis einer solchen Analyse erhält man in der Regel eine Taxonomie, wie man sie z.B. aus der Biologie kennt: Tiere lassen sich in Wirbeltiere und wirbellose Tiere unterteilen, die Wirbeltiere wiederum in Fische, Reptilien, Säugetiere etc. Bei der "task analysis" versuchen wir jedoch die Objekte und Aktionen, die an der Bearbeitung der Aufgabe beteiligt sind, zu klassifizieren. Hierzu erstellt man zuerst eine möglichst umfangreiche Liste aller beteiligten Objekte und sortiert diese. Dies kann von einem Experten des Gebiets übernommen werden, worauf jedoch noch im Abschnitt über Informationsquellen näher eingegangen wird. Genau wie bei der HTA, hängt auch bei den wissensbasierten Techniken von der Zielsetzung der Analyse ab, wie detailliert man die Daten sammelt und sortiert. Es ist

allerdings besonders in der Anfangsphase besser, mehr Daten als nötig zusammenzutragen, da das Streichen überflüssiger Dinge einfacher ist als nachträgliches Einfügen. Eine konkrete Technik aus diesem Bereich ist die "task analysis for knowledge description" (TAKD) die als zugrunde liegende Taxonomie die "task descriptive hierarchy" (TDH) benutzt. In dieser Taxonomie gibt es drei Arten von "Zweigen": ausschließendes Oder (XOR), einschließendes Oder (OR), sowie Und (AND). Betrachten wir nun Tabelle 3: Wir können die Fahrzeuge bezüglich verschiedener Eigenschaften sortieren. Zuerst einmal nach der Anzahl ihrer Räder und ihrer Motorisierung. Da jedes der vier Fahrzeuge sowohl eine bestimmte Anzahl Räder, als auch eine Form des Antriebs besitzt, werden diese Kategorien durch ein "AND" verbunden. Die Unterscheidungen innerhalb der Radanzahl und der Motorisierung werden jedoch auf "entweder-oder"-Basis getroffen und sind eindeutig. Daher wird hier "XOR" verwendet. Wenn ein Objekt in mehr als einer Kategorie vorkommen kann, wird dies mit dem Schlüsselwort "OR"

Fahrzeug AND

Antrieb XOR

Motorisiert

Auto, Motorrad

Nicht motorisiert

Fahrrad, Seifenkiste

Anzahl Räder XOR

Vier Räder

Auto, Seifenkiste

Zwei Räder

Fahrrad, Motorrad

Tabelle 3: TAKD mit Hilfe von TDH

Fahrzeug OR
Ein Passagier
Fahrrad, Auto, Motorrad
Zwei Passagiere
Auto, Motorrad
Drei oder mehr Passagiere
Auto

Tabelle 4: OR-Zweig in einer TDH

gekennzeichnet, wie in Tabelle 4 gezeigt. TAKD versucht jedes Objekt eindeutig zu repräsentieren. Diese Regel kann jedoch nicht immer eingehalten werden, da unter Umständen die Taxonomie viel zu detailliert wird. Auch hier hängt wiederum die Detailtiefe von der geplanten Anwendung der Analyse ab. In Tabelle 5 ist nun jedes Objekt durch den zu ihm führenden Pfad im Baum von jedem beliebigen Anderen zu unterscheiden. Die voranstehenden Zeichen “/”, “|” und “{” stehen in der TDH jeweils für AND, XOR und OR und ersetzen diese. Der eindeutig zu einem Objekt führende Pfad wird z.B. für das Fahrrad wie folgt beschrieben: Fahrzeug/Antrieb(nicht motorisiert)/Anzahl möglicher Passagiere{Ein Passagier}/. Eine solche Beschreibung nennt man einen “Satz” in der “knowledge representation grammar” (KRG). Diese ermöglicht es, Wissen über Objekte formal darzustellen. Sie hilft folglich dabei, in einer TDH schnell die Eindeutigkeit zu überprüfen.

Möchte man nun eine Taxonomie oder TDH von Aktionen aufstellen, geht man genauso vor wie bei Objekten: Zuerst wird eine Liste aller Aktionen aufgestellt und dann in einer Hierarchie sortiert. Hierzu ziehen wir wieder das Beispiel aus Abbildung 1 heran. In Tabelle 6 haben wir dazu einige der Aktionen in eine TDH einsortiert. Auch hier sehen wir wieder, dass einige Aktionen sich nicht eindeutig einordnen

Fahrzeug
/__Antrieb
/ __Motorisiert
/ Auto, Motorrad
/ __Nicht motorisiert
/ Fahrrad
/__Anzahl möglicher Passagiere
{__Ein Passagier
{ Fahrrad, Auto, Motorrad
{__Mehr als zwei Passagiere
Auto

