

# Tangible User Interfaces

## Beispiele und eine Taxonomie

Patrick Frigg  
Betreuer: Michael Rohs

ETH Zürich – Departement für Informatik  
Seminar Verteilte Systeme SS2005 zum Thema  
“Smarte Objekte und smarte Umgebungen”  
padi@vis.ethz.ch

Abstract – Die Komplexität von Software nimmt laufend zu und Ubiquitous Computing lässt die digitalen Diener des Alltags schrumpfen. Beide Entwicklungen machen neue Ansätze im Bereich der Human-Computer Interaction dringend notwendig. Bei Tangible User Interfaces soll die Interaktion weg von den virtuellen GUIs zurück in die physische Welt, in unsere Hände gebracht werden. Dieser Report gibt einen Einblick in das umfangreiche Gebiet der TUIs und stellt eine entsprechende Taxonomie vor.

## 1 EINFÜHRUNG

In dieser Arbeit werden wir zunächst ein konkretes Beispiel zu einem Tangible User Interface betrachten und uns anschliessend der Frage widmen, wie man Arbeiten in diesem Gebiet klassifizieren könnte.

### A. Tangible User Interfaces

Das Gebiet der Tangible User Interfaces (TUIs) kann als eine Ausdehnung der Human-Computer Interaction Forschung betrachtet werden. Die traditionell virtuelle Tastatur/Maus-Bildschirm Interaktion (GUIs) wurde durch neue physische Möglichkeiten erweitert.

Verschiedenste Begriffe wurden für dieses Gebiet im Laufe des letzten Jahrzehnts eingeführt wie Graspable-, Physical-, Extreme- oder Embodied interfaces. Tangible Interfaces wird jedoch als der am stärksten verbreitete Begriff betrachtet [1]. Ishii und Ulmer definierten TUIs als Benutzerschnittstellen, welche die Reale Welt erweitern, indem sie Alltagsgegenstände und Umgebungen mit digitalen Informationen koppeln [2]. Daran lässt sich deutlich der Einfluss von Ubiquitous Computing und Augmented Reality erkennen, wobei sich TUIs der Thematik aus der Sicht des Users annehmen.

Der Schritt von der virtuellen zur realen Welt stellt die HCI Forschung vor neue Aufgaben. Dieser Schritt erweitert jedoch den Gestaltungsraum für

die Eingabe und Repräsentation von Information so signifikant, dass man sich von Tangible User Interfaces grosse Fortschritte bezüglich Intuitivität, Benutzbarkeit und Effizienz verspricht.

### B. Beispiele

Gleich zu Beginn möchten wir drei Beispiele vorstellen um einen Eindruck zu schaffen was TUIs sind und wie verschieden sie in ihrer Erscheinung sein können. Diesen Beispielen werden wir unter anderem im zweiten Teil der Arbeit wieder begegnen wenn wir Fishkins Taxonomie betrachten [3].

„Great Dome“ [1]

Der Anwender arbeitet mit physischen Objekten wie zum Beispiel einem kleinen Model des MIT Campusgebäudes „Great Dome.“ Auf einem Augmented Desktop wird ein Kartenausschnitt der Umgebung des MIT Gebäudes angezeigt. Das System kann Verschiebungen oder Drehungen des Gebäudes auf diesem Desktop erfassen und passt den projizierten Kartenausschnitt so an, dass das Model immer auf dem richtigen Fleck der Karte steht.



Abb. 1 – Great Dome auf dem Augmented Desktop. (Quelle: metaDESK Demo-Video '97)

„Shakepad“ [4]

Das Display des kleinen Schlüsselanhänger-Computer kann durch Schütteln gelöscht werden.

Neigungssensoren erkennen das Input Event „Schütteln“ und löschen die Ausgabe welche sich direkt im Eingabegerät selbst befindet.



**Abb. 2 – Shakepad Schlüsselanhänger-Computer** (Quelle: [4])

„ToonTown“ [5]

Dieses System ist ein virtuelles Auditorium in welchem die Teilnehmer eines Audio-Chats als kleine Animationsfiguren dargestellt werden. Wenn man die Figuren versetzt ändert sich die Lautstärke des entsprechenden Chat-Teilnehmers.



