

Sascha Wolter

Jenseits mobiler Anwendungen: Super-natural interaction und Spielzeuge

Telekommunikation_SMS_Prototyping_Super-natural interaction_Internet of Things

Zusammenfassung. Das Internet der Dinge ist schon jetzt allgegenwärtig und Apps beschränken sich nicht mehr auf traditionelle PCs und mobile Geräte. Eine SMS schickende Kuh, der Nachrichten austauschende Müllwagen und das telefonierende Projektmanagement-Werkzeug sind nur einige Beispiele. Sowohl industrielle Lösungen als auch Alltagsgegenstände werden immer interaktiver und vernetzter, so dass der Nutzer nicht mehr nur mit einem einzelnen Gerät interagiert, sondern sich inmitten einer interaktiven Umwelt bewegt. Unternehmen

und Entwickler müssen sich dieser Herausforderungen stellen und unterschiedlichste Geräteformen, Interaktionsformen und Netzwerke miteinander kombinieren. Doch noch wichtiger ist es, überhaupt erst einmal neue Ideen und Szenarien für diese Anforderungen zu entwickeln. Hier sind Entwickler und Gestalter zentrale Innovationstreiber, weshalb viele Unternehmen ihre bestehenden Dienste möglichst entwicklerfreundlich und standardisiert als APIs öffnen.

1. Super-natural interaction

Schon jetzt durchdringen Navigationssysteme, Smart Phone, Smart TV, Staubsaugerroboter und andere intelligente Systeme unser Leben. Darüber hinaus schafft die Kombination dieser Geräte neue komplexe Systeme: Wo früher einzelne Haushaltsgeräte durch eingebettete Computer in unserer Umgebung lokal genutzt wurden, sind diese zukünftig global über das Internet vernetzt. Diese interaktiven Systeme dringen in immer unterschiedlicheren Geräteformen in immer mehr Lebensbereiche vor. Ein solches interaktives System setzt sich aus einer Vielzahl weiterer Systeme zusammen (Stary, 1996). Dazu zählen der Anwender, die Anwendungs-Software, das Betriebssystem, das Hardware-System und das Netzwerk-System (siehe Abbildung 1).

Die Schnittstelle zwischen Nutzer und Anwendung, die Interaktivität inkl. Sensoren und Aktuatoren (ein Aktuator wandelt elektrische Signale meist in mechanische Arbeit um und bildet somit das Gegenstück zum Sensor) werden dabei immer vielfältiger und „intelligenter“; vernetzte elektronische Geräte reagieren zunehmend auf Ihre Umge-

bung (*Ambient intelligence*). Je nach Umgebung kommt der Wahl des Netzwerk-Systems ebenfalls eine wichtige Rolle zu – ob also z.B. ein GSM-Netz oder DSL verfügbar sind, ob eine SMS geschickt oder IP-basierte Nachrichten übertragen werden. Das ist keine einfache Entscheidung, da die Verfügbarkeit, Vielfalt und Leistungsfähigkeit von Netzwerken regional variieren und sowohl Geschwindigkeit als auch Energieverbrauch von Bedeutung sind. Das

Datenvolumen ist ebenfalls ein Aspekt, welcher nicht nur für die soziale Interaktion, sondern auch für industrielle Anwendungen von großer Bedeutung ist. Zum Vergleich: Die SMS hat ein Datenvolumen von gerade einmal 1/1000 gegenüber einer Gesprächsminute (Wikipedia, 2013)!

Die Durchdringung des Alltags mit Computern, die Vernetzung dieser Geräte untereinander und die Möglichkeiten in der Mensch-Maschine-Interaktion for-

Interaktives System



Abbildung 1: Interaktives System.

cieren einen Paradigmenwechsel weg von der über Jahrzehnte erlernten 1:1-Beziehung zwischen Mensch und Maschine hin zu komplexen 1:n Systemen aus vielen vernetzten Geräten in deren Mittelpunkt der Nutzer steht (ganz zu schweigen von kollaborativen Systemen mit mehreren Nutzern). Anders als bisher üblich interagiert der Nutzer nicht mehr mit einem einzelnen Gerät über dessen Bildschirm, sondern mit zahlreichen vernetzten Geräten die nicht zwangsläufig über einen Bildschirm verfügen (Headless System). Unsere gesamte Umgebung wird laut Wilson (2012) zur Benutzungsschnittstelle („*Super-natural interaction*“, siehe Abbildung 2).

Diese Vielfalt erfordert neue Herangehensweisen, um frühzeitig deren Nutzen im Kontext evaluieren zu können. Andernfalls wird die *Kluft zwischen Mensch und Maschine* eher zunehmen als verringert (Norman & Draper, 1986, siehe Abbildung 3).

Dabei ist nicht nur die Spezifizierung von Absichten durch den Nutzer sondern auch die Auswertung der Ergebnisse von Bedeutung – zumal eben nicht mehr nur mit einem einzelnen Gerät mit Bildschirm interagiert wird. Auch wenn laut Reisinger (2012) Smartphones und Tablets bis 2015 noch rund 80 % der Entwicklungstätigkeit bestimmen, gehen (nicht nur) Entwickler bereits jetzt davon aus, dass Anwendungen zukünftig für weitere Geräteformen erstellt werden (Appcelerator, 2012), darunter Smart TV, Connected Car, Spielkonsolen und Google Glass.

Und immer mehr dieser Geräte sind miteinander verbunden: Die OECD, (2012) erwartet, dass die Anzahl der Geräte allein im Bereich der Maschine-Mit-Maschine-Kommunikation (M2M) von heute rund 5 Milliarden auf 50 Milliarden im Jahr 2020 wächst. Der Umsatz soll laut Forrester Research bereits 2016 von heute 4,2 Milliarden auf dann 17 Milliarden US Dollar steigen (Forrester, 2011). Und ein Ende dieses Wachstums ist nicht in Sicht. Eine wesentliche Rolle bei dieser Entwicklung könnte dem Notrufsystem eCall (kurz für emergency call) zuteilwerden (Wikipedia, 2013). Denn ab 2015 muss jedes in der Europäischen Union neu zugelassene Fahrzeug über eine Notruffunktion verfügen, wodurch sich



Abbildung 2: Projekte rund um „Super-natural interaction“. Sketchnote der spielerischen Herangehensweise (Alt, 2013).

Kluft zwischen Mensch und Maschine

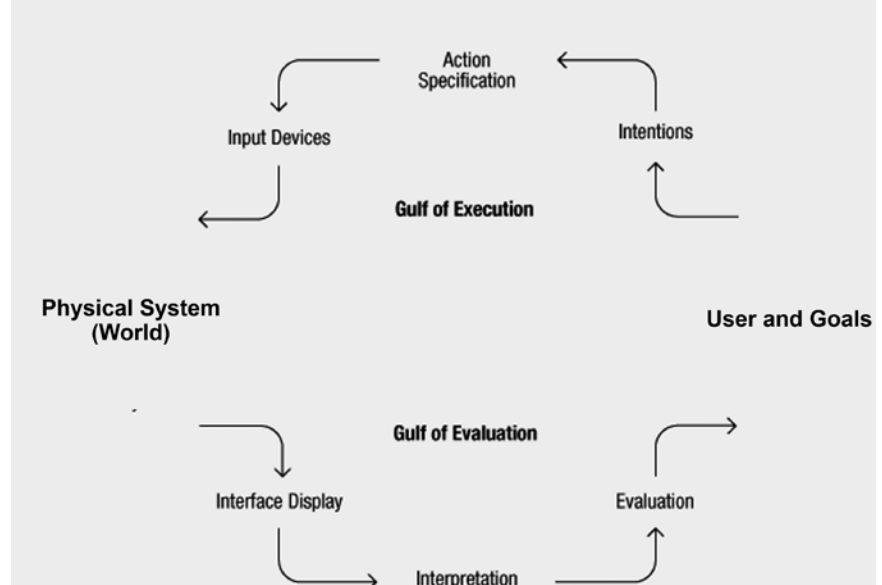


Abbildung 3: Kluft zwischen Mensch und Maschine (nach Norman & Draper, 1986, S. 111).

jeder neue PKW in ein vernetztes und mobiles Endgerät verwandelt.

Die Schlüssel zu neuen und erfolgreichen Anwendungen im Sinne der „Super-natural interaction“ sind die für die Umsetzung notwendigen Ressourcen (u.a. Entwickler) und damit einhergehend Vorgehensmodelle (z.B. Prototyping), die helfen in diesem doch häufig noch unbekannten Terrain der Benutzererlebnisse frühzeitig Ergebnisse zu liefern.

2. Entwickler

Gerade Telekommunikationsunternehmen (kurz Telcos) beschäftigen sich intensiv mit der Vernetzung von Geräten, dem sogenannten Internet der Dinge (Internet of Things). Sie suchen nach den nächsten Formen der Interaktion, Kommunikation und Vernetzung. Bestehende Dienste wie z.B. der Kurznachrichten SMS verfügen hier durchaus noch über eine tragende Rolle, um Innovationen zu

schaffen. Das ist auch ökonomisch naheliegend, denn wenn man den Markt der Mobilfunkgeräte betrachtet, ist es offensichtlich, dass allein Endkunden (Konsumenten) nur noch geringes Wachstumspotential bieten. Schließlich gab es 2012 z.B. allein in Deutschland laut der [Bundesnetzagentur, 2012] bereits mehr als 114 Millionen SIM-Karten bei nur gut 80 Millionen Einwohnern. Hier gilt es, neue Einsatzgebiete zu finden. Auch Geräte und Tiere können zu Mitgliedern einer vernetzten Gesellschaft werden und miteinander Informationen austauschen, beispielsweise regelmäßige Information über die Körpertemperatur der Kuh per SMS, zur Abschätzung des Zeitpunkts einer bevorstehenden Geburt (siehe Abbildung 4). Der „simsende Bienenstock“ ist schon jetzt Realität (siehe Kapitel 4).

Doch ohne passend qualifizierte Entwickler in ausreichender Menge fehlt es an der Möglichkeit, diese Form der Innovationen überhaupt umzusetzen. Dies gilt für alle Unternehmen. Außerdem ziehen Entwickler auch externe Investoren an, die Innovation und Wachstum finanzieren (Developer Economics, 2012). Der Entwickler wird zum Prosumer (Consumer und Producer): Sprich, der Entwickler konsumiert die Dienste eines

Telcos und produzieren auf dieser Basis neue Lösungen. Der Wettbewerb um die Entwickler hat durch die Vielzahl der Ökosysteme mit ihren App-Stores zugenommen: Laut Developer Economics (2012) haben Telcos massiv an Einfluss verloren und erreichen nur noch rund drei Prozent der Entwickler. Die Entwicklung des Arbeitsmarktes tut ein Übriges dazu. Laut BITKOM (2012) ist die Anzahl offener Stellen in Deutschland im IT-Bereich allein in 2012 um 13 Prozent gestiegenen (75 % der offenen Stellen richten sich an Softwareentwickler).

Um für Entwickler an Attraktivität zu gewinnen, bieten viele Unternehmen speziell auf Entwickler zugeschnittene Portale und machen darüber APIs zugänglich. Die Deutsche Telekom bietet beispielsweise unter <http://developer.garden.com/> zahlreiche APIs und Services – darunter Empfang und Versandt von SMS, MMS, Spracherkennung/Telefonie und M2M. Dort finden sich auch ein Blog mit themenübergreifenden Inhalten und eine Community mit Events und Diskussionsforen (Englisch und Deutsch). Darüber hinaus verfolgt der Developer Garden eine Enabling-Strategie in Zusammenarbeit mit Inkubatoren wie hub:raum.

Um den umworbenen Entwickler für sich zu gewinnen, müssen die Angebote möglichst einfach zugänglich sein und die Opportunitätskosten gering gehalten werden (Developer Economics, 2012). Viele Anbieter setzen auf offene Standards (beispielsweise REST und JSON) und arbeiten mit den entsprechenden Organisationen zusammen.

Außerdem hilfreich sind spezielle Entwickler, die als Sprachrohr zwischen dem Anbieter und den Nutzern der APIs und Services agieren. Diese im angelsächsischen Raum auch Ambassador oder Evangelist getauften Entwickler nutzen die Angebote in eigenen Projekten und arbeitet so bereits frühzeitig an einer stetigen Verbesserung mit. Sie tragen die Visionen und Begeisterung in der Sprache der Entwickler nach außen und kommunizieren die Erkenntnisse aus der Projekt-Praxis zurück ins Team.

3. Prototyping und Innovation

Das reine Angebot von APIs und Services ist jedoch nicht ausreichend, um neue Ideen zu entwickeln und zu evaluieren. Letztendlich müssen Entwickler motiviert werden, sich damit aktiv zu beschäftigen. Der Softwareanbieter Atlassian hat mehrere Möglichkeiten zur Entwickler-Motivation unter zusammengefasst (Peters, 2011). Speziell zeitlich beschränkte Experimentierphasen haben sich etabliert. In Anlehnung an die Lieferzeit von Paketzustellern werden diese als FedEx Days bezeichnet. Das Konzept ähnelt sogenannten Hackathons. Diese haben jedoch meist einen Event-artigeren und offeneren Charakter und werden vom Anbieter der APIs durch Evangelisten (siehe oben) betreut. Diese Idee der betreuten Projekte kann aber auch gemeinsam mit dem Kunden im Sinne der Ideenfindung und des Business Developments durchgeführt werden.

Um der Vielfalt der Möglichkeiten und Anforderungen gerecht zu werden, sollten auch iterative Vorgehensmodelle wie Prototyping beherrscht werden. Ganz im Sinne der Agilität müssen Individuen und Interaktionen mehr gelten als Prozesse und Werkzeuge. Ein Durchstich ist wichtiger als die Dokumentation. Der

Kühe können SMS senden

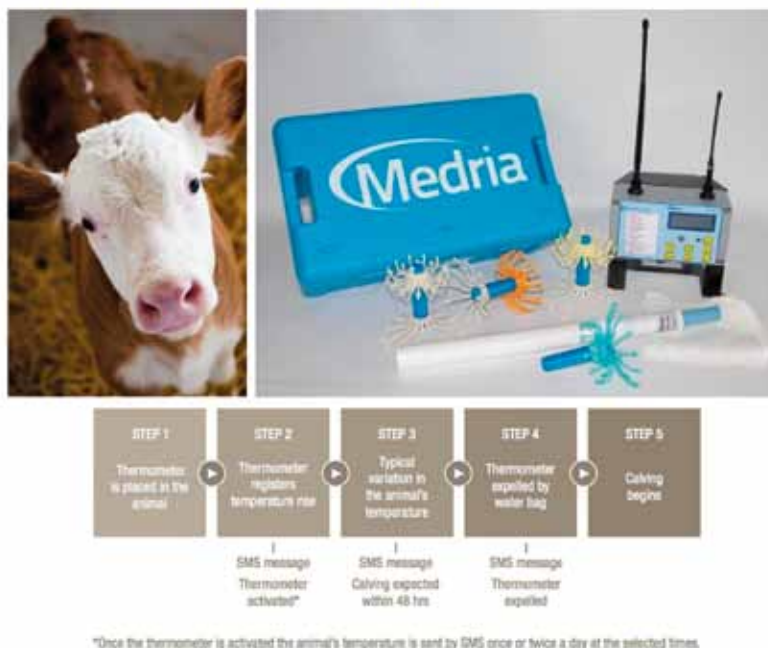


Abbildung 4: SMS sendende Tiere sind schon Realität (Medria, 2012).

Anwender und der Nutzen befinden sich im Mittelpunkt. Und der Mut und die Offenheit für Änderungen stehen über dem Befolgen eines festgelegten Plans. Letztendlich ist diese Herangehensweise ein sehr natürlicher Prozess, der dem kindlichen Spielen mit Lego entspricht, so wie von Jonathan Gay beschrieben (Gay, 2001, siehe Abbildung 5):

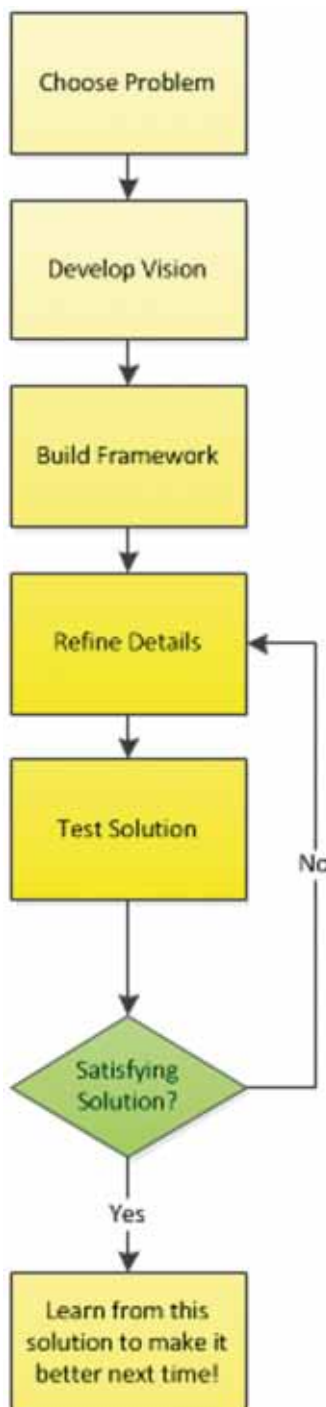


Abbildung 5: LEGO-basierter Design Prozess nach [Gay, 2001].

1. Choose a problem: Build a LEGO ship.
2. Develop a vision: What sort of ship will it be? How big will it be? What will it carry?
3. Build: Build the framework of the ship.
4. Fill in the details: Design and build the details of the ship, ramps, doors, etc.
5. Test: Drive the cars around the ship and sail the ship while exploring the house.
6. Refine: Take parts of the ship apart and make them better.
7. Learn: Take what you learned from building this ship and use it to build a better one next time.

Doch auch für diese Herangehensweise müssen Rahmenbedingungen geschaffen werden, um zu einem befriedigenden Ergebnis zu kommen. Zum einen muss die Erwartungshaltung klar spezifiziert werden. Und die Bausteine für die Durchführung müssen vorbereitet sein: APIs und Services werden als einfach zu nutzenden Code-Blöcken vorbereitet. Je nach Projekt kann auch ein Modell-basiertes Entwurfsmuster wie MVC, MVVM oder das Presentation Model zum Einsatz kommen. Dies entspricht der Idee der Vorhangfassade aus der Architektur (siehe Abb. 6): Der Rohbau zur Nutzung der APIs ist vorbereitet und die eigentliche Anwendung wird wie eine Fassade nur noch an den Anschlusspunkten befestigt.

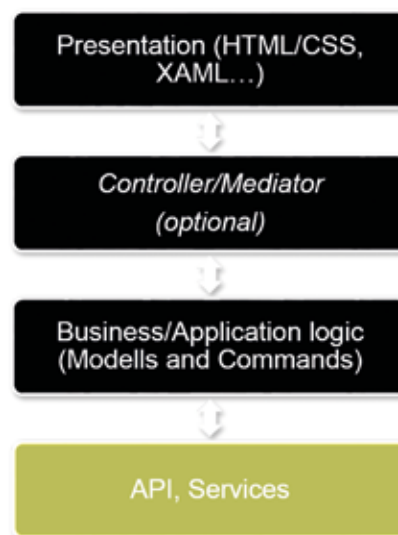


Abbildung 6: Softwarearchitektur im Sinne einer Vorhangfassade.

Auf Seiten der Hardware kann der Aufwand ebenfalls minimiert werden, indem man auf bereits existierende Produkte zurückgreift. Insbesondere Spielzeuge bieten sich für Prototypen an: Sie sind in der Regel kostengünstig und leicht zu modifizieren (siehe Abb. 7).



Abbildung 7: Zahlreiche interaktive Spielzeuge lassen sich leicht und kostengünstig in Prototypen verwenden.

4. Der simsende Bienenstock

Wie einfach sich das Internet der Dinge realisieren lässt zeigt das Beispiel eines simsenden Bienenstocks. Für die einfachste Variante ist nur eine elektronische Waage notwendig, um die Aktivität und Veränderungen des Bienenvolkes zu messen (bessere Systeme sammeln zusätzlich noch Umweltdaten). Für einen Prototypen kann man sich auch eines Wii Balanceboards bedienen, möchte man erst einmal in der Welt der Spielzeuge bleiben und diese per Bluetooth an einen Rechner koppeln. Dieser Rechner macht nichts anderes, als eine SMS mit den aktuellen Daten zu senden. Falls ein moderner Einplatinenrechner bzw. ein Embedded Device wie ein Arduino mit einem GSM-Modul zum Einsatz kommt, sieht der Code dank standardisierter AT-Befehle im Wesentlichen aus wie in Abbildung 8, linke Spalte (siehe auch Wolter, 2013):

Die Verarbeitung der Daten kann dann direkt von einem Service wie der Global SMS API (2013) übernommen werden, der als virtuelle Telefonnummer agiert und wie ein normaler Webserver programmiert wird. Die Angabe der URL zum Empfang einer

```

void sendMessage() {
    // Because we want to send
    // the SMS in text mode
    mySerial.print("AT+CMGF=1\r");
    delay(200);
    // Send message
    mySerial.print("AT+CMGS=\"+49151987654321\"\r");
    delay(200);
    // Content of the message
    mySerial.print("Alarm, something happened!");
    delay(200);
    // Finish message and use
    // the ASCII code for ctrl+z
    mySerial.print((char)26);
}

{
    "inboundSMSMessageNotification" : {
        "inboundSMSMessage" : {
            "dateTime" :
                "2013-06-03T11:43:10",
            "destinationAddress" :
                "tel:+4915112345678",
            "messageId" :
                "2c5a421a05764c11b7a9093769d2b7",
            "message" :
                "Greetings to Joe Doe",
            "senderAddress" :
                "tel:+4915187654321"
        }
    }
}

$postdata =
    file_get_contents("php://input");
$json_result =
    json_decode($postData, false);
$message =
    $json_result->
    inboundSMSMessageNotification->
    inboundSMSMessage->
    message;

```

Abbildung 8: Code zur Programmierung des simsenden Bienenstocks

SMS erfolgt im Falle der Global SMS API im Entwicklerkonto des Developer Garden (www.developergarden.com/de/mein-konto/api-verwaltung/). Dort muss der Service nur aktiviert und eine Rufnummer zum SMS-Empfang beantragt werden. Hier gibt man auch direkt die URL für den Empfang von der SMS-Details an. Die Daten einer Kurznachricht werden dann im gängigen JSON-Format dorthin übermittelt (Abb. 8, Mitte). Für die anschließende Verarbeitung reichen ein paar wenige Zeilen (hier als Beispiel PHP, Abb. 8, rechte Spalte).

Auf diese Art und Weise steht einer interaktiven Steuerung von Anwendungen und Maschinen durch den Empfang von SMS und einer Benachrichtigung von Nutzern durch den Versand von Nachrichten (Push-Notifications) nichts mehr im Wege; Plattformabhängig und teure Apps sind damit obsolet, denn das funktioniert mit jedem SMS-fähigen Gerät.

5. Bedarf wecken

Die Zutaten für einen erfolgreichen Umgang mit „Super-natural interactions“ sind überschaubar. Motivierte und kompetente Entwickler, einfach zu verwendende Bausteine in Form von Software (APIs und Services), Konnektivität und leicht zugängliche Hardware für Interaktion und Sensorik (siehe Bild 1). Das gewürzt mit einer iterativen und prototypischen Herangehensweise erlaubt es auch in unbekanntem Terrain anhand von praktischen Beispielen zu forschen, ohne unnötig Ressourcen zu vergeuden und sich durch industrielle Zwänge

zu beschränken. Dafür entsprechen die Ergebnisse nicht unbedingt den Anforderungen eines Produktes – es ist eher das Paretoprinzip, das hier zum Tragen kommt.

Die Idee der spielerischen Herangehensweise auf Basis von vorgefertigten Bausteinen zeichnet sich aber nicht nur dadurch aus, dass man schnell zu Ergebnissen gelangt. Neben der eigentlichen Ideenfindung und schnellen Evaluierung hat dieses Vorgehen noch einen weiteren Vorteil: Während der aktiven Auseinandersetzung mit „Super-natural interactions“ werden Bedürfnisse überhaupt erst geweckt, die vorher noch gar nicht bewusst waren...

Literatur

- Alt, B.: UX Barcamp Europe 2013 - Toys are us, <http://www.flickr.com/photos/83052714@N05/9177425662/> (Stand 06/13)
- Appcelerator/IDC Q3 2012 Mobile Developer Report, <https://pages.appcelerator.com/Q32012AppceleratorIDCSurveyReport.html>
- Bitkom: 43.000 offene Stellen für IT-Experten, http://www.bitkom.org/de/themen/54633_73892.aspx (Stand 10/12)
- Bundesnetzagentur: Wettbewerbsintensität im Mobilfunk nimmt weiter zu, http://www.bundesnetzagentur.de/cln_1931/Shared-Docs/Pressemitteilungen/DE/2012/120824_WettbewerbMobilfunk.html (Stand 08/12)
- Developer Economics: The new mobile app economy, <http://www.developereconomics.com> (Stand 06/12)
- Forrester Research, M2M Connectivity Helps Telcos Offset Declining Traditional Services, <http://www.forrester.com/M2M+Connectivity+Helps+Telcos+Offset+Declining+Traditional+Services/fulltext/-E-RES56893?docid=56893> (Stand 12/11)

- Gay, J.: The History of Flash, http://www.adobe.com/macromedia/events/john_gay/ (Stand 01)
- Media: <http://www.media.fr> (Stand vom 11/12)
- Norman, D. A.; Draper, S. W. (Eds.) User Centered System Design: New Perspectives on Human-computer Interaction. CRC Press, 1986
- OECD: Machine-to-Machine Communications: Connecting Billions of Devices, OECD Digital Economy Papers, No. 192, OECD Publishing. <http://dx.doi.org/10.1787/5k9gsh2gp043-en> (Stand 01/12)
- Peters, S.: Motivation für Softwareteams, http://svenpet.com/2011/12/06/motivation_softwareteam/ (Stand 12/11)
- Reisinger, D.: Gartner Enterprise Apps Slideshow: Mobile Application Development: A Top CIO Priority, <http://www.cioinsight.com/c/a/Enterprise-Apps/Mobile-Application-Development-A-Top-CIO-Priority-895960/> (Stand 07/13)
- Global SMS API, <http://www.developergarden.com/de/apis/apis-sdks/global-sms-api/> (Stand 07/13)
- Stary, C.: Interaktive Systeme. 2. Auflage, Vieweg, 1996.
- Telekom Tropo API, <http://www.developergarden.com/de/apis/apis-sdks/telekom-tropo-api/> (Stand 07/13)
- Wolter, S. 2013: Connected Things mit Arduino und (fast) ohne Internet, <http://www.wolter.biz/2013/09/connected-things-mit-arduino-und-fast-ohne-internet/> (Stand 09/13)
- Wikipedia: eCall, <http://de.wikipedia.org/wiki/ECall> (Stand 07/13)
- Wikipedia: Short Message Service, https://de.wikipedia.org/wiki/Short_Message_Service (Stand 07/13)
- Wilson, A.: Super-natural interaction. Microsoft Build Konferenz, Redmond USA, <http://channel9.msdn.com/Events/Build/2012/2-007> (Stand 10/12)



Sascha Wolter

studierte Informatik und Psychologie mit dem Schwerpunkt Benutzungs-schnittstellen an der Universität Paderborn. Er ist Experte für die Planung und Umsetzung von reichhaltigen und plattformunabhängigen sowie geräteübergreifenden Anwendungen (Desktop, Mobil, Web und M2M). Seit 1995 arbeitet er als Berater, Autor, Trainer, Entwickler und Software-Architekt für zahlreiche Firmen. Darunter Adobe, Deutsche Telekom, Microsoft, SAP und Vodafone. Als Gründer der Adobe User Group flashforum.de mit mehr als 100.000 Mitgliedern und als Mitgründer der international bekannten Konferenz beyond tellerrand engagiert sich Sascha auch für die Belange der Anwender. Wenn er nicht gerade als Developer Evangelist neue Möglichkeiten für die Deutsche Telekom AG (Developer Garden) erkundet, dann spielt er in seiner Freizeit mit seinen Kindern Lego.

E-Mail: s.wolter@telekom.de
(<http://wolter.biz>)