Tabelle 5: KRG ablesbar in TDH

Spiegelei zubereiten OR
Vorbereiten
Pfanne auf Herd stellen, Ei herausholen
Zubereiten
Herd anschalten, Ei braten, Herd ausschalten
Nachbereitung
Ei auf den Teller legen, Herd ausschalten

Tabelle 6: Aktionen in einer TDH

lassen und eigentlich eine weitere Unterteilung nötig wäre. Der Unterschied zwischen HTA und einer solchen TDH besteht darin, dass HTA darauf abzielt, für eine Aufgabe eine Sequenz von Teilaufgaben zu finden, die das übergeordnete Ziel erreichen, während eine TDH wie in Tabelle 6 versucht, Gleichheit zwischen verschiedenen Aufgabentypen aufzuzeigen. Hat man nun auf einer Seite die zu einer Aufgabe gehörenden Objekte und auf der anderen Seite die zugehörigen Aktionen sortiert vorliegen, findet man oft auf den ersten Blick Beziehungen zwischen den beiden Seiten. Anhand dieser Beziehungen lassen sich schnell notwendige Umstrukturierungen erkennen. So ist es z.B. meistens leichter zu beobachten, welche Gegenstände Leute benutzen, um eine Handlung auszuführen, als die Handlung selbst zu beschreiben. Auch wenn TDH-basierte Techniken wie TAKD und die KRG für viele Anwendungen zu formal und zu komplex sind, kann das Prinzip der Taxonomien benutzt werden, um Beschreibungen und Präsentationen zu strukturieren. Sie helfen nämlich dabei, allgemeine Klassifizierungen von Aktionen und zugehörigen Objekten zu erstellen.

Entity-relationship-based techniques

Bei den Objekt- und Bezugsbasierten Methoden wird wie bei den wissensbasierten Analysen ist das Ziel eine umfassende Kategorisierung der an einer Aufgabe beteiligten Objekte und der durchgeführten Aktionen. Bei diesen werden allerdings nicht vorrangig Gemeinsamkeiten zwischen verschiedenen Objekten herausgestellt, sondern der Fokus auf die Beziehungen zwischen diesen gelegt. Dies ähnelt aus computernahen Bereichen bekannten Methoden wie z.B. der objektorientierten Programmierung, betrachtet im Gegensatz zu dieser aber auch “außerhalb” des Computers stattfindende Handlungen. Man kann auch eine solche Analyse als “objektorientiert” bezeichnen, da oftmals großer Wert darauf gelegt wird, einem “Objekt” (dies kann von einer ausführenden Person bis zu einem benutzten Gegenstand alles beschreiben) die mit ihm verbundenen Aktionen zuzuordnen. Betrachten wir als Beispiel eine Spedition und listen zunächst einmal alle beteiligten Objekte auf: Jupp, der Chef der Firma, die drei Fahrer Horst, Heinz und Holger, ihre drei LKWs, der Gabelstapler, der Paketladeroboter, drei Sackkarren und das Warenlager. Es ist klar, dass sich einige der Objekte unter

Objekt LKW1 konkret – ein LKW**Attributes**

Fahrtüchtigkeit: Ja/Nein

Baujahr: 1998

Tabelle 7: LKW als Beispielobjekt

Oberbegriffen zusammenfassen lassen können, so würde Jupp würde, wenn von seinen drei “Fahrern” reden, mit Sicherheit er über Heinz, Horst und Holger sprechen. Diese Zusammenfassung nennt man ein “composite object”. Ein weiteres Beispiel dafür wären die Fahrzeuge, worunter der Stapler und die LKWs fallen würden. Es gibt zwei weitere Arten von Objekten: “concrete objects”, also bestimmte Objekte wie ein einzelner LKW, und “actors” wie den Chef, die Fahrer aber auch der Laderoboter, wie wir später sehen werden. Als nächstes erstellt man für jedes Objekt eine Liste von Attributen, z.B. für einen LKW die Fahrtüchtigkeit und den Baujahr. Dies kann man durchaus in einer formalen Weise notieren, wie in Tabelle 7 zu sehen. Auch bei dieser Technik wird zu Beginn oft mehr Information als nötig gesammelt, die sich später als für die Analyse nicht relevant herausstellt. In unserem Beispiel könnte das Baujahr weggelassen werden, da es für den Lade- und Liefervorgang von geringem Interesse ist. Genauso wichtig wie die Objekte sind die Aktionen, die von und mit ihnen durchgeführt werden. Vier Fachbegriffe werden benutzt, um eine Handlung zu beschreiben: “agent”, “patient”, “action”, “instrument”. Zur Veranschaulichung folgender Satz: Horst (*agent*) belädt (*action*) den LKW (*patient*) mit dem Gabelstapler (*instrument*). Die Handlung lässt sich hierbei dadurch identifizieren, dass sie den Zustand eines “patients” ändert (hier ändert sich der Status des LKWs von “unbeladen” zu “beladen”) und von einem “agent” ausgeführt wird (hier Horst). Dabei ist zu beachten, dass es nicht immer ein “instrument” benutzt werden muss. Aufgabe des Analysierenden ist es herauszufinden, wer der Handelnde hinter einer Aktion ist. Dies ist insbesondere dann nicht intuitiv, wenn eine Tätigkeit im Passiv beschrieben wird, z.B.: Der LKW wurde beladen. Betrachtet man in diesem Beispiel die möglichen “agents”, zeigt sich, dass nicht nur Personen in Frage kommen, sondern auch nicht-menschliche “actors” - hier der Laderoboter. Am Beispiel des Roboters zeigt sich auch ein weiteres Problem, nämlich indirekt handelnde “agents”. So wird der Chef den Roboter so programmieren, dass dieser die LKWs bereits belädt, bevor die Fahrer ankommen. Hier stellt sich die Frage, ob nun der Chef der “agent” und der Roboter ein “instrument” oder beide “agents” sind. Dieses Problem kann man nach persönlicher Präferenz lösen, bewährt hat sich aber die Benutzung zweier Konzepte: Zum Einen die bereits erwähnten nicht-menschlichen “actors” sowie “Nachrichten”. Diese Nachrichten stellen eine eigenständige Aktion dar, in der Jupp der “agent” und der Roboter “patient” ist, worauf folgt, dass der Roboter den