**Abb. 3 – ToonTown Audio-Chat System** (Quelle: [5])

## 2 TOOLSTONE

Im ersten Teil befassen wir uns, wie bereits erwähnt, etwas genauer mit einem konkreten Beispiel. Beim ToolStone handelt es sich um ein drahtloses multiple degree-of-freedom (MDOF) Eingabegerät. Jun Rekimoto und Eduardo Sciammarella entwickelten dieses Gerät für die nicht dominante Hand in bimanuellen (zweihändigen) Schnittstellen. [6]

### A. Motivation

Die Computermaus ist zweifelsohne noch immer das erfolgreichste Eingabegerät neben der Tastatur. Mit der kontinuierlich steigenden Komplexität von Software speziell im Bereich professioneller Anwendungen (Desktop Publishing, Bildbearbeitung, CAD, usw.) werden die Einschränkungen dieses generischen Werkzeugs jedoch zunehmend frustrierend. Die Arbeit mit der Maus setzt voraus, dass man die Funktionen am Bildschirm sehen und selektieren kann. Zahlreiche Werkzeugleisten, Pop-Up Menus, Scrollbalken und Menuleisten müssen angezeigt werden und lassen die eigentliche Ar-

beitsfläche schrumpfen. Abhilfe schafft meistens nur ein grösserer oder sogar ein zweiter Monitor was jedoch wiederum die Distanz und somit den Aufwand für die Mausbewegungen in die Höhe treibt.

Die nicht-dominante Hand (normalerweise die linke) kann selbstverständlich die Tastatur bedienen und gewisse Tasten oder Kombinationen zur Steuerung verwenden. Während diese bei Apple noch relativ strukturiert sind muss man bei anderen Betriebssystemen für jede Applikation ein eigenes Shortcut-Schema erlernen, abgesehen von ein paar systemweit vorgegebenen Befehlen. Selbst bei Apple gibt es aufgrund von Softwareproduzenten welche ihre Strukturen über die verschiedenen Plattformen hinweg wahren wollten Inkonsistenzen. Zu diesem Mangel an Struktur kommt hinzu, dass es für Tastaturbefehle keine visuelle Unterstützung gibt welche (abgesehen von den Buchstaben) die unterlegten Funktionen anzeigen würde.

ToolStone soll also helfen aus einer grossen Palette von Funktionen klar strukturiert, Platz sparend und leicht erlernbar auswählen zu können. Dabei sollen möglichst alle physischen Möglichkeiten des Menschen genutzt werden. Neben dem Verschieben und dem Klicken kann die menschliche Hand problemlos einen Gegenstand rotieren, kippen und auf eine andere Fläche umlegen, ohne dass dabei die visuelle Aufmerksamkeit des Benutzers notwendig wäre. Der Mensch kann ganze Abfolgen von solchen Manipulationen erlernen und automatisch ausführen. Und nicht zuletzt besitzt er auch die Fähigkeit die Funktion eines Gegenstands zu Differenzieren mit der Art und Weise wie er ihn in der Hand hält.



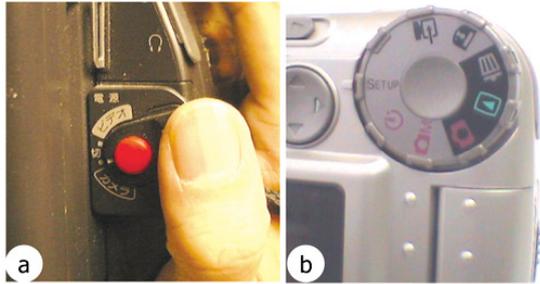
**Abb. 4 – Funktionalität differenzieren je nachdem wie man ein Instrument hält.**

### B. Design-Prinzipien

Rekimoto und Sciammarella bezeichnen eine durch solche Manipulationen erweiterte Eingabemethode als Rich-Action Input (RAI). Auf der Suche nach einem Werkzeug welches einen solchen RAI ermöglichen setzen sie zwei Prinzipien voraus:

1. Der aktuelle Zustand des Werkzeuges soll mittels Berührung wahrgenommen werden können.

Ein gutes Design zeichnet sich unter anderem dadurch aus, dass der Zustand des Objektes wahrgenommen werden kann ohne auf visuelles Feedback angewiesen zu sein.



**Abb. 5 – Wahrnehmen des aktuellen Zustands mittels Berührung.** (Quelle: [6])

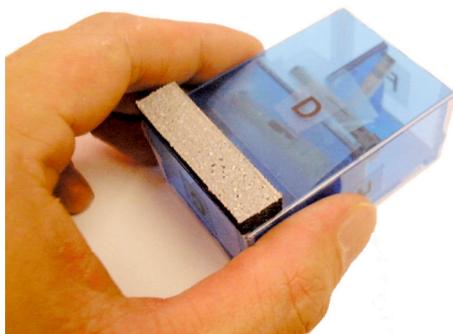
Am Beispiel dieser beiden Knöpfe ist gut ersichtlich was ein gutes Design auszeichnet. Während bei Schalter (a) jeder der drei Zustände (Kamera-Modus, Playback-Modus und Aus) mit dem Daumen und jederzeit während des Einsatzes wahrgenommen werden kann ist bei (b) der Benutzer auf eine visuelle Kontrolle des aktuellen Zustands angewiesen und muss diesen anhand der Beschriftung am Rad ablesen.

2. Der Interaktionsraum soll schnell erfasst und einfach zu verstehen sein.

Technisch gesehen ist es möglich eine Unzahl von Funktionen in ein einziges Gerät zu packen. Dies ist jedoch weitgehend nutzlos solange der Benutzer diese nicht finden kann. Die visuelle Erscheinung ist diesbezüglich von Wichtigkeit. Ein Benutzer sollte beim Betrachten des Gerätes erahnen können wie dieses bedient werden könnte.

### C. Design des ToolStone

Das Ergebnis auf der Suche nach einem solchen RAI Werkzeug war ein rechteckiges, drahtloses Objekt welches auf die zwei Gestaltungsprinzipien folgendermassen eingeht:



**Abb. 6 – Der ToolStone für die nicht-dominante Hand.** (Quelle: [6])

1. Wahrnehmung des aktuellen Zustands mittels Form und Orientierungshilfe.

Zum einen sind sämtliche Abmessungen wie Höhe, Breite und Tiefe voneinander verschieden. Somit ist es möglich zu ertasten ob das Gerät auf der „Fläche“, auf der „kurzen Kante“ oder der „langen Kante“ steht. Zusätzlich ist ein kleiner Balken am

unteren Ende einer der Flächen angebracht, der es ermöglicht über die Berührung festzustellen wie der Quader orientiert ist, respektive auf welcher der beiden jeweils kongruenten Flächen er liegt.

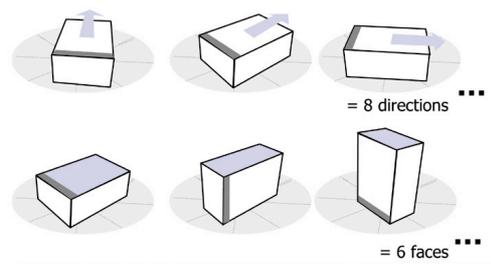
2. Semi-6DOF Eingabegerät.

Der Funktionsraum ist gegeben durch die Möglichkeiten einen solchen Quader zu manipulieren wenn er immer mit der Arbeitsfläche in Kontakt bleiben muss (mindestens mit einer Kante). Folgende Informationen werden vom ToolStone erfasst wenn er sich auf einem Graphiktablett befindet:

- X-Y Position auf einem Tablett.
- Orientierung des Quaders.
- Auf dem Tablett aufliegende Fläche.
- Oder den Neigungswinkel wenn nur eine Kante das Tablett berührt.

### D. Art der Interaktion

In der Motivation haben wir gelesen, dass der ToolStone dazu dienen soll, den Platzverbrauch von Werkzeugpaletten zu minimieren und die Effizienz bei der Selektion eines Werkzeuges zu maximieren. Dazu steuert ToolStone mit der nicht-dominanten Hand eine Werkzeugpalette die auf dem Bildschirm angezeigt wird. Jedem der 8 Sektoren des Quaders (welche durch  $45^\circ$  unterteilt werden) kann eine Werkzeugpalette zugeteilt werden. Der Benutzer kann zwischen den verschiedenen Paletten wählen indem er den Quader entsprechend rotiert. Zusätzlich können jeder Fläche des Quaders weitere acht Paletten zugeordnet werden.

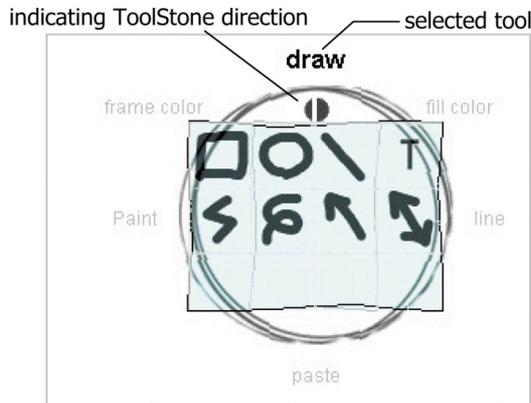


**Abb. 7 – Insgesamt 48 mögliche Positionen des Quaders.** (Quelle: [6])

Dadurch ergeben sich theoretisch  $8 \times 6 = 48$  mögliche Kombinationen, welchen jeweils eine ganze Toolpalette zugeordnet werden kann. Diese ist auf dem Bildschirm sichtbar und folgt den Bewegungen des ToolStone. Dadurch kann die Palette immer in die Nähe des Arbeitsbereiches gebracht werden, was die nötigen Bewegungen mit der dominanten Hand (z.B. Maus) reduziert. Ausgewählt werden die Werkzeuge dann mit der dominanten Hand wie bei einer normalen Toolleiste.

## E. Visuelle Unterstützung

Sich alle 48 Stellungen und die zugeordneten Paletten zu merken dürfte zu kompliziert sein. Ein effizienter Umgang mit dem ToolStone wird durch die simultane, visuelle Unterstützung erreicht.



**Abb. 8 – ToolStone Visualisierung auf dem Bildschirm.** (Quelle: [6])

Die Abb. 8 zeigt die semi-transparente Visualisierung des ToolStone mit der aktiven Toolpalette in der Mitte. Um die Palette herum sind die 45° Sektoren mit den Namen ihrer zugeordneten Toolpaletten angeschrieben. Der Benutzer kann schnell ersehen, ob die gewünschte Funktion auf der aktuellen Seite des Quaders enthalten ist und in welchem Sektor diese liegt. Falls nicht, kann er schnell die anderen Flächen durchprobieren bis er sein Tool gefunden hat. Diese simultane Unterstützung fördert ein exploratives Vorgehen, begleitet von einem Lerneffekt, wodurch der Benutzer immer stärker ein Gefühl für die Anordnung seiner Toolpaletten bekommt.

## F. MDOF Interaktionen

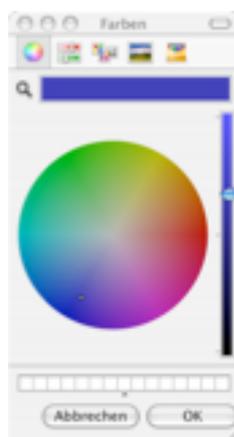
Ist ein Werkzeug selektiert können die Freiheitsgrade des ToolStone für andere Zwecke eingesetzt werden.

### Farbraum auswählen:

Bei manchen Interaktionen geht es darum, Parameter mit mehr als 2 Dimensionen zu kontrollieren. Ein Beispiel ist der Farbraum, welcher aus den 3 Dimensionen

1. Farbton
2. Sättigung
3. Helligkeit

aufgespannt wird. Bestehende Farbwahldialoge zwingen den Benutzer häufig zu unintuitiv-



**Abb. 9 – Farbraum ist dreidimensional.**

ven Operationen da es ihnen nicht gelingt den 3-D Farbraum einleuchtend auf die 2-D Bildschirmfläche abzubilden. Häufig wird einer der drei Parameter separat als ein Scrollbalken oder ähnliches dargestellt (Wie z.B. die Helligkeit in Abb. 9).

Die X- (oder Y-) Position des ToolStone kann nun verwendet werden um diesen dritten Parameter (Helligkeit) simultan zu kontrollieren während der Benutzer mit der Maus aus der verbleibenden 2-D Fläche die Farbe auswählt.

### Zoomen und Scrollen der Arbeitsfläche:

Der Benutzer muss sich mit der Maus an den Arbeitsflächenrand begeben um die Scrollbalken zu manipulieren. Stattdessen könnte er den ToolStone auf die entsprechende Fläche legen auf welcher die X-Y Bewegung des Quaders die Arbeitsfläche verschiebt und eine Drehung des Quaders ein rein- und raus-Zoomen bewirkt.

### Rotieren von 3-D Objekten:

In einer anderen Anwendung könnte man ein 3-D Objekt mit der Maus selektieren und festhalten während mit der Verschiebung des ToolStone die Rotationsachsen ausgerichtet werden und das Drehen des Quaders den Rotationswinkel des Objektes verändern würde.

### Steuerung von virtuellen Kameras:

Wenn der ToolStone auf einer entsprechenden Fläche liegt, dient er als Steuerung für virtuelle Kamerafahrten - zum Beispiel durch eine Wohnung. Die Maus bleibt dabei frei für andere Interaktionen wie zum Beispiel der Änderung des Sichtfeldes der Kamera. Denkbar wäre auch, dass die Kamera sich vollständig im dreidimensionalen bewegen kann durch die Kombination von Maus und ToolStone.

## 3 TAXONOMIE

In diesem zweiten Teil wird nun die Möglichkeit einer Klassifizierung für TUIs beleuchtet, welche Fishkin in seiner Arbeit „A taxonomy for and analysis of tangible interfaces“ [3] vorschlägt.

### A. Motivation

Kenneth P. Fishkin ist bei Intel Research Seattle tätig und war früher bei Xerox PARC. Dort war er Co-Gründer der „extreme UI“ Gruppe, welche sich mit dem Thema der TUIs befasste. Er machte immer wieder die Erfahrung, dass TUIs im Wesentlichen ein „I know one when I see one“ Forschungsgebiet ist - bis jetzt also noch nicht klar genug definiert wurde. Zwar wurden ein paar wenige Versuche unternommen eine Definition oder Organisation aufzustellen, diese reichten jedoch selten über ein „proof of concept“ hinaus.

Die in der Einführung kurz vorgestellten Beispiele und auch der ToolStone aus dem letzten Kapitel zeigen, wie umfangreich das Gebiet ist und wie vielfältig das Erscheinungsbild und die Interaktion ausfallen können. Fishkins Ziel ist ein möglichst allgemeines System, welches diese ganze Vielfalt enthält und gleichzeitig die früheren Bemühungen um eine Organisation des Gebietes berücksichtigt.

Eine solche Taxonomie soll es ermöglichen, Arbeiten miteinander zu vergleichen, die Designphase solcher Arbeiten zu unterstützen, eventuell wenig erforschte Klassen von TUIs zu identifizieren und die bestehenden Klassifizierungen zu konsolidieren.

## B. Definition für TUIs

Zunächst soll definiert werden wie der Raum für TUIs eingeschränkt werden soll, oder eben nicht. Ursprünglich definierten Ishii und Ullmer diese als Schnittstellen welche „augment the real physical world by coupling digital information to everyday physical objects and environments.“ [2] In einem späteren Paper engten sie dieses Definition weiter ein und gaben an, dass ein solches Interface die Unterscheidung zwischen Eingabe- und Ausgabe-Gerät aufheben würde, obschon „interesting interaction regimes are highlighted by relaxing these expectations.“ [7]

Aufgrund dieser Unstimmigkeit setzt Fishkin in seinem Paper [3] eine Definition an, welche so allgemein wie möglich formuliert ist, noch allgemeiner als die erste Definition von Ishii und Ullmer.

1. Ein Input Event erfolgt. Typischerweise handelt es sich dabei um eine physische Manipulation, die der Benutzer mit seinen Händen an einem (scheinbaren) Alltagsgegenstand durchführt. Dazu zählen Dinge wie kippen, schütteln, quetschen, drücken oder einfach bewegen.
2. Ein Computersystem registriert diesen Event und verändert seinen Zustand.
3. Ein Output Event erfolgt. Dieses Feedback geschieht durch die Änderung der physischen Erscheinung eines Objektes. Das Objekt kann seinen Bildschirminhalt ändern, aber auch wachsen, schrumpfen, einen Ton erzeugen oder ein haptisches Feedback geben.

## C. Eine Taxonomie zur Analyse

Eine derart breit gefasste Definition bringt wenig Klarheit darüber, wie TUIs eingeordnet werden sollen. Neben den bereits erarbeiteten Beispielen schliesst die Definition auch Schnittstellen von technischen Apparaturen aus dem Gebiet des Industrie-Designs ein. Konsolen zum verstellen des Autositzes die selbst aussehen wie kleine Autositze, Glückwunschskarten die beim Öffnen eine Melodie

abspielen, klassische Joysticks, Joysticks für Rennsimulationen die aussehen wie Lenkräder oder die Computertastatur um nur ein paar wenige zu nennen.

Fishkin schliesst nach der Analyse von 60 Interfaces, dass sich kein vernünftiges ja/nein Kriterium finden lässt, welches gewisse Schnittstellen eindeutig zu TUIs erklärt und andere ausschliesst. Stattdessen schlägt er vor „Tangibility“ als ein mehrdimensionales Attribut zu betrachten, konkret als einen 2-dimensionalen Raum der von den Achsen Embodiment und Metaphor aufgespannt wird. Weitere Achsen wären möglich, würden jedoch die Einfachheit und Klarheit der Klassifizierung reduzieren.

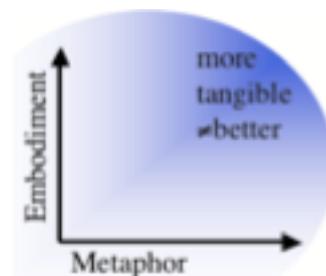


Abb. 10 – Tangibility als 2-D Raum.

Je weiter weg man sich nun vom Ursprung bewegt, also je höher die Level der beiden Attribute werden, desto ausgeprägter ist die Tangibility des Systems. Dies bedeutet jedoch nicht, dass das System „besser“ oder „schlechter“ ist. Die Qualität einer Schnittstelle hängt bekanntlich von der konkreten Implementierung und der Beachtung einer Vielzahl weiterer Kriterien ab und nicht nur von der Tangibility.

### C.1 Erste Achse: Embodiment



Schritt 1 und Schritt 3 aus der Definition kann sich sowohl an verschiedenen als auch am gleichen Ort ereignen. Embodiment (Verdinglichung, Verkörperung) befasst sich mit dieser Distanz zwischen Input und Output oder - anders ausgedrückt - wie stark der Benutzer das Gefühl hat, dass das System mit dem er interagiert sich in dem Werkzeug selbst befindet.

**Full:** In diesem Grenzfall ist das Outputgerät mit dem Inputgerät identisch. Es handelt sich dabei um die natürlichste Art der Interaktion aus unserer realen Welt. Wenn wir zum Beispiel ein Blatt falten ändert genau dieses Blatt seinen Zustand. Ein TUI aus dieser Kategorie ist der eingangs vorgestellte „Shakepad“ bei dem das Schütteln des Gerätes seinen Bildschirminhalt ändert.

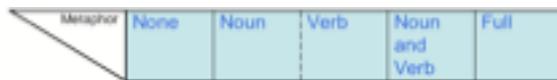
**Nearby:** Hier findet der Output in unmittelbarer Nähe des Input Objektes statt, respektive ist eng an dessen Fokus gekoppelt. Ein Beispiel aus dem

Alltag ist ein Stift welcher den Zustand des direkt darunter liegenden Papiers ändert. Auch der „Great Dome“ aus der Einführung ändert die darunter liegende Kartenprojektion.

**Environmental:** In diesem Fall befindet sich der Output um den Benutzer herum. Zum Beispiel als Ton, Umgebungslicht oder Zimmertemperatur. „ToonTown“ ist ein ideales Beispiel für diesen Level von Embodiment.

**Distant:** In dieser Kategorie findet der Output „dort drüben“ auf einem Bildschirm oder sogar in einem anderen Raum statt. Die Computertastatur ist ein beispielhafter Vertreter von Distant Systemen.

## C.2 Zweite Achse: Metaphor



Metaphern sind allgemein bekannt als ein sehr mächtiges und gut erforschtes Instrument wenn es um Design geht. Fishkin ist der Ansicht, dass gerade TUIs prädestiniert sind um sich Metaphern zu nutzen zu machen, weil sie durch ihre physische Greifbarkeit auf eine viel grössere Auswahl an Metaphern zurückgreifen können als bei einer rein visuellen Repräsentation. [3]

Die Metaphern werden grob in zwei Gruppen unterteilt. „Metaphern des Nomens“ beziehen sich auf die gesamte Erscheinung des Objektes. Das Aussehen, die haptischen Eigenschaften bis hin zum Geruch und der Akustischen Wahrnehmung können uns an Alltagsgegenstände erinnern. Die zweite Gruppe „Metaphern des Verbs“ beinhalten die Analogien der Bewegung, also der Natur der Manipulation mit welcher das Interface bedient werden soll.

**1. None:** Zum dem Interface existieren keinerlei Analogien aus dem Alltag.

Bsp.: Tastatur.

**2A. Noun:** Die Metapher des Nomens schafft die Analogie über ihre Erscheinung. Wie bereits erwähnt kann sie die Form, das Aussehen, den Klang oder den Geruch eines bekannten Objekts besitzen oder sich wie ein solches anfühlen. Die Analogie endet jedoch mit dem Aussehen und die Bedienung des Interfaces erfolgt durch Bewegungen die mit dem realen Objekt nichts zu tun haben.

„An <A> in our system is like an <A> in the real world.“ [6]

Bsp.: Windows/Desktop Systeme. Mit RFID Tags ausgerüsteten Gegenstände, welche (in die Nähe des Systems gebracht) eine Interaktion auslösen.

**2B. Verb:** Die auszuführende Geste steht im Mittelpunkt bei der Metapher des Verbs. Während das

Aussehen des Interfaces weitgehend irrelevant ist, kann der Benutzer bei dessen Bedienung auf ihm vertraute Manipulationen von realen Objekten zurückgreifen.

„<X>-ing in our system is like <X>-ing in the real world.“ [6]

Bsp.: Kippen eines PDA um ein Dokument zu scrollen (zur Seite gleiten lassen). Drag & Drop (nur die Tätigkeit als solches, ungeachtet ob die bewegten Symbole Metaphercharakter haben).

**3. Noun and verb:** Sowohl die Erscheinung wie auch die Bedienung ist angelehnt an ein Objekt aus der realen Welt.

„<X>-ing an <A> in our system is like <X>-ing something <A>-ish in the real world.“ [6]

Bsp.: Auf einem Windows/Desktop System ein Dokument per Drag & Drop in den Papierkorb fallen lassen.

**4. Full:** In diesem Extremfall bedarf es keiner Analogien mehr, da für den Benutzer das virtuelle System und das physische System *identisch* sind.

Bsp.: Tablet PCs (Der Stift mutiert tatsächlich das elektronische Dokument).

Zusammenfassend lässt sich also sagen, dass Metaphern einen echten Mehrwert für das Design einer Mensch-Maschine Schnittstelle bedeuten können. Gleichzeitig sollten sie aber nur unter grosser Sorgfalt eingesetzt werden. Eine mehrdeutige oder falsche Metapher kann sogar kontraproduktiv sein und den Lernaufwand für den Benutzer erhöhen. Ein gern herangezogenes Beispiel war der Papierkorb von Apples Mac OS. Das Löschen/vernichten eines Dokuments durch Drag & Drop in den Papierkorb erwies sich als eine hervorragende Metapher. Apple erweiterte die Funktionalität des Papierkorbs und benutzte ihn ebenfalls um Datenträger (Disketten) aus dem Laufwerk auszuwerfen. Dies verunsicherte die Benutzer welche mit dem Papierkorb das Wegwerfen/Löschen von Dingen assoziierten. In späteren Versionen wechselte das Icon des Papierkorbs zu einem Eject Symbol sobald ein Datenträger angewählt wurde damit die Benutzer nicht mehr befürchten mussten sie würden ihren Datenträger löschen.

## D. Anwendung der Taxonomie

Die beiden Achsen zusammen ergeben die Ebene für Fishkins Taxonomie. Oben rechts im Gebiet Full-Full ist die tangible Ausprägung am stärksten. Noun und Verb sind gleichwertig bezüglich der Tangibility und aus diesem Grund nur mit einer gestrichelten Linie getrennt.

## D.1 Einordnung der Beispiele

Wir wollen nun zu den einleitenden Beispielen zurückkehren um deren Platz in dieser Taxonomie zu bestimmen.

Metaphor \ Embodiment	None	Noun	Verb	Noun and Verb	Full
Full					
Nearby					
Env.					
Distant					

Abb. 11 – TUIs Klassifiziert.

Der „Great Dome“ hat offensichtlich ein Nearby Embodiment da das Gebäudemodell direkt auf der Karte liegt und der Fokus somit eng gekoppelt ist. Da es sich um eine Miniatur des Gebäudes handelt liegt eine Noun Metaphor nahe. Aber auch das verschieben des Gebäudes kann mit dem Verändern der eigenen Position auf der Karte verglichen werden. Der „Great Dome“ ist also unter Noun and Verb Metaphor einzuordnen.

Beim „Shakepad“ hat der Benutzer den Eindruck das ganze System in der Hand zu halten. Es gibt keine kognitive Distanz zwischen Eingabe und Ausgabe was für ein Full Embodiment spricht. Die Metapher ist eine verbale, weil das Schütteln mit dem Zerstören/Randomisieren/Löschen einer Ordnung (Zeichnung) assoziiert werden kann.

„ToonTown“ ist mit seiner Tonausgabe eindeutig ein Environmental Embodiment. Auch ist es leicht einzusehen, dass die Animationsfiguren Menschen darstellen und ein Versetzen der Figuren einem Sitzplatzwechsel der Teilnehmer entspricht. Also ebenfalls eine Noun and Verb Metaphor.

Die erwähnten Peripheriegeräte wie Tastatur, Joystick und Steeringwheel werden allesamt unter Distant Embodiment eingeordnet. Das zur Interaktion gehörende Feedback findet ganz klar „dort drüben“ auf dem Bildschirm statt. Das System scheint nicht im Interface integriert zu sein. Die Klassifizierung von Joystick und Steeringwheel ist ein gutes Beispiel um die Konzepte Noun and Verb nochmals zu verdeutlichen: Während beim Joystick die Manipulation des Sticks den Zustand des Systems analog ändert (nach rechts, Auto steuert nach rechts) erinnert sein Erscheinungsbild an kein reales Objekt (von modernen Steuerknüppeln abgesehen). Die Ähnlichkeit welche das Steeringwheel mit einem Lenkrad hat leuchtet jedoch ein. Folglich besitzt dieses beide Metaphern - Noun and Verb.

Metaphor \ Embodiment	None	Noun	Verb	Noun and Verb	Full
Full					
Nearby					
Env.					
Distant					

Abb. 12 – Traditionelle Peripheriegeräte Klassifiziert.

Doch gerade hier stellt sich eine interessante Frage: Wann wird ein technisches Interface/Gerät aufgrund seiner Alltäglichkeit zu seiner eigenen Metapher? (Das Telefon zum Beispiel oder eben auch der Joystick, welcher moderne Flugzeuge, Kräne und eine Vielzahl anderer Maschinen steuert)

## D.2 Einordnung des ToolStone

Abschliessend wollen wir nun nochmals ToolStone zurückkommen. Der blaue Quader gibt uns kaum eine Idee um was es sich handeln könnte oder wozu er dienen könnte. Dies schliesst eine Noun Metaphor aus. Für die Gesten zur Bedienung des ToolStone können aber durchaus Assoziationen gefunden werden. Zum Beispiel bei der Rotation entspricht das drehen des ToolStone dem Drehen des Objektes oder das Fahren mit dem ToolStone entspricht dem Fahren mit der Kamera. Entsprechend ist der ToolStone in seiner gegenwärtigen Form auf der Verb Metaphor einzustufen.

Bezüglich Embodiment wurde der ToolStone von Fishkin nicht wie die klassischen Peripheriegeräte als Distant eingestuft (siehe Abb. 12). [3] Der Grund dafür liegt in der visuellen Repräsentation auf welche sich der Benutzer grundsätzlich konzentriert. Diese findet auf dem Bildschirm und im Idealfall in unmittelbarer Nähe des Arbeitsbereiches statt. Der kognitive Fokus des Benutzers wird folglich kaum abgelenkt was einem Nearby Embodiment entspricht.

Metaphor \ Embodiment	None	Noun	Verb	Noun and Verb	Full
Full					
Nearby					
Env.					
Distant					

Abb. 13 – ToolStone Klassifiziert.

## 4 SCHLUSSBETRACHTUNGEN

Persönlich würde ich eine Unterstützung wie den ToolStone sehr zu schätzen wissen. Im Bereich der Bildbearbeitung oder des Desktop Publishing ist ein unendlicher Wechsel des selektierten Werkzeuges an der Tagesordnung. Mit der Tastatur lässt sich theoretisch jede Funktion aufrufen. Meistens besitzt man jedoch nur ein kleines Repertoire von Tastenkombinationen die man auswendig gelernt hat. So ist auch beim ToolStone die kritische Eigenschaft eine effiziente, simultane Visualisierung in nächster Nähe der Arbeitsfläche.

Ob solche Werkzeuge jedoch weiter als bis zu Professionellen Anwendern gelangen würden, wage ich zu bezweifeln. Die Interaktion mit dem klassischen Computer ist doch schon seit sehr langer Zeit bei Tastatur und Maus stehen geblieben. Wahrscheinlicher ist es, dass das Ubiquitous Computing zuvorkommen wird und der Computer wie wir ihn heute kennen aufgeteilt wird auf verschiedene smarte Gegenstände, welche gewisse Funktionen zur Verfügung stellen und untereinander kommunizieren.

Wie wir mit solchen Full Embodied und möglichst auf Metaphern basierenden Gegenständen interagieren sollen, könnte in Zukunft eine wegweisende Fragestellung sein. Vielleicht wird aber auch die Technik der HCI zuvorkommen und den Menschen wieder in ein Schema drücken. Die Lernbereitschaft ist doch erstaunlich, wenn man sich vor Augen führt, wie selbstverständlich das Schreiben von Kurznachrichten auf den zwölf Tasten des Mobiltelefons geworden ist. Und das nicht nur für Early Adopters sondern Generationsübergreifend und weitverbreitet.

Ein sehr schönes Zitat, das meine Ansicht ausdrückt, stammt von D. Bishop, der mit seiner berühmten Marble Answering Machine von 1992 einer der Pioniere der TUIs war und einmal bemerkte:

„don't you think that ,tangible user interface' might someday sound like ,horseless carriage?""  
[1]

## 5 BIBLIOGRAPHIE

- [1] Holmquist L E, Schmidt A, Ullmer B (2004) *Guest editor's introduction*. In: Personal and Ubiquitous Computing, Vol. 8, Issue 5, Springer, 2004, 291 - 293
- [2] Ishii H, Ulmer B (1997) *Tangible bits: towards seamless interfaces between people, bits, and atoms*. In: Proceedings of the CHI'97 conference on human factors in computing systems, Atlanta, Georgia, March 1997, pp 234 - 241
- [3] Fishkin P. K. (2004) *A taxonomy for and analysis of tangible interfaces*. In: Personal and Ubiquitous Computing, Vol. 8, Issue 5, Springer, 2004, 347 - 358
- [4] Levin G, Yarin P (1999) *Bringing sketching tools to keychain computers with an acceleration-based interface*. In: Extended abstracts of the CHI'99 conference on human factors in computing systems, Pittsburgh, Pennsylvania, May 1999, pp 268 - 269
- [5] Singer A, Hindus D, Stifelman L, White S (1999) *ToonTown: less is more in somewire audio spaces*. In: Proceedings of the CHI'99 conference on human factors in computing systems, Pittsburgh, Pennsylvania, May 1999, pp 104 - 111
- [6] Rekimoto J, Sciammarella E (2000) *ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices*, Proc. Of UIST2000.  
<http://www.csl.sony.co.jp/person/rekimoto/papers/uist00.pdf>
- [7] Ulmer B, Ishii H (2001) *Emerging frameworks for Tangible User Interfaces*. In: Carroll JM (ed) Human-computer interaction in the new millennium. Addison-Wesley, New York, pp 579-601