

“I can“-Design: Methodik für das benutzerzentrierte Design nicht-standardisierter Icons

Christian Wolff
Professur für Medieninformatik
Institut für Information und
Medien, Sprache und Kultur
Universität Regensburg
93040 Regensburg
christian.wolff@computer.org

Stefanie Götzfried
Tools & Equipment
MBtech Group
Kolumbusstraße 2
71063 Sindelfingen
Stefanie.Goetzfried@mbtech-
group.com

Abstract

Wir stellten eine Methode für das benutzerzentrierte Design nicht-standardisierter Icons vor. Die Methode, die auf zwei aufeinander aufbauenden Icon-Testverfahren basiert, wurde im Rahmen eines Forschungsprojekts zur Entwicklung eines Traceability-Tools zur anwendungszentrierten Architekturentwicklung von eingebetteten Echtzeitsystemen entwickelt und eingesetzt.

Sie stützt sich zum einen auf die Erfahrungen nach Nielsen & Sano (1994) zum anderen stellen die Testverfahren der ISO-Norm für graphische Symbole (ISO - 9186-1+2 2007, 2008) eine Grundlage für die Testmethodik dar.

Das Testverfahren wurde von Softwaredesignern und Architekten gleichermaßen gut angenommen. Auch der Raum zum Einbringen eigener Ideen wurde sehr häufig genutzt.

So konnten nicht nur passende Visualisierungen für Einzelfunktionen gefunden, sondern zudem das Verständnis der einzelnen Icons in ihrem Kontext als Gruppe überprüft werden. Das Verfahren eignet sich zum Design von Icons für neuartige Funktionen als Discount Usability Methode bei geringer User-Anzahl.

Keywords

Icon Design, User Experience, User Test, Screen Design

1.0 Einleitung

Benutzerzentrierte Entwicklungsverfahren haben im Bereich der Ingenieur-Informatik nach wie vor Außen-seiterstatus. Viele eingebettete Softwaresysteme sind weitgehend interaktionsfrei, weshalb User Centered Design (UCD) dort eine untergeordnete Rolle einnimmt. Vernachlässigt wird auch die Einbindung der Softwareentwickler als Nutzer, wenn es um das Design von Entwicklungswerkzeugen für diesen speziellen Bereich geht.

Im Forschungsprojekt „Anwendungszentrierte Architekturentwicklung von embedded Echtzeitsystemen“ (Turban et al. 2009) stellte diese Thematik den Schwerpunkt des Projektpartners Medieninformatik (Universität Regensburg)

dar. Gemeinsam mit den Kooperationspartnern des Kompetenzzentrums für Software Engineering der Hochschule Regensburg und der MBtech Group wurde ein Werkzeug entwickelt, das die Traceability-Lücke zwischen Anforderungsmanagement und modellbasierter Softwareentwicklung schließt (Turban et al. 2009). Dabei rücken die Anwender eines solchen Tools, d. h. Software-Designer und -Architekten, in den Mittelpunkt der Entwicklung.

Im Projektverlauf wurden unterschiedliche Verfahren aus dem Methodenspektrum des UCD eingesetzt und an die projektbestimmende agile Entwicklungsmethodik angepasst. Einen besonderen Schwerpunkt bildete die Suche nach geeigneten ikonischen Repräsentationen (Signifikanten) für die

systemspezifischen und damit nicht-standardisierten Funktionen des Werkzeugs (auch Objekt oder Signifikat/Signatum) (Ferreira et al. 2006).

Ikonische Elemente gehören zu den Kernbestandteilen des Paradigmas grafisch-direktmanipulativer Benutzerschnittstellen, die auf eine lange Forschungstradition zurückblicken. Sowohl die theoretische Modellbildung (Rohr 1984; Chang 1987) als auch empirische Studien, die verschiedene Interaktionsparadigmen vergleichen (Benbasat & Todd 1993; Whiteside 1985), reichen bis in die 80er Jahre zurück.

Für unseren Projektkontext wurde eine Icon-Testdesign-Methode entwickelt, die in Anlehnung an die ISO Testmethodik zur Entwicklung von grafischen Symbolen (ISO - International Organization for

Standardization 9186-1+2 2007, 2008) eine benutzerzentrierte Entwicklung geeigneter Repräsentanten sicherstellt.

2.0 Voraussetzungen

Repräsentation allein durch Ikonen eignen sich dann, wenn wenig Platz für aussagekräftiges *wording* zur Verfügung steht, Benutzer verschiedene Elemente in einer Liste auseinander halten müssen oder Status kommuniziert werden soll. Zudem wird ihre Verwendung bei einem internationalen Nutzerkreis empfohlen (Goodwin 2009).

Diese Kriterien decken sich mit den Charakteristika unseres Werkzeugs: Es interagiert als zwischengeschaltete Komponente stetig mit anderer Software-Designsoftware (Anforderungsverwaltung, Entwurfseditoren), was die Platzverhältnisse zu sehr begrenzt, um im Dialogdesign Buttons mit aussagekräftigem *wording* für die teilweise neuartigen Funktionen einzusetzen. Durch den Einsatz einer Icon-Toolbar konnte eine mit beschrifteten Schaltflächen „übersäte“ Reiterleiste abgelöst und dadurch dem Nutzer ein Tabwechsel erspart werden.

Zudem handelt es sich um sehr abstrakte Funktionen (bspw. *Resource Constraint* oder *Design Constraint*), die selbst im Umfeld des Software Engineering noch nicht konventionalisiert sind. Sie werden gemeinsam in einer Baumstruktur dargestellt – ihre Unterscheidung erfolgt also mittels der vorangestellten Ikonen. Hierfür waren in einer frühen Projektphase „schnelle Erstlösungen“ (*rapid prototyping*) im Einsatz, die in den Testverlauf zur Evaluation mit aufgenommen wurden.

Das im Projekt entwickelte Werkzeug ist in die Produktreihe PROVEtech der Firma MBtech aufgenommen worden. Da die Standortverteilung des Unternehmens unter anderem die USA als auch

China umfasst, war eine weitgehende Internationalisierung des Werkzeugs eine weitere zu erfüllende Anforderung.

3.0 Methode/Testdesign

Die Methodik für eine passende visuelle Sprache sollte das mentale Model des spezifischen Nutzerkreises aufgreifen. Daher stand die Frage im Vordergrund, mit welchen Symbolen Software Designer und Architekten die neuen Fachfunktionen des Tools am ehesten assoziieren.

Abb 1 zeigt den Testablauf auf der Grundlage der Testverfahren für graphische Symbole nach (ISO - 9186-1+2 2007, 2008). Da beide Verfahren für das Design von Symbolen im öffentlichen Raum entwickelt wurden, und damit die vorausgesetzten Bedingungen (Größe des Zeichens, Ansprechen einer breiten Masse, Einzelabbildungen) von den Projektvorgaben (16 x 16 Pixel-Icons, sehr spezifische Zielgruppe, meist Abbildungen in einer Gruppe) abweichen, wurde das Verfahren vor allem durch die Nutzereinbindung bei der Ideensammlung und den abschließenden Usertest modifiziert.

Zudem musste sich das Testdesign an die projektbestimmende *agile Vorgehensweise* anpassen. Daher waren kurze Design-Test-Redesign-Zyklen und – in der jeweiligen Phase – ein rasches Erfassen der notwendigen Informationen anzusetzen, da die Nutzer nur begrenzt zu *usability*-Zwecken freigestellt wurden.

Den Kern der Methode bildet ein *n*-stufiger (im Projekt: zweistufiger) *judgement test* durch eine schriftliche Befragung, der durch einen anschließenden *comprehension test* (Benutzertest) ergänzt wird, wobei bereits bei

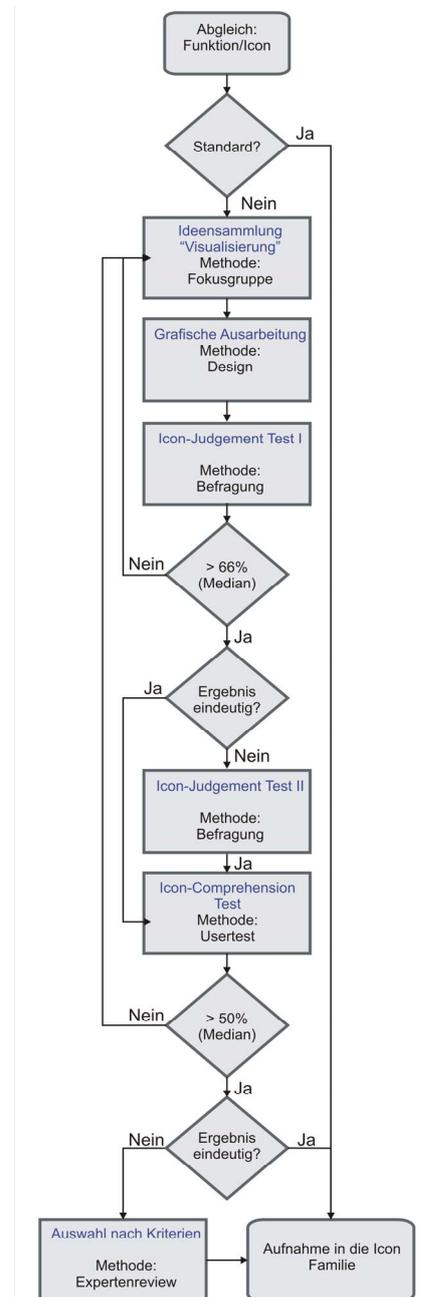


Abb 1: Ablauf des Testdesigns

der anfänglichen Ideensammlung die Nutzer miteinbezogen wurden.

3.1 Abgleich: Funktion/Icon

In einem ersten Schritt erfolgte der Abgleich der vorhandenen Funktionen mit bereits standardisiertem Icon-

Material, d. h. fertigen Icon-Familien. Eine klare Zuordenbarkeit von Icons zu einer gemeinsamen Icon-Familie ist ein wesentliches Design-Qualitätsmerkmal (Goodwin 2009). Dies war in unserem Fall auch aus Gründen der Abgrenzung zu korrespondierenden Designtools ein wichtiger Faktor. Daher wurde eine bestehende Icon-Familie ausgewählt, die alle benötigten Standard-Icons beinhaltet und zudem über eine Icon-Varietätsbreite verfügt, die als ideengebend für alle neuen Funktionen eingeschätzt wurde (Coelho o.D.). Das Design der neu zu entwickelnden Icons wurde damit durch die visuelle Sprache der gewählten Icon-Familie mitbestimmt.

Als Resultat dieser Phase folgte eine Auflistung aller Standard-Funktionen mit deren ikonischen Repräsentanten der gewählten Icon-Familie und – falls vorhanden – des bisherigen Repräsentanten aus dem ersten Tool-Prototyp. Alle nicht-standardisierten Funktionen wurden ebenso mit ihrem ggf. vorhandenen Repräsentanten aufgelistet und mit einem Platzhalter für die zu entwickelnde Visualisierung ausgewiesen.

3.2 Ideensammlung „Visualisierung“

Bewährte Icons bilden üblicherweise vertraute Objekte, Aktionen oder gewünschte Zustände der Zielgruppe ab (Heinecke 2004). Wenn möglich, sollten Objekte daher durch einfache Bilder visualisiert werden (Z. B. Dokument, Ordner, Person, Computer). Werkzeuge und Funktionen werden am besten durch eine Abbildung des Objekts und eine Aktion, wie beispielsweise einem Pfeil für das Öffnen eines Ordners dargestellt (Goodwin 2009). Abstrakte Symbole, wie beispielsweise eine Zylinderform als Abbild einer Datenbank sind schwieriger zu erlernen – wenn sie nicht bereits Teil des mentalen Modells einer Zielgruppe sind.

C. Icon: Resource Constraint

Das Icon steht für:

Ein Requirement, das
Aus Ressource-Begrenztheit resultiert (also bisher nicht explizit erfasst wurde);
Beispiel: EPROM ist in einem Projekt begrenzt;

Das Icon hat die Funktion:

die verschiedenen Requirement-Arten („Requiremental Items“) unterscheiden zu können; Also „Resource Constraints“ von „normalen“ Requirements und „Design Constraints“ abzugrenzen.
speziell: Ressourcen an EA-Elemente verteilen zu können;

Das Icon wird eingesetzt:

in der Liste der zugeordneten Requirements;
In der Decision-Anzeige;
als Butcon in der Toolbar;

Bitte bewerten Sie nun die Verständlichkeit jedes hier gezeigten Symbols, indem Sie den Prozentsatz der Grundgesamtheit an Personen eintragen, von dem Sie annehmen, dass das Symbol verstanden werden würde. Bitte bedenken Sie, dass 100% bedeutet, dass ALLE das Symbol verstehen würde, während 0% bedeutet, dass KEINER es verstehen würde.

Leervorlage/07.06.04/Revision_Vorlage: 1.8

zuletzt aktualisiert am: 04.03.2008 14:05:00
Erstellt von Goetzfried_Stefanie

Abb 2: Beispiel des *judgement tests* für "Resource Constraint"

Falls die Abbildung eines Icons durch die genannten Möglichkeiten nicht darstellbar sind, können rhetorische Mittel der natürlichen Sprache verwendet werden (bspw. Metonymie, Synekdoche oder Metaphern (Goodwin 2009), um eine verständliche Visualisierung zu erreichen.

Im Fallbeispiel wurden die Nutzer bereits bei dem ersten Brainstorming zur Findung geeigneter Signifikanten miteinbezogen. Dies geschah bewusst unabhängig von den in der Literatur genannten Repräsentanten-Empfehlungen. Die Nutzer sollten ohne Restriktionen ihre Assoziationen zu den teils sehr speziellen Funktionen äußern können. Zu diesem Zweck wurde ein kleiner Nutzer-

kreis zu einer Fokusgruppenrunde eingeladen, deren Ziel ein möglichst breites Spektrum an Darstellungsvarianten der Signaten war. Die Ideen wurden anschließend visuell aufbereitet, um sie in einem *judgement test* einem größeren Nutzerkreis vorzustellen und bewerten zu lassen (Rail Safety and Standards Board 2003).

3.3 Icon-Judgement Test

Nach dem ISO-Testverfahren für grafische Symbole (ISO - 9186-1+2 2007, 2008) erfolgt ein *icon judgement test* durch Abbildung möglicher Varianten der Signaten auf einem DIN A4 Blatt mit folgenden zusätzlichen Angaben:

- Name des Symbols
- Funktion des Symbols
- Einsatzort
- Ausgeschlossene Funktionen

Dieses Format wurde an die computer-spezifischen Bedingungen angepasst. In einer ersten Stufe wird den Teilnehmern ein Konzept vorgelegt, das beschreibt, welche Bedeutung das Symbol haben soll („Das Symbol steht für...“), welche Funktion das Symbol hat (bspw. „Unterscheidung“, „bloße Anzeige“, „Aktionsaufruf“) und wo es eingesetzt wird (bspw. „Toolbar“, „Treeview“). Darunter findet der Teilnehmer mehrere Darstellungsalternativen, wobei zusätzlich Platz für das Zeichnen eigener Ideen eingeräumt wird. Der Nutzer ist dann aufgefordert, jedes Symbol nach Verständlichkeit zu bewerten, indem er den prozentualen Anteil der Personen einträgt, von dem er annimmt, dass das Symbol von ihm verstanden werden würde (Abb 2).

Der *judgement test*-Zyklus erfolgt nach Bedarf iterativ, d. h. wird bei einem ersten Durchlauf keine Variante mit einem Median > 66% bewertet, muss erneut ein Fokusgruppentreffen einberufen werden, um neue Ideen zu finden sowie die weiteren Nutzervorschläge mit einzubeziehen. Überraschenderweise wur-

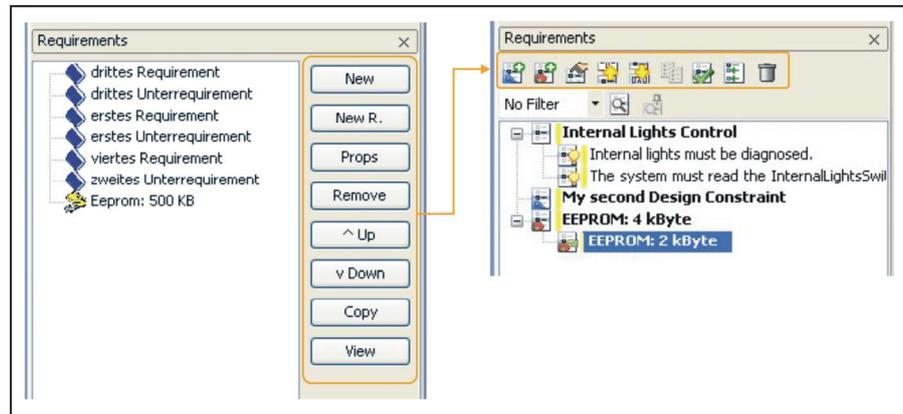


Abb 3: Vergleich des Vorher-/Nachher-Zustands am Beispiel "Requirement Tab"

de im Anwendungsfall der eingeräumte Platz für eigene Ideen des Nutzerkreises sehr gut angenommen, was zu sehr guten Ergebnissen in einem zweiten Durchlauf führte.

Im Gegensatz zum ISO-Testverfahren beinhaltet das hier vorgestellte Icon-testverfahren einen weiteren Zwischenschritt, bevor das Verständnis des besten Repräsentanten nochmals in einem *comprehension test* überprüft wird. Da der Mensch Bedeutung sowohl durch Inhalt als auch durch Kontext erfasst, kann die Bedeutung eines unbekanntes Icons auch durch dessen Anordnung in der Toolbar erschlossen werden (Barr et al. 2003). Daher werden im zweiten Teil des Tests die verschiedenen Varianten derjenigen Symbole, die über einer durchschnittlichen Verständnis-Wahrscheinlichkeitsrate von 66% lagen, in ihrer Gruppe und Originalgröße als Screenshot angezeigt und auf gleiche Weise bewertet.

3.4 Icon Comprehension Test

Beide Stufen werden durch einen abschließenden *icon comprehension test* verifiziert, in dem die Benutzer das Ergebnis-Icon in seiner Umsetzung sowohl direkt auf Verständnis prüfen und bewerten. Im Test wird von einem

Szenario ausgegangen, das der Zielgruppe aus dem Arbeitsalltag geläufig ist. Jeder Testperson werden dann die Realisierungsmöglichkeiten sowohl auf dem Display als auch auf Papier vorgelegt mit der Bitte, dem Testleiter laut mitzuteilen, was das Icon bedeuten könnte. Nach Bearbeitung aller zu testenden Icons wird das Gesamtkonzept erklärt, das mit einem kurzen abschließenden Fragebogen gesondert bewertet wird. Die erforderliche *success rate* (Median) eines Icons wird im *comprehension test* niedriger angelegt als im *judgement test*. Sie liegt bei einer Erkennungsrate von 50% und deckt sich damit mit den Empfehlungen der ISO bei grafischen Zeichen. Dies ist darauf zurückzuführen, dass das Erschließen einer Icon-Bedeutung ohne weitere Angabe schwieriger ist (in einer Toolbar könnten Tooltips weiterhelfen) als die Abschätzung einer Erkennungswahrscheinlichkeit unter Angabe der Anzeigebedingungen. Bedenkt man, dass sich die ISO-Vorgaben vor allem an die Entwicklung von sicherheitsrelevanten Zeichen richten, ist die hier angesetzte Schwelle der Bewertung bei Icons vergleichsweise hoch.

3.5 Auswahl nach Kriterien

Liegen mehrere Icons über der im *comprehension test* angesetzten 50%

Hürde, wird die beste Alternative in einem Expertenreview sondiert. Als Kriterien können herangezogen werden:

- Der Repräsentant mit dem höchsten durchschnittlichen Verständniswert
- Der Repräsentant, dessen Verwechslungsgefahr mit anderen Icons am niedrigsten eingeschätzt wird

Das letzte Kriterium kann zusätzlich mittels eines *mutual confusion tests* überprüft werden, in dem alle Icons des Systems auf einer gemeinsamen Anzeige dargestellt werden. Die Nutzer deuten nach Nennung des Objekts auf denjenigen Repräsentanten, der ihnen gemäß der Nennung als zugehörig erscheint.

4.0 Fazit

Im beschriebenen Fallbeispiel führte die abgewandelte Form der ISO Empfehlungen zu einer konsistenten Gesamtlösung (Abb 3): Objekte, die eine neue Funktion auszeichneten konnten nach Durchlauf zweier *judgement test* Zyklen und einer *comprehension test* Stufe ikonisch ausgezeichnet werden.

Zudem darf auf die sehr engagierte Teilnahme der Testpersonen hingewiesen werden: Der Testablauf stieß auf breites Interesse, was besonders durch das sehr hilfreiche aktive Mitwirken bei der Symbolfindung durch eigene Zeichnungen oder Ideen der Software-Designer und -Architekten zu tragen kam. Als Nebeneffekt führte die Testentwicklung damit auch zu einem verstärkten Verständnis für User Centered Design unter den Software-Ingenieuren.

Die Methode fällt damit gleichzeitig in das derzeit aktuelle Paradigma des *user generated content*: Die Zielgruppe ist unmittelbar in das Kommunikationsdesign eingebunden und trägt aktiv zur Ausgestaltung der visuellen Interaktionssprache bei.

Zu überlegen ist ferner, ob die Testmethodik auch auf andere Bereiche der visuellen Gestaltung übertragen werden kann, beispielsweise im Bereich visueller Symbole im Web Design oder bei visuellen Ordnungssystemen (Uebele 2006).

5.0 Literaturverzeichnis

Barr, P.; Noble, J.; Biddle, R. (2003): Icons R icons. In: Proceedings of the Fourth Australasian user interface conference on User interfaces 2003 - Volume 18, Adelaide, Australia: Australian Computer Society, Inc. S. 25-32.

Benbasat, I.; Todd, P. (1993): An Experimental Investigation of Interface Design Alternatives: Icon vs. Text and Direct Manipulation vs. Menus. In: International Journal of Man-Machine Studies 38 (1993), S. 369-402.

Chang, S.-K.; Yu, B.; Guercio, A.; Tortora, G. (1987): Icon Purity: Toward A Formal Theory Of Icons. In: International Journal of Pattern Recognition and Artificial Intelligence, 1(3) (1987), 377-392.

Coelho, E. (o.D.): Chrystal Icons. Abgefragt am 5.06.2009, von: <http://www.everaldo.com>

Ferreira, J.; Noble J.; Biddle R. (2006): A case for iconic icons. In: Proceedings of the 7th Australasian User interface conference - Volume 50, Hobart, Australia: Australian Computer Society, Inc., S. 97-100.

Goodwin, K. (2009): Designing for the Digital Age: How to Create Human-Centered Products and Services. Indianapolis: Wiley & Sons.

Heinecke, A. (2004): Mensch-Computer-Interaktion. München: Hanser Fachbuchverlag.

ISO - International Organization for Standardization (Hrsg.) (2007, Januar): Graphical Symbols - Test Methods - Part 1: Methods for testing comprehensibility. Abgefragt am 03.06.2009, von: http://www.iso.org/iso/iso_catalogue/catalogue_ue_tc/catalogue_detail.htm?csnumber=41090

ISO - International Organization for Standardization (Hrsg.) (2008, Juni): Graphical Symbols - Test Methods - Part 2: Methods for testing perceptual quality. Abgefragt am 3.06.2009, von:

http://www.iso.org/iso/iso_catalogue/catalogue_e_tc/catalogue_detail.htm?csnumber=43484

Nielsen, J.; Sano, D. (1994): 1994 Design of SunWeb: Sun Microsystems' Intranet. Abgefragt am 3.06.2009, von: <http://www.useit.com/papers/sunweb/>

Rail Safety and Standards Board (Hrsg.) (2003): Engineering. Improvements to safety signage on passenger trains. Guidelines for symbol design and testing, London: Davis Associates for Interfleet Technology Ltd.

Rohr, G. (1984): Understanding Visual Symbols. In: Proc. of the 1984 IEEE Computer Society Workshop on Visual Languages. Hiroshima, Dezember 1984. Silver Spring/MD: IEEE Computer Society Press, S. 184-191.

Turban, B.; Kucera, M.; Tsakpinis, A.; Wolff, C. (2009): Bridging the Requirements to Design Traceability Gap. In: R. E. D. Seepold & N. Martínez Madrid (Hrsg.), Intelligent Technical Systems. Berlin et al.: Springer, S. 275-288.

Uebele, A. (2006): Orientierungssysteme und Signaletik. Ein Planungshandbuch für Architekten, Produktgestalter und Kommunikationsdesigner. Mainz: Verlag Hermann Schmidt.

Whiteside, John et al. (1985): User Performance with Command, Menu, and Iconic Interfaces. In: Borman, L.; Curtis, B. (Hrsg.) (1985). Human Factors in Computing Systems-II. Proc. of the CHI '85 Conference. Amsterdam et al: North-Holland (Elsevier), S. 185-191.

Usability Speed Dating