Objekt Jupp menschlicher Agierender – der Chef**Aktionen: als Chef**

J1: Verwaltung

J2: Kundenpflege

Aktionen: als Lagerarbeiter

J3: LKW beladen

Tabelle 8: Rollenverteilung

Ladevorgang als “agent” ausführt. Was ist aber nun, wenn der Roboter fehlerbedingt ausfällt und die Fahrer noch nicht im Haus sind? Es ist dann sicherlich möglich, dass der Chef selbst mit Hilfe des Gabelstaplers die LKWs belädt. Offensichtlich erfüllt er dann von seinem eigentlichen Aufgabenbereich (z.B. der Verwaltung) gänzlich verschiedene Aufgaben. Dieses Konzept bezeichnet man als Rollen (*roles*). Eine mögliche Notation dafür wird dafür in Tabelle 8 gezeigt. Würde eine neue Software den Roboter fehlerfrei machen, fiel Jupps Rolle als Lagerarbeiter weg, woran man sieht, dass sich die Rollenverteilung mit der Einführung eines neuen Systems oder mit der Änderung eines bereits existenten stark verändern kann.

Zwar bedingt jede “action” per Definition eine Statusänderung, aber nicht jede Änderung ist Folge einer Aktion. Daraus folgt der Begriff “event”, mit dem alles gekennzeichnet wird, was “passiert”, ohne dass es einen erkennbaren “agent” gibt. Nehmen wir wieder das Beispiel, das der Laderoboter fehlerhaftes Verhalten zeigt und nicht mehr weiter arbeiten kann. Der Roboter ist nicht klar als “agent” zu identifizieren, da der Fehler z.B. durch äußere Einflüsse wie Blitzschlag verursacht worden sein kann. Auch zu einer bestimmten Zeit oder nach Ablauf einer Zeitspanne passierende Ereignisse werden hier als “events” bezeichnet. Da wir nun alle grundlegenden Begriffe eingeführt und erläutert haben, kommen wir nun zu dem eigentlichen Ziel einer “entity-relationship-based analysis”, den Beziehungen zwischen Objekten. Die einfachsten Beziehungen sind die zwischen zwei Objekten, so ist der

Events

Ev1: Lagerroboter fällt fehlerbedingt aus

Relations: object - object

location (Lagerroboter, Warenlager)

Relations: action - object

patient (J3, LKW1) – Jupp belädt den ersten LKW

instrument (J3, Gabelstapler) – .. mit dem Stapler

Relations: action - event

triggers (Ev1, J3)

Tabelle 9: Ereignisse und Beziehungen

0. Den LKW beladen

1. Der Roboter belädt den LKW
2. Der Chef belädt den LKW selbst
 - 2.1 Er holt den Gabelstapler
 - 2.2 Er belädt den LKW

Plan 0: führe 1 – falls der Roboter ausfällt 2 in dieser Reihenfolge aus

Plan 2: führe 2.1 – 2.2 in dieser Reihenfolge aus

Tabelle 10: HTA angewendet auf Liste von Objekten und Beziehungen

Chef in unserem Speditionsbeispiel natürlich den Fahrern übergeordnet und der Roboter steht im Warenlager. Interessanter ist es nun, die Beziehungen zwischen Aktionen und Objekten zu betrachten, wobei die zwischen “agent” und durchgeführter “action” in der vorgestellten Notation implizit ist, da bei einem “actor” auch die von ihm durchgeführten Aktionen aufgelistet sind. Es macht in der Regel allerdings Sinn, die bei einer Aktion explizit veränderten “patients” sowie die benutzten “instruments” aufzulisten. Ebenso sind Beziehungen zwischen “events” und “actions” wichtig, da sie zeitliche Zusammenhänge und Kausalitäten verdeutlichen. Auf unser Beispiel angewandt könnte dies wie in Tabelle 9 aussehen (aus Gründen der Übersichtlichkeit wurde auf eine vollständige Liste aller Objekte verzichtet).

