

# Micro Online Reverse Auctions für allgegenwärtige C2B-Koordination

Simon Vogt

Institut für Informatik  
Helmut-Schmidt-Universität Hamburg  
Holstenhofweg 85  
22043 Hamburg  
simonvogt@hsu-hamburg.de

**Abstract:** Dieser Beitrag untersucht die Nutzung von Online Reverse Auctions in Verbindung mit dem Konzept des Ubiquitous Computing um eine Echtzeit-Koordination von alltäglichen Markttransaktionen im Business-to-Consumer-Kontext zu ermöglichen. Dazu werden die Eigenschaften von pervasiven Systemen untersucht und darüber hinaus ein geeigneter Koordinationsmechanismus von Angebot und Nachfrage identifiziert. Daraus ergeben sich die Anforderungen an ein Web-basiertes System, welches implementiert, beispielhaft veranschaulicht und abschließend bewertet wird.

## 1 Einführung

*“This is one of the areas of Web 2.0 where we expect to see some of the greatest change, as more and more devices are connected to the new platform. What applications become possible when our phones and our cars are not consuming data but reporting it?” [Or05]*

Tim O’Reillys Definition des Begriffes “Web 2.0” aus dem Jahr 2005 betrachtet bereits das Konzept von „software above the level of a single device“. Sie betrachtet dies als einen Schlüsselfaktor für die zukünftige Entwicklung, obwohl Smartphones und Tablet-PCs zur damaligen Zeit sehr selten waren. Nach nun sieben Jahren wird zunehmend absehbar, wie die Antwort auf O’Reillys Frage aussehen könnte. Während des dritten Quartals 2011 stiegen die Smartphone-Verkäufe weltweit auf 115 Millionen Stück, ein Zuwachs von 42 % verglichen mit dem Vorjahreszeitraum [GR11]. Begleitet wird diese Entwicklung durch einen wachsenden Wettbewerb auf dem Sektor der Tablet-PCs (insb. durch Apple, Google und Amazon). Die steigende Nutzung mobiler Geräte unterstützt die existierenden Web-2.0 Anwendungen wie soziale Netzwerke und Microblogs. Das Veröffentlichen aktueller Aktivitäten, Orte, Gedanken und Bedürfnisse mit etwa Facebook, Twitter und Google+ ist inzwischen alltäglich geworden. Darüber hinaus werden die Funktionen von sozialen Netzwerken nun durch mobile Geräte stark erweitert, da diese das Teilen von exakten GPS-Positionen [Bi10], Kamera-Schnappschüssen und Videos erlauben. Die Kombination der mehr und mehr allgegenwärtigen mobilen Geräte mit Social-Media-Anwendungen eröffnet eine neue Dimension des *Ubiquitous Computing*.

Bereits während der vergangenen drei Dekaden sind elektronische Märkte zu einer populären Plattform für die Abwicklung von Transaktionen und Handel geworden [PSV03]. Das Internet als Werkzeug und Katalysator für die Koordination zwischen Käufern und Abnehmern hat die Effektivität von Preis- und Wertverhandlungen enorm gesteigert. Essentielle Koordinationsmechanismen in diesem Kontext sind Auktionen, Ausschreibungen und Kleinanzeigen. Diese profitieren besonders von den Charakteristika des Online-Marktes, brechen die Einschränkungen von Ort und Zeit und senken die Zutrittsbarrieren für potentielle Teilnehmer. Diese Attribute werden durch die skizzierte stark steigende Verbreitung von mobilen Endgeräten weiter potenziert und es ergeben sich mannigfaltige neue Anwendungsfelder. Eines davon ist die Echtzeit-Abwicklung von Markttransaktionen im Alltag und somit im Business-to-Consumer-Kontext. Hieraus ergibt sich für diesen Artikel die Forschungsfrage, wie ein konkretes plattformübergreifendes und omnipräsentes System für die Vereinfachung der Markt-Koordination in Echtzeit zwischen Anbietern und Nachfragern in alltäglichen Situationen aussehen kann.

Dazu wird ein Design-orientierter wissenschaftlicher Ansatz verfolgt. Zu Beginn werden Lücken und Ansatzpunkte literaturbasiert identifiziert und daraus neue Perspektiven und Anforderungen an das zu entwickelnde System abgeleitet. Dieses wird anschließend entworfen und mit Hilfe eines Fallbeispiels implementiert und demonstriert. Abschnitt 2 gibt einen Überblick über den Begriff des „Ubiquitous Computing“ und leitet daraus Anforderungen an den späteren Entwicklungsprozess ab. Abschnitt 3 untersucht und identifiziert Methoden für effektive Marktkoordination, um das passende Mittel für das Systemdesign zu bestimmen, welches dann in Abschnitt 4 entwickelt wird. Das resultierende Konzept wird nachfolgend implementiert und anhand eines Fallbeispiels demonstriert.

## **2 Pervasive und Ubiquitous Computing**

### **2.1 Begriffsklärung**

Das Konzept des „Ubiquitous Computing“ beschreibt die Omnipräsenz von Informationstechnologie und Informationsverarbeitung im täglichen Leben [We91]. Computer und deren Nutzung werden so stark wie möglich in Objekte, Prozesse und menschliche Verhaltensweisen eingebunden. Die Prämisse von Ubiquitous- oder Pervasive Computing ist, Technologie unserem Leben anzupassen [WB96], indem die Interaktion zwischen Menschen und deren Umgebung vereinfacht und unterstützt wird [HSF11]. Da Weiser keine klare Abgrenzung oder Definition der Begriffe „Pervasive Computing“ und „Ubiquitous Computing“ gegeben hat, variieren die Interpretationen. In diesem Aufsatz werden diese jedoch synonym verwendet. Hoffmann et al. haben diesbezüglich eine Übersicht über verschiedene Definitionsmöglichkeiten und gemeinsame Schlüsselfaktoren erarbeitet [HSF11]:

- **Aufmerksamkeitsschwerpunkt:** Die Wahrnehmung der Technologie durch die Nutzer soll durch intuitive Nutzeroberflächen verringert werden.

- **Kontextwahrnehmung:** Die Systeme fühlen Informationen der technischen und physikalischen Umwelt selbsttätig und konfigurieren sich entsprechend.
- **Einfachheit:** Programme und Nutzeroberflächen können ohne technisches Know-How bedient werden.
- **Mobilität:** Auf Geräte und Anwendungen kann überall zugegriffen werden, während sich diese automatisch anpassen.
- **Kommunikation und Konnektivität:** Bidirektionaler Austausch zwischen Hard- und Software.

Zusätzlich spielen die Faktoren implizite Systemeingaben/Automated Capture eine Rolle in einigen Definitionen. All diese Faktoren sollten in der Entwicklung von mobilen Ubiquitous-Computing-Systemen beachtet werden (vgl. Abschnitt 5), woraus sich spezielle Anforderungen und Aufgaben für den Entwicklungsprozess an sich ergeben.

## **2.2 Entwicklung sozio-technischer Anwendungen**

Basierend auf den im vorherigen Abschnitt erwähnten Schlüsselfaktoren ergeben sich spezielle Aufgaben und Hürden bezüglich der Entwicklung von sozio-technischen, pervasiven Anwendungen. Diese benötigen eine leicht zu bedienende oder gar unsichtbare Nutzeroberfläche/Schnittstelle. Folglich erfordert das Design ein hohes Maß an Innovation und Kreativität. Darüber hinaus sind nahezu unsichtbare Nutzerschnittstellen schwer darstell- und demonstrierbar, da sie eine versteckte Funktionalität zum Ziel haben. Neben diesen Herausforderungen an ein angemessenes Design und dessen Evaluation bestehen einige weitere Faktoren, die als einzigartig in diesem Kontext anzusehen sind: Mobile pervasive Systeme streben die komplette Integration in das alltägliche Leben an [We91], was eine Tendenz zu großskaligen, nahezu unsichtbaren und stark vernetzten Modulen anstatt zu kleinen Anwendungen mit überschaubarem Horizont impliziert. Offenkundig haben solche Systeme einen starken Einfluss auf soziale Strukturen und Umfelder und benötigen daher ein entsprechendes Maß an Vertrauen und Akzeptanz der involvierten Nutzergruppen [La03]. Der größtenteils verdeckte Austausch von Nutzerdaten kann zudem Bedenken bezüglich des Datenschutzes hervorrufen [Sh10]. In der weiteren Konsequenz führt dieser Gedanke zu dem Einfluss von Gesetzen und Vorschriften, welche definitiv bereits im Entwicklungsprozess berücksichtigt werden müssen [HSF11]. Der folgende Abschnitt untersucht Methoden für effiziente Marktkoordination mit dem Ziel ein geeignetes Instrument für eine pervasive Echtzeit-Anwendung zu identifizieren und später in das Systemdesign (vgl. Abschnitt 4) einzubinden.

## 3 Mechanismen für effiziente Markt-Koordination

### 3.1 Phasen einer Markttransaktion

Für die Beschreibung der Phasen einer Markttransaktion existieren diverse Theorien und Modelle. Guttman et al. führen diese mit speziellem Fokus auf das Verhalten des Konsumenten zum „Consumer Buying Behavior Model“ (CBB-Modell) zusammen [GMM98]. Dabei können im Wesentlichen sechs fundamentale Phasen des Markttransaktionsprozesses (aus Konsumentensicht) differenziert werden:

1. Need Identification (Bedürfnisidentifizierung)
2. Product Brokering (Suche nach geeignetem Produkt)
3. Merchant Brokering (Suche nach geeignetem Anbieter)
4. Negotiation (Verhandlung der Konditionen)
5. Purchase and Delivery (Kauf und Lieferung)
6. Service and Evaluation (Service und Auswertung)

Insbesondere in den Phasen 2.-4. können durch Web-basierte Tools Vereinfachungen und somit eine Reduktion des Suchaufwandes und der Suchkosten des Konsumenten erzielt werden. Viele dieser Tools fokussieren sich jedoch lediglich auf eine oder zwei dieser Phasen [GMM98]. Um jedoch eine möglichst hohe Vereinfachung der Markttransaktion zu erreichen, strebt dieser Artikel die Unterstützung sowohl der *Product Brokering*, *Merchant Brokering* und *Negotiation-Phase* an. Um dieses Ziel zu erreichen ist es zunächst von Nöten, eine geeignete Transaktionsform beziehungsweise den entsprechenden Preisbildungsmechanismus zu bestimmen.

In der Preistheorie wird zwischen statischer und dynamischer Preisbildung unterschieden, wobei letztere noch in einseitig-dynamische und zweiseitig-dynamische Preisbildung untergliedert werden kann [Wu01]. Zu den Mechanismen der statischen Preisbildung zählen fixe Anbieterpreise (Festpreisangebote), Preislimits durch die Nachfragenden beziehungsweise ein beidseitiger Tauschhandel. Einseitig dynamische Preise werden in erster Linie mittels Auktionen oder reverse Auktionen (Ausschreibungen) realisiert. Daneben sind Börsen eine Plattform für zweiseitig dynamische Preise. Diese Preisbildungsarten können dabei sowohl auf Business-to-Business (B2B), Business-to-Consumer (B2C) und Consumer-to-Consumer (C2C)-Märkten angewandt werden. Die empirische Betrachtung zeigt jedoch, dass je nach Marktplatz bestimmte Mechanismen genutzt werden, wogegen andere in den Hintergrund treten (siehe Abbildung 1). Der B2B-Markt wird in erster Linie durch individuelle, komplexe Beziehungen zwischen Vertragspartnern bestimmt, in denen die zu handelnde Leistung/Ware und deren Preise beidseitig ausgehandelt werden. Daneben werden jedoch insbesondere hier auch Ausschreibungen für Beschaffungstransaktionen eingesetzt, sowohl zwischen öffentlichen Auftraggebern und Unternehmen, als auch zwischen Unternehmen selbst [MS02],[CJ08]. Ein großer Teil wird (wenn die Komplexität dies zulässt) online abgewickelt [Ja02]. Im B2C-Umfeld dominieren in erster Linie die statisch gesetzten Festpreisangebote. Zwar gibt es die Möglichkeit von Rabatten und Skonti, jedoch ist der Preis grundsätzlich durch den Verkäufer determiniert. Im C2C-Kontext finden häufig

Gesuche beziehungsweise Kleinanzeigen von Konsumenten untereinander Anwendung, dabei kann auch hier auf Festpreisangebote zurückgegriffen werden. Marktplatzübergreifend spielen Auktionen eine wichtige Rolle. Diese sind in allen drei Fällen anwendbar.



Abbildung 1: Märkte und deren Transaktionsformen.

Verglichen mit dem CBB-Modell bieten reverse Auktionen den einzigen Koordinationsmechanismus, der die Phasen 2.-4. aus Sicht des Konsumenten/Nachfrager repräsentiert. Jedoch wurden diese wie dargestellt bislang überwiegend im B2B-Umfeld genutzt. Dagegen bieten insbesondere die neuen Möglichkeiten des Ubiquitous Computing eine Vielzahl an Eigenschaften, die die Übertragung dieses Koordinationsmechanismus in den B2C- und C2C-Kontext zulassen. Im folgenden Abschnitt werden daher (via Internet umgesetzte) reverse Auktionen näher betrachtet.

### 3.2 Online Reverse Auctions

Neben den fundamentalen Auktionstypen existieren reverse Auktionen, welche für Beschaffungsprozesse genutzt werden. In einer reversen Auktion sind die Rollen des Käufers und Verkäufers einer traditionellen (Englischen) Auktion umgekehrt. Anbieter eines benötigten Gutes/einer benötigten Dienstleistung bieten sukzessive eine Reduktion des Preises, bis das zuletzt verbleibende niedrigste Gebot gewinnt [MS02], [K104], [SC03]. Grundannahmen sind ein genügendes Angebot und eine ausreichende Gewinnmarge der Verkäufer. Da reverse Auktionen offen und transparent ablaufen, indem jeder Teilnehmer eine Übersicht über die abgegebenen Gebote hat, werden die Verkäufer/Anbieter selbst die Preise solange reduzieren, bis ein ausbalanciertes Niveau erreicht ist [SC03].

Bisher werden Online durchgeführte reverse Auktionen („Online Reverse Auctions“, ORAs) oft für großskalige Beschaffungsprozesse im B2B-Umfeld genutzt. Die empirischen Befunde in diesem Kontext variieren zwischen drastischen Einsparungen der Beschaffungskosten durch die Umstellung auf Ausschreibungen auf der einen Seite und nur kleinen Verbesserungen auf der anderen, insbesondere auf längere Sicht [MS02]. B2B-Beschaffungsprozesse sind ein integraler Bestandteil der komplexen Beziehungen zwischen Unternehmen. Ein sehr wichtiger Aspekt dieser Beziehungen ist das Vertrauen

zwischen den involvierten Partnern. Das Etablieren von Vertrauen ist ein sensibler kontinuierlicher Prozess, der zu Wettbewerbsvorteilen und Kostenreduktion führen kann [BH94], indem der Austausch von Informationen und Wissen unterstützt und Konfliktmanagement und die Koordination ökonomischer Aktivitäten vereinfacht werden [CJ08]. ORAs könnten diese reziproke Verbindung durch das Eröffnen von Wettbewerbssituationen schädigen und die Gefahr von Konflikten und opportunistischem Verhalten erhöhen [MJM08], [CR09]. Diesbezüglich entwickelten Mabert and Skeels die in Abbildung 2 dargestellte Klassifikation von günstigen Attributen, die zu einer effizienten Anwendung von ORAs führen sollen. Jap entwickelt darüber hinaus ein konzeptionelles Rahmenmodell für ORAs unter der Berücksichtigung von Einflüssen auf Käufer-Verkäufer-Beziehungen [Ja07]. Da dieser Artikel seinen Schwerpunkt jedoch auf das B2C-Umfeld legt, können Bedenken bezüglich langfristiger Vertrauensbeziehungen zwischen Geschäftspartnern verworfen werden. Dennoch behalten die meisten der in Abbildung 2 dargestellten Attribute ihre Gültigkeit, mit dem Unterschied dass explizit keine strategischen Güter, sondern Alltagsbedürfnisse in den Fokus gerückt werden. Ein Vergleich dieser Attribute mit dem angestrebten Ziel verdeutlicht, dass ORAs ein adäquates Werkzeug für alltägliche Bedürfnisse (wie zum Beispiel “ich brauche jetzt eine Pizza” oder “Ich brauche ein Taxi zum Flughafen”) sind. Sie ermöglichen die Koordination der Product Brokering, Merchant Brokering und Negotiation-Phase und sind somit ein geeignetes Werkzeug für das Forschungsziel dieses Artikels. Die präzise Benennung des für diese Zwecke zu verwendenden Auktionstypens lautet folglich Multi-Attribut-Mikro-ORAs.

<b>Dimension</b>	<b>Favorable attributes for reverse auction</b>
<b>Item (commodity)</b>	Strategic item/service Easy to group into attractive bid lots Easy to define item/service specifications (RFQ)
<b>Supply base</b>	Many qualified bidders available (min. of three) Excess capacity in supply base Supply base electronically enabled
<b>Buying firm</b>	Incur low vendor qualification costs Incur low vendor switching costs No contract barriers Few or no plant supplier preferences Purchasing group electronically enabled

Abbildung 2: Kriterien für den Einsatz von reversen Auktionen (B2B) [MS02].

## 4 Konzeption eines pervasiven Mikro-ORA-Systems

### 4.1 Anforderungen und grundlegende Architektur

Ausgehend von den vorherigen Abschnitten ist das Erkenntnisziel für diesen Aufsatz das Entwickeln eines plattformübergreifenden und pervasiven Systems zur Vereinfachung der Echtzeit-B2C-Koordination in alltäglichen Situationen. Basierend auf den Ergebnissen bezüