Task analysis

Analyse und Modellierung von Aufgaben

Gliederung

- Einführung
- Techniken zur Analyse und Modellierung
  - task decomposition
  - knowledge-based techniques
  - entity-relationship-based techniques
- Informationsquellen und Sammlung von Daten
- Anwendungsgebiete
- Zusammenfassung

Einführung

- Analyse und Modellierung von Aufgaben
- verschiedene Methoden
- hilft das System aus der Sicht des Nutzers zu betrachten

Unterschiede zu anderen Techniken

- setzt existierendes System voraus
- Fokus:
  - „WAS“ macht der Benutzer?
  - „WIE“ macht er es?

Task decomposition

- Zerlegung in Teilaufgaben
- Modell entspricht menschlicher Vorgehensweise
- Beispiel: *hierarchical task analysis* (*HTA*)

Hierarchical task analysis

Ergebnis:
- hierarchische Anordnung von Aufgaben und Unteraufgaben
- „Pläne“ beschreiben Abfolge der Teilaufgaben

Beispiel: Zubereitung eines Spiegeleis

Beispiel: Zubereitung eines Spiegeleis

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

Beispiel: Zubereitung eines Spiegeleis

- für Anfänger zu grobe Aufteilung
- Konkretisierung notwendig

➡ weitere Ebene einführen

Beispiel: Zubereitung eines Spiegeleis

0. Spiegelei zubereiten
    1. Pfanne und Herd vorbereiten
        1.1 Pfanne aus dem Schrank holen und auf den Herd stellen
        1.2 Herd anschalten
        1.3 Warten bis die Pfanne heiß ist
    2. Spiegelei braten
        2.1 Ei aus dem Kühlschrank holen und in die Pfanne schlagen
        2.2 Warten bis das Ei durch ist
        2.3 Zweite Seite des Eis
            2.3.1 Ei wenden
            2.3.2 Warten bis die zweite Seite auch durch ist
    3. Ei auf den Teller legen
    4. Herd ausschalten

Plan 0: führe 1-2-3-4 in dieser Reihenfolge durch
Plan 1: führe 1.1-1.2-1.3 in dieser Reihenfolge durch
Plan 2: führe 2.1-2.2-falls gewünscht 2.3 in dieser Reihenfolge durch
Plan 2.3: führe 2.3.1-2.3.2 in dieser Reihenfolge durch

Strukturen in einer *HTA*

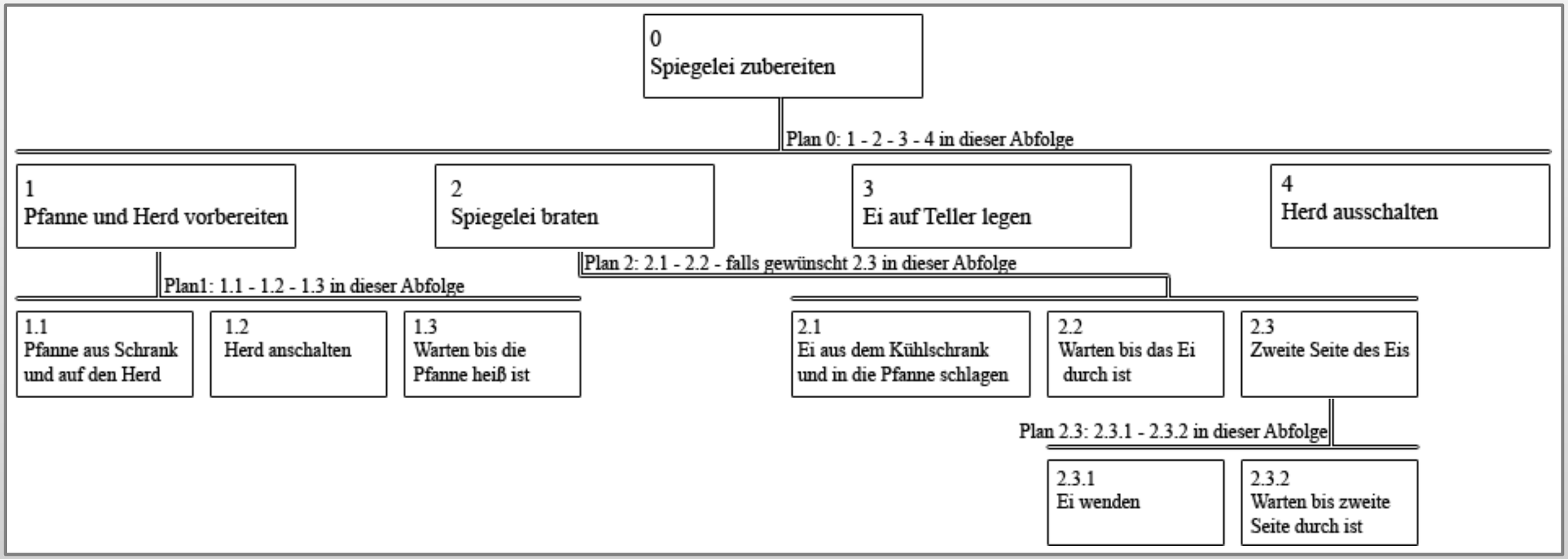
lineare und nicht lineare Abfolge:

- *„busy waiting“* und *„idle waiting“*
- bedingte Ausführung
- Schleifen

Eigenschaften einer *HTA*

- Ergebnis abhängig von:
  - Sichtweise des Analysten
  - Ziel der Analyse
- Vorgang ist iterativ
- „P x C – Regel“

HTA als Baumdiagramm



Probleme bei der *HTA*

- komplexe „interne “ Entscheidungen können nicht modelliert werden
- Prozess trotz Iteration nicht geradlinig
- angemessene Detailtiefe schwer zu bestimmen

Knowledge-based techniques

- Ziel: Verständnis für das zur Lösung einer Aufgabe nötige Wissen
- Ergebnis: „Taxonomie“
- Taxonomie: Klassifizierung der an der Aufgabe beteiligten Aktionen und Objekte

Task analysis for knowledge description

- *TAKD* benutzt „task descriptive hierarchy (*TDH*)“

- Sortierung bzgl. verschiedener Eigenschaften

- 3 Arten von „Zweigen“:
  - *XOR*
  - *AND*
  - *OR*

Beispiel: *TAKD* mit Hilfe von *TDH*

Fahrzeug ***AND***
- Antrieb ***XOR***
  - motorisiert
    - Auto, Motorrad
  - nicht motorisiert
    - Fahrrad
- Anzahl möglicher Passagier ***OR***
  - ein Passagier
    - Fahrrad, Auto, Motorrad
  - mehr als 2 Passagiere
    - Auto

Knowledge representation grammar

- *TAKD* mit alternativer Notation in *KRG*
- Ersetzen von „Zweig“-Bezeichnungen:
  - *AND* ➡ */*
  - *XOR* ➡ |
  - *OR* ➡ {
- ermöglicht „Satzbildung“ in *KRG*:
  Fahrzeug/Antrieb(nicht motorisiert)/Anzahl möglicher Passagier{ein Passagier}/ = Fahrrad
- Sätze einfach auf Eindeutigkeit zu prüfen

Beispiel: *TAKD* mit Hilfe von *KRG*

```
Fahrzeug
/___ Antrieb
/    |___ motorisiert
/    |          Auto, Motorrad
/    |___ nicht motorisiert
/              Fahrrad
/___ Anzahl möglicher Passagier
     {___ ein Passagier
     {          Fahrrad, Auto, Motorrad
     {___ mehr als 2 Passagiere
                Auto
```

Task analysis for knowledge description

- Aktionen können wie Objekte klassifiziert werden
- Aber: *TAKD* und *KRG* sehr formal und komplex ➡ selten angewendet
- Taxonomien trotzdem wichtig
- Gemeinsamkeiten und Unterschiede werden sichtbar

Entity-relationship-based techniques

- Ebenfalls Kategorisierung von Aktionen und Objekten

- Fokus liegt jedoch auf Beziehungen zwischen diesen

- ähnliches Konzept wie in der objektorientierten Programmierung
  - z.B. Vererbung zwischen Klassen

Objekte

- Arten von Objekten:
  - *„concrete objects“*
  - *„composite objects“*
  - *„actors“*
- Attribute

Beispiel: Spedition

Objektart	Objektname
concrete objects	LKW (3x), Sackkarren (3x), Gabelstapler (1x), Warenlager (1x)
composite objects	Fahrer (Heinz, Holger, Horst), Fahrzeuge (LKWs, Stapler)
actors	Jupp (Chef), Heinz, Holger, Horst, Paketladeroboter

Object LKW1 **concrete** – *ein LKW*

Attributes

Fahrtüchtigkeit: Ja/Nein

Baujahr: 1998

Aktionen

Beispiel: Horst belädt den LKW mit dem Gabelstapler

- „*agent*“: Horst
- „*action*“: beladen
- „*patient*“: LKW
- „*instrument*“: Gabelstapler

- Handlung identifizierbar an:
  - wird von *agent* ausgeführt
  - ändert Zustand des *patients*