Eine solche Analyse der Beziehungen kann ihre volle Wirkung besonders in Zusammenarbeit mit anderen Methoden entwickeln. Wir können z.B. eine HTA auf eine vollständige Liste von Objekten, Aktionen, Ereignissen und Beziehungen anwenden, um Teilaufgaben nach ihrer zeitlichen Abfolge zu sortieren. Betrachten wir Tabelle 10, die dies am bereits bekannten Beispiel des ausfallenden Lagerroboters aufzeigt. Hier entspricht die Bedingung, dass bei Ausfall des Roboters Teilaufgabe 2 ausgeführt wird, genau dem Ereignis Ev1 in Tabelle 9, das dann die Aktion J3 auslöst. Diese Handlung ist dann durch die Teilaufgabe 2 repräsentiert. So lassen sich Kausalitäten und zeitliche Abfolgen schneller erkennen, als in der zur Auflistung der Objekte etc. verwendeten Notation.

Aber auch bei der Erstellung von Taxonomien wie sie bei den wissensbasierten Analysemethoden vorgestellt wurden kann eine Objekt- und Bezugsbasierte Analyse helfen: Listet man z.B. zunächst für zwei Objekte alle zu jeweils einem in Bezug stehenden Aktionen auf und vergleicht die Ergebnisse, werden Gemeinsamkeiten und Unterschiede oft auf den ersten Blick sichtbar – was die Klassifizierung vereinfacht.

Modellierung von Aufgaben für kontextsensitive Interfaces

Die drei bisher vorgestellten Methoden sind nicht geeignet,

Aufgaben in einem sich verändernden Umfeld zu modellieren. Eine Änderung des Kontexts erfordert hier eine aufwändige Anpassung der gesamten Analyse, oftmals vom ersten Schritt an. Ein Modell, das kontextsensitiv und besonders geeignet zum Erstellen von “user interfaces” ist, wurde 2001 vorgestellt [2]. Dieses soll hier kurz zusammengefasst werden. Definieren wir zunächst einmal den Begriff “Kontext”: Als Kontext wird hier die gesamte Umgebung bezeichnet, in der der Nutzer eine Aufgabe ausführt. Hierbei kann man hauptsächlich zwischen zwei Arten von Eigenschaften unterscheiden: Interne, also das Interface und das dahinterliegende System betreffende, sowie Externe, worunter Dinge fallen wie das Vorwissen des Nutzers, Umgebungsbedingungen, Stress etc. Jede Veränderung an mindestens einer dieser Eigenschaften bedeutet eine Änderung des Kontexts. Das Ziel ist nun, ein Interface zu schaffen, das sich solchen Veränderung anpasst und seine Funktionalität beibehält. Vier Schritte wurden identifiziert, die nötig sind um solch ein Modell zu entwickeln:

1. Entwicklung eines kontextsensitiven Gesamtmodells. Dieses sollte im Idealfall zwischen kontextsensitiven Teilaufgaben und nicht kontextsensitiven Teilen unterscheiden. Daraus folgen zwei weitere Punkte.
2. Design eines allgemeinen Interfacemodells, das den nicht-kontextsensitiven Teil der Nutzeroberfläche modelliert.
3. Design eines spezifischen Interfacemodells, das sich dem veränderlichen Kontext anpasst. Es kann in Abhängigkeit vom Kontext nötig sein, mehrere solche Modelle zu entwickeln, um alle möglichen Veränderungen abzufangen.
4. Produktion der lauffähigen, finalen Oberfläche. Dies kann z.B. durch automatische Auswertung des vorher entwickelten Modells geschehen.

Als Grundlage für ein solches Modell wurde die “ConcurTaskTree” Notation (kurz: CTT) gewählt, da diese geeignet ist um zeitliche Abfolgen und Hierarchien von Aufgaben in Bäumen darzustellen. Das endgültige Modell enthält einen nicht-kontextsensitiven Teil, modelliert in CTT-Notation, einen Entscheidungsbaum und einen kontextsensitiven Teil, der als eine Folge von Teilbäumen wieder in CTT notiert ist. Der Entscheidungsbaum verbindet verschiedene allgemeine Kontexte auf der ersten Ebene sequentiell wie optionale Teilaufgaben mit dem Sequenz-Operator “>>” aus CTTE. Auf der nächsttieferen Ebene werden dann die verschiedenen Bedingungen in einem Kontext durch den Operator “[]” gekennzeichnet, der einen gegenseitigen Ausschluss bezeichnet. Die Blätter des Entscheidungsbaums entsprechen den Wurzeln der kontextsensitiven Teilbäume. Veranschaulicht ist dies in Abbildung 2. Um zu verhindern, dass Teilaufgaben mehrfach

