

# Evaluation von multimodalen Systemen: TYCOON und CARE

Daniel Naber <dnaber@techfak.uni-bielefeld.de>, 2000-11-23

Durch die verstärkte Forschung im Bereich MMMK (multimodale Mensch-Maschine-Kommunikation) existieren viele Systeme, die untereinander nur schwer zu vergleichen sind. Mit TYCOON und CARE existieren Mittel, die bei Konzeption, Entwicklung und Bewertung von solchen Systemen helfen sollen. Beide Systeme sind sich sehr ähnlich, sie bestehen aus einem theoretischen Framework und einer formalen Schreibweise für Merkmale von multimodalen Systemen. Beide Systeme wurden mit real existierender Software "getestet", d.h. die Merkmale neu entwickelter Systeme wurden formal erfasst.

## 0.1 Ziele aller multimodalen Systeme

- Die Entwickler von TYCOON nennen folgende Ziele: hohe Erkennungsrate, schnelle Interaktion, exakte Interpretation, intuitive Bedienbarkeit, leichte Erlernbarkeit, Anpassungsfähigkeit

## 0.2 CARE-Properties: Relationen zwischen Modalitäten

- CARE wurde entwickelt ca. 1995 von Coutaz/Nigay/Salber am IMAG, Frankreich
- bezieht sich auf Ein- und Ausgabe; beschreibt Relationen zwischen Modalitäten
- Die CARE Properties (Eigenschaften) sind der Kern der Idee:
- *Äquivalenz* - zwei Modalitäten sind bezüglich eines Ziels äquivalent, wenn entweder die eine oder die andere zum Erreichen des Ziels benutzt werden kann
- *Redundanz* - zwei Modalitäten können (zeitgleich oder zeitversetzt) die selbe Information spezifizieren, das multimodale System erkennt

die Redundanz, kann also mit überqualifizierten Eingaben umgehen

- *Komplementarität* - zwei verschiedene Modalitäten werden ergänzend benutzt, um ein Ziel zu erreichen (typisches Beispiel: "Put that there": Zeigegeste ergänzt Sprache)
- *Zuweisung* - ein Ziel kann nur mit einer bestimmten Modalität erreicht werden
- Bis auf Zuweisung steigern alle diese Eigenschaften prinzipiell die Benutzbarkeit eines Systems, ihre Unterstützung ist also als positiv anzusehen. Zuweisung kann beim Output sinnvoll sein (z.B. Fehler werden immer durch Piepen gemeldet)
- Beispiel, formale Definition von *Äquivalenz*:  
 $\text{äquivalent}(M, g) \Rightarrow (\forall m \in M \text{ reach}(m, g)) \wedge (|M| > 1)$   
"äquivalente Modalitäten bezüglich eines Zielzustandes"

## 0.3 CARE: User-Preferences

Die Entwickler von CARE legen grossen Wert darauf, die Eigenschaften auch aus Sicht des Users zu beschreiben:

- U-Zuweisung: User benutzt nur eine Modalität bezüglich eines Ziels
- U-Äquivalenz: User ist unentschlossen zwischen Modalitäten, benutzt mal diese und mal jene
- U-Redundanz: User benutzt mehrere Modalitäten gleichzeitig, um das gleiche auszudrücken
- U-Komplementarität: User benutzt mehrere Modalitäten ergänzend für ein Ziel

## 0.4 Benutzbarkeit (Usability)

- Idealzustand: System-Eigenschaften und User-Preferences sind "kompatibel", d.h. das System stellt den Präferenzen des Nutzers entsprechende Modalitäten zur Verfügung

- Welche Eigenschaft dient welchem Ziel? Beispiel:  
*schnelle Interaktion* durch Äquivalenz (Tastatur $\leftrightarrow$ Sprache)  
*intuitive Bedienung* durch Redundanz (z.B. Auswahl von Orten)  
 und viele weitere...

### 0.5 NEIMO - Software für ein Usability Lab

1. Versuchsperson bekommt Aufgabe an einem multimodalen System
2. Verhalten wird protokolliert - auf verschiedenen Ebenen
3. Auswertung durch Experten (mit Hilfe einer Videorecorder-Metapher, d.h. die Experten können das abgeschlossene Experiment wie eine Aufzeichnung anschauen)

### 0.6 VIENA - analysiert mit den CARE-Properties

- z.B. äquivalent( $\{m1, m2\}, g$ ) für alle  $g \in G_{MEA}$ ,  $tw = \infty$ ,  $tr = 1 \in TR_{MEA}$   
 $m1 =$  gesprochene Sprache,  $m2 =$  geschriebene Sprache  
 also: gesprochene und geschriebene Eingabe sind bezüglich aller Ziele äquivalent
- einzige *Zuweisung* bei VIENA: Befehle, z.B. "move"

### 0.7 TYCOON - Unterschiede zu CARE

- entwickelt 1995 von Jean-Claude Martin am LIMSI, Frankreich
- "Types and goals of COOperation"
- Kritik (von CARE!): "zu wenig auf den Benutzer bezogen"
- zusätzliche Eigenschaft: Transfer (z.B. eine Aufgabe auf eine andere Modalität lenken)

### 0.8 COMIT - Software zur Erstellung grafischer Benutzeroberflächen

- unterstützt Sprache, Tastatur, Maus
- COMIT definiert eine eigene Sprache zur Definition der möglichen User-Eingaben, z.B.:  
*specialization V1 SPEECH "button"*  
*complementary\_coinc V2 SPEECH "called"*  
*KEYBOARD \**  
*complementary\_coinc V3 SPEECH "here"*  
*MOUSE click \**  
*complementary\_sequence V1 V2 V3*  
*complementary\_duration 1500*
- => "create a button called <OK> here <click>"
- Prediction: bessere Erkennung von Kommandofolgen durch Wahrscheinlichkeiten bestimmter Abfolgen
- Beispiel für *schnellere Interaktion* durch Redundanz: Verzicht auf Sicherheitsabfrage beim Verlassen des Programms, wenn "Quit" sowohl gesprochen als auch getippt wird

### 0.9 Zu klärende Fragen

- Gibt es Unterschiede im Verhalten einzelner User?
- Gibt es eine Änderung des User-Verhaltens mit der Zeit?
- Experimente über diese Fragen kommen oft zu unterschiedlichen Ergebnissen

### 0.10 Quellen

- Martin, J.-C.: TYCOON: Theoretical Framework and Software Tools for Multimodal Interfaces, 1995.
- Nigay/Coutaz: Multifeature Systems: The CARE Properties and Their Impact on Software Design, 1995.
- Nigay/Coutaz/Salber: Multimodality from the User and System Perspectives, 1995.