Rollen und Ereignisse

- *actors* können verschiedene Rollen einnehmen
- jede Rolle hat eigene Aktionen
- Ereignisse: jede Attributänderung, die keinen klar erkennbaren *agent* hat

Beispiel: Spedition

Object Jupp **human actor** – *Chef der Spedition*

Actions: as Chef

J1: Verwaltung

J2: Kundenpflege

Actions: as Lagerarbeiter

J3: LKW beladen

Beziehungen

Arten von Beziehungen:

- *object – object*
- *action – object:*
  - Beziehung zwischen *agent* und *action* implizit
  - sinnvoll: sowohl *patients* als auch *instruments* auflisten
- *action – event:* verdeutlichen zeitliche und kausale Zusammenhänge

Beispiel: Spedition

Events

Ev1: Lagerroboter fällt fehlerbedingt aus

Relations: object – object

location (Lagerroboter, Warenlager)

Relations: action – object

patient(J3, LKW1) – Jupp belädt den LKW

instrument(J3, Gabelstapler) - …mit dem Stapler

Relations: action – event

triggers(Ev1,J3)

Informationsquellen und Datensammlung

- wissenschaftliche Analyse beruht auf Daten
- Qualität der Daten bedingt Qualität der Analyse
- *task analysis* meist iterativ
- ideal: Datensammlung und Analyse wechseln sich ab
- Ziel: Kompromiss zwischen Kosten und Qualität der Daten

Mögliche Quellen

- Dokumentationen
- Beobachtungen
- Befragungen

Dokumentationen

- Vorteile
  - oft leicht zugänglich und kostengünstig
  - guter erster Überblick
- Nachteile:
  - Diskrepanz zwischen Aufgaben-beschreibung und tatsächlicher Ausführung
  - Beschreibung der Funktionen aber nicht der Benutzung

Beobachtung

- Idealfall:
  - erster Eindruck durch „Mitarbeit“
  - Experten beobachten
  - unter reellen Bedingungen
- Arten:
  - aktiv – Zwischenfragen etc.
  - passiv – z.B. *„post-task walkthrough“*

Befragung

- *walkthrough:*
  - Unterschiede zwischen Aufgabenbeschreibung und Ausführung
  - Durchführung mit Drittem
- Expertenbefragung:
  - von allgemeinen Fragen zu konkreteren übergehen
  - Ausnahme: *HTA*

➡ ist oft kostengünstig und effektiv

Sortierung und Klassifizierung

- gesammelte Daten Experten vorlegen
- Sortierung durch diese nach vorgegeben Kriterien
- viele Methoden führen zu ähnlichen Ergebnissen

➡ nach mehreren Durchgängen erhält man guten Überblick über Struktur der Aufgabe

Anwendungsgebiete

- Handbücher und Lehrmaterial
- Erfassung von Anforderungen und Systementwicklung
- Entwicklung von Interfaces

Handbücher und Lehrmaterial

- *HTA* eignet sich zur Strukturierung von einfachem Unterrichtsmaterial
- für komplexere Aufgaben: *knowledge-based techniques*
  - erlauben schnellen Vergleich zweier Systeme

➡ Trainingsaufwand bei Systemtransfer abschätzen

Anforderungen erfassen und Systemdesign

- erfasst **wie** System benutzt wird

- Taxonomien ermöglichen es das System intern nach den externen Erwartungen des Nutzers zu modellieren

- Vorhersagen über Nutzung des Systems möglich

Entwicklung von Interfaces

- *TDH*-basierte Taxonomie kann direkt in Menüstruktur umgewandelt werden
- Orientierung des Menüs aber auch an Rollenverteilung möglich
- Aktionen auf einem Objekt sortierbar nach seiner Funktion („*agent*“, „*patient*“)
- Menü entsprechend einer Abfolge von Aktionen aufbauen (*HTA*)

Anwendungsbeispiel: *MTA*

- Abkürzung für: *„maintenance task analysis“*
- Werkzeug zur automatischen Analyse von Wartungstätigkeiten
- Datenbank mit Inhalten aus dem gesamten Lebenszyklus eines Produktes
- hilft bei automatischer Erstellung von Dokumentation und Arbeitsabläufen

Zusammenfassung

- Analysetechniken:
  - *task decomposition*: *HTA*
  - *knowledge-based*: *TAKD* mit *TDH* und *KRG*
  - *entity-relationship-based*
- Informationsquellen:
  - Dokumentation
  - Beobachtung
  - Befragung

Zusammenfassung

- Anwendungsgebiete:
  - Lehr- und Trainingsmaterial
  - Systemdesign
  - Interfaceentwicklung
- Anwendungsbeispiel: *MTA*

Ende

Vielen Dank für die Aufmerksamkeit!