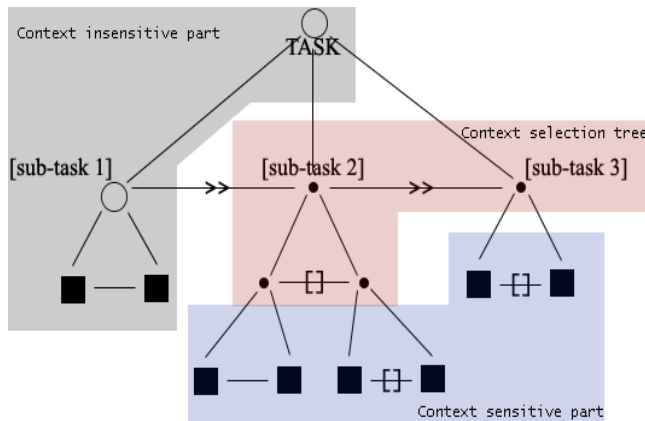


Abbildung 2: Kontextsensitive Modellierung

auftauchen, sucht man ausgehend von der untersten Ebene iterativ nach Gemeinsamkeiten. Sind zwei Teilaufgaben unter verschiedenen Kontextbedingungen identisch, kann man sie in die nächsthöhere Ebene als neue Aufgabe einfügen, unter Umständen kann sie sogar dadurch komplett nicht-kontextfrei werden. Diese Art der Modellierung kann aufgrund ihrer Formalität genutzt werden um automatisch eine Nutzeroberfläche entwerfen. Hierzu muss man allerdings noch einige zusätzliche Informationen eintragen, wie z.B. an jedem Entscheidungsknoten alle Werte, die dieser Knoten einnehmen kann. Allerdings ist auch diese Technik noch nicht optimal zur Analyse einer Aufgabe unter Einbeziehungen ihres Kontexts: dynamische Änderungen, wie sie während der Erledigung einer Aufgabe entstehen können, werden auch in diesem Modell nicht modelliert.

INFORMATIONSQLUELLEN UND SAMMLUNG VON DATEN

Allgemein

Wie jede wissenschaftliche Analyse beruht auch die "task analysis" auf den zu Grunde liegenden Daten. Dabei ist die Qualität der Daten ausschlaggebend für die Qualität des Ergebnisses, weshalb man auf dem Ziel angepasste Erhebungsmethoden zurückgreifen sollte. Die Wichtigsten dieser Methoden werden wir im Folgenden kurz vorstellen. Dabei sollte im Hinterkopf behalten werden, dass der Prozess einer "task analysis" selten linear ist, sondern aus vielen Iterationsschritten besteht. In diesen Schritten sollten sich Analyse und Datensammlung im Idealfall abwechseln, da sowohl neue Daten, als auch Zwischenergebnisse oft neue Fragen aufwerfen. Dies ist jedoch aus Kosten- und Zeitgründen selten möglich. Daher versucht ein geschickter Analysierender den Prozess durch Nutzung billiger und leicht zugänglicher Quellen zu optimieren.

Informationsquellen

Dokumentationen

Da sich die "task analysis" meistens mit bereits

existierenden Systemen auseinandersetzt, stellen die zugänglichsten und kostengünstigsten Informationsquellen bereits vorhandene Handbücher, Beschreibungen und Lehrmaterial für die zu untersuchende Aufgabe dar. Für allgemeinere Beschreibungen kann man übergeordnete Quellen nutzen. Als Beispiel seien hier Berufsbeschreibungen genannt. Eine Diskrepanz findet sich dabei häufig zwischen den Beschreibungen, wie eine Aufgabe erledigt werden sollte und wie sie tatsächlich bearbeitet wird. Dieser Unterschied kann durch Beobachtung des entsprechenden Handlungsablaufs erkannt werden. Auf diese Technik gehen wir jedoch später noch detaillierter ein. Ein weiteres Problem erkennt man an folgendem Beispiel: Ein Handbuch für eine Schlagbohrmaschine könnte z.B. detailliert die Einstellmöglichkeiten und Wartung des Bohrers beschreiben. Die korrekte Handhabung desselben, die für ein optimales Ergebnis erforderlich ist, könnte jedoch mit keinem Wort erwähnt werden. Diese Problematik schränkt allerdings den Nutzen des Handbuchs für die Analyse nicht völlig ein, man kann diesem immer noch einige grundlegende Tätigkeiten und beteiligte Objekte entnehmen, wie z.B. das Wechseln des Bohrkopfs.

Beobachtung

Führen wir das Beispiel der Schlagbohrmaschine weiter: Wie erkennt man nun die tatsächlich durchzuführenden Tätigkeiten? Hier bietet sich die direkte Beobachtung eines Nutzers an, der im Idealfall mit der Aufgabe vertraut ist. Bevor man einen komplexen Versuchsaufbau entwirft, lohnt es sich meistens, einen ersten Eindruck für die Situation zu bekommen, in dem man eine gewisse Zeit dafür aufbringt, mit den beteiligten Personen zu sprechen und sie unvoreingenommen bei den zu analysierenden Aufgaben beobachtet. Zur Erstellung eines Versuchsaufbaus kann man einschlägige Literatur aus der Psychologie heranziehen, hier möchten wir nur kurz einige zu beachtende Punkte anschnitten. Obwohl die Bedingungen im Labor besser sind, eignet sich die Beobachtung im "Feld" für die "task analysis" oft besser, da die Handlungsabläufe dort unverfälscht sind. Man unterscheidet primär zwischen passiver und aktiver Beobachtung: die Passive findet selbstverständlich ohne Beeinflussung statt, bei der Aktiven kann man z.B. Zwischenfragen stellen. Ein "post-task walkthrough" bietet sich bei der passiven Beobachtung an, bei diesem bespricht man die einzelnen Schritte mit der handelnden Person nach Ende der Aufgabe. Welche Fragen man dabei stellen sollte werden wir im nächsten Abschnitt erläutern.

Interviews

Als erster Ansatzpunkt für Fragen bei einem "post-task walkthrough" bieten sich offensichtliche Unterschiede zwischen Beschreibung einer Aufgabe und der tatsächlichen Ausführung an, z.B. welche Gründe die

Abweichung hat etc. Man kann den “walkthrough” auch mit einem “Experten” durchführen, der nicht die beobachtete Person war. Dies kann weitere Blickwinkel auf die Aufgabe und ihre Erledigung eröffnen, indem man die Erwartung des Dritten mit den gezeigten Aktionen des Beobachteten vergleicht. Einen solchen Experten (hiermit kann jeder gemeint sein, der ein Wissen über den beobachteten Aufgabenbereich hat) kann man allerdings auch außerhalb einer Beobachtungssituation befragen. Dabei startet man z.B. mit allgemeinen Fragen zum Handlungsablauf, um zu konkreteren Fragestellungen überzugehen. Eine Ausnahme zu dieser Abfolge stellt HTA dar. Aufgrund seiner Struktur ist es sinnvoll dort den “top-down-Ansatz” zu verfolgen und von einer kurzen, präzisen Aussage auszugehen und diese schrittweise zu erweitern. Das Interview eines Experten ist oft eine kostengünstige und effektive Methode um sich in kurzer Zeit einen Überblick über die Aufgabe, mögliche Probleme etc. zu verschaffen.

Sortieren und Klassifizieren

Wir haben bereits einige Techniken zur Sortierung ein Klassifizierung von Objekten, Aufgaben etc. kennengelernt, z.B. die Taxonomien. Doch wie kann man die gesammelten Daten geeignet vorsortieren, so dass man überflüssige Informationen aussortieren und wichtige Daten direkt kategorisieren kann? Eine nützliche Technik ist es, die Objekte und Aktionen auf einzelne Zettel zu schreiben und einem oder mehreren Experten vorzulegen. Diese werden dann gebeten, die Zettel zu sortieren, nach vorgegebenen Kriterien wie z.B. ähnliche Aufgaben auf einen Stapel und die zugehörigen Objekte direkt daneben. Führt man dies mehrfach durch, erhält man einen ersten Überblick über die Struktur der Aufgabe. Es gibt viele ähnliche Techniken, z.B. die Wichtigkeit verschiedener Aufgaben gegeneinander abschätzen zu lassen, sie führen jedoch alle zu ähnlichen Ergebnissen.

ANWENDUNGSGEBIETE

Allgemein

Das Resultat einer “task analysis” kann auf verschiedene Weisen genutzt werden. Daher sollte man sich bei der Wahl der Analysemethode an der geplanten Anwendung orientiert haben. Liegt nun das Ergebnis in geeigneter Form vor, ergeben sich drei hauptsächliche Einsatzgebiete, wobei nur das erste dieser auf das eigentliche System abzielt. Die anderen beiden beschäftigen sich mit der Entwicklung neuer Systeme.

Handbücher und Lehrmaterial

Die zu Beginn vorgestellte HTA eignet sich aufgrund ihrer Struktur gut zum Planen von Unterrichtsmaterial und Ähnlichem. So ließe sich z.B. aus der in Abbildung 1 gezeigten HTA schnell eine Schritt-für-Schritt Anleitung zur Zubereitung eines Spiegeleis erstellen, wie man sie

auch in Kochbüchern finden könnte. Auch wenn sich diese schrittweise Beschreibung gut eignet, um den Arbeitsablauf einem Anfänger nahe zu bringen, stößt sie bei komplexen Aufgabenstellungen an ihre Grenzen. Die wissensbasierten Ansätze hingegen sind ideal für solche Situationen, da sie geeignet sind, um das nötige Vorwissen zu analysieren. Da ähnliche Aufgaben ähnliches Wissen voraussetzen, kann man Trainingsmaterial gut anhand einer z.B. in TDH erstellten Taxonomie aufbauen.

Aber auch beim Transfer zwischen zwei unterschiedlichen Systemen zeigt sich die Stärke der wissensbasierten Techniken: Sie erlauben schnelle Vergleiche auf Gemeinsamkeiten und Unterschiede, so dass der Trainingsaufwand beim Umstieg leicht zu erkennen ist.

Erfassung von Anforderungen und Systementwicklung

Im eigentlichen Sinne ist “task analysis” nicht geeignet, um Anforderungen an neue Systeme zu erfassen, da sie sich in der Regel mit einem existierenden System beschäftigt und viele Elemente, die nicht zum eigentlichen System gehören, mit einbezieht. Der Beitrag der Analyse von Aufgaben zu diesem Bereich besteht darin, dass sie erfasst, *wie* ein System benutzt wird, während traditionelle Anforderungen oft so formuliert sind, dass sie nur danach fragen, *was* das gesuchte System können muss. Taxonomien, die in TDH verfasst sind, können dann z.B. im späteren Entwicklungsstadium helfen, das System intern entsprechend den externen Erwartungen des Nutzers zu gestalten. Hat die Analyse z.B. ergeben, dass die späteren Nutzer des Systems zwei bestimmte Funktionen als sehr ähnlich und zusammengehörig betrachten, sollte diese Struktur auch intern wiedergespiegelt werden. Weiterhin kann man mit “task analysis” in diesem Stadium Voraussagen treffen, wie das System später genutzt wird. So kann es z.B. sein, dass bisher gültige Arbeitsschritte in einer neuen Version eines Systems überflüssig werden. Eine entsprechende Analyse kann dies frühzeitig aufzeigen und entsprechendes Trainingsmaterial vorbereitet werden.

Entwicklung von Interfaces

Die Nützlichkeit von TDH-basierten Taxonomien bei der Konzipierung von Menüs ist auf den ersten Blick ersichtlich: Jede Ebene im TDH-Baum entspricht einer Ebene im Menü. Es ist hier sicherlich sinnvoll, den Baum zuerst ausschließlich mit strikten Entweder/Oder-Unterscheidungen aufzubauen, damit nur genau ein Pfad zu jedem Menübefehl führt. Dies lässt sich wiederum gut mit der vorgestellten “knowledge representation grammar” überprüfen.

Aber auch die “entity-relationship-basierten” Ansätze haben ihre Anwendungsmöglichkeiten im Bereich des Interfacedesign. So könnte man ein Layout an der Rollenverteilung orientieren, indem man als erste Menuebene für jedes Objekt die verschiedenen Rollen nimmt und die nächste Ebene aus den zu einer Rolle

gehörigen Aktionen bildet.

Wir haben auch gesehen, dass man mit einer solchen Analyse schnell alle Aktionen aufzählen kann, die mit einem Objekt in Bezug stehen, und zwar sowohl die Aktionen, bei denen das Objekt "agent" ist, als auch die, bei denen es "patient" ist. Ein darauf basierendes Menu könnte eben genau diese Unterscheidung übernehmen.

Wird eine bestimmte Abfolge von Aktionen immer wieder vom Nutzer ausgeführt, kann man die HTA nutzen, um das zugehörige Menu entsprechend der Abfolge der Teilaufgaben zu konstruieren. Dies ist allerdings ein systemnaher Ansatz und sollte nur verwendet werden, wenn dem Nutzer die Aufgabe bekannt ist.

Anwendungsbeispiele

Beispielhafte Anwendung einer "task analysis"

In einer 2005 veröffentlichten Arbeit wurde untersucht, wie Informationen bei der Erfüllung einer Aufgabe genutzt werden. Dazu wurden 20 Bioinformatiker bei einer sehr speziellen Tätigkeit beobachtet: Sie untersuchten, wie man anhand der über ein Gen gewonnenen Daten vorhersagen kann welche Funktion dieses wahrscheinlich ausübt. Die Daten wurden hauptsächlich durch Interviews mit den 20 Probanden gesammelt, um ein Modell für den Prozess zu entwickeln, den die Wissenschaftler verfolgten um zu ihrem Ziel zu kommen. In jedem Interview begann man damit, sich den von dieser Person verfolgten Ansatz beschreiben zu lassen und stellte dabei mit gezielten Nachfragen sicher, dass alle wichtigen Punkte beachtet wurden. Dabei wurde besonders Wert darauf gelegt, zu erfahren welche Informationen vor Beginn der Genanalyse zur Verfügung standen, welche im Verlauf der Analyse hinzukamen und wie diese Informationen verwendet wurden. Durch insgesamt sechs Iterationsschritte wurden dann die gesammelten Daten zu einem Modell des Prozesses zusammengeführt, das das Vorgehen der Wissenschaftler als eine Abfolge von Aufgaben und Teilaufgaben beschrieb. Einer weitere Analyse der Daten nutzte einen wissensbasierten Ansatz um zu modellieren, *warum* einzelne Schritte auf eine bestimmte Art und Weise durchgeführt wurden. Die Ergebnisse dieser Methode wurden dann mit der Analyse der Teilaufgaben zusammengeführt, um für einzelne Schritte genau zu bestimmen, welche Informationen als Input in den Schritt einfließen, welchen Output der Vorgang hatte und welche Interpretationen und Widersprüche auf Basis der gesammelten Informationen gefunden wurden. Als Ergebnis der gesamten Analyse fanden sich drei Kategorien, unter denen sich der Nutzen von Information in einer solchen wissenschaftlichen Umgebung zusammenfassen ließ:

2. Interpretation: Die gewonne Information wird genutzt, um ein Phänomen zu verstehen
3. Richtung: Die Information wurde benutzt, um eine Entscheidung zu fällen.

Dieses Beispiel zeigt, wie eine Kombination verschiedener Methoden der "task analysis" genutzt werden kann, um zu modellieren, wie Informationen und Wissen in einer wissenschaftlichen Umgebung genutzt werden.

MTA - ein Werkzeug zur automatischen Analyse von Wartungstätigkeiten

Um den aktuellen Stand der Forschung im Bereich der automatischen "task analysis" aufzuzeigen, möchten wir nun kurz beispielhaft ein 2005 vorgestelltes Werkzeug zur Analyse von Wartungstätigkeiten vorstellen. Dieses wurde entwickelt, um die Effizienz von Entwicklung, Design und Wartung zu steigern. Dazu ist es nötig, die Zusammenarbeit und den Informationsaustausch zwischen den verschiedenen zuständigen Abteilungen zu optimieren. Hierzu werden von diesen alle relevanten Daten in eine gemeinsame Datenbank eingetragen. So tragen z.B. Die Techniker ein, wie zeitaufwändig die Reparatur häufig ausfallender Teile ist, während auf der anderen Seite beispielsweise die Entwickler in die Datenbank einpflegen, welche Änderung sie in zukünftigen Versionen planen. Da alle Beteiligten Zugriff auf die Datenbank haben, fließen Information verzögerungslos zwischen den Abteilungen und ermögliche eine größtmögliche Abstimmung. Die Datenbank stellt damit eine dynamische Repräsentation des Lebenszyklus' von der Entwicklung bis zur Wartung dar. MTA stellt für die Bearbeitung und Nutzung der Datenbank eine grafische Oberfläche zur Verfügung. Weiterhin enthält das Werkzeug enthält ein Modul zu automatischen Erstellung von Arbeitsabläufen und dazugehörigem Illustrationsmaterial.

ZUSAMMENFASSUNG

Mit dieser Ausarbeitung haben wir einen ersten Einblick in das Gebiet der Analyse und Modellierung von Aufgaben gegeben. Es wurden einige grundlegende Techniken eingeführt um die verschiedenen Ansätze (hierarchisch, wissensbasiert, objekt- und bezugsbasiert sowie kontextsensitiv) aufzuzeigen, die man verfolgen kann, um die Bewältigung einer Aufgabe zu untersuchen. Weiterhin haben wir erläutert, auf welche Art und Weise man die einer Analyse zu Grunde liegenden Daten sammeln und auswerten kann. Abschließend haben wir gezeigt, für welche Anwendungen man eine "task analysis" nutzen kann und dies an einigen Beispielen verdeutlicht.

1. Input: Information wird in einem anderen Schritt verwendet

QUELLENVERZEICHNIS

1. A. Dix, J. Finlay, G. D. Abowd, R. Beale: *Human Computer Interaction*, Prentice Hall, 3. Auflage, 2003
2. Wise, G.B. Lizzi, J.M. Hoebel, L.J.: *MTA - a tool for automated task analysis and lifecycle support*, Reliability and Maintainability Symposium, 2005. Proceedings. Annual, IEEE Press, 61- 66.
3. Costin Pribeanu, Quentin Limbourg, Jean Vanderdonckt: *Task Modelling for Context-Sensitive User Interfaces*, Lecture Notes in Computer Science. Volume 2220/2001, Springer Berlin / Heidelberg, 49 – 67
4. Joan C. Bartlett, Elaine G. Toms, *How is Information used? Applying task analysis to understanding information use*, 2005, http://www.cais-acsi.ca/proceedings/2005/bartlett_2005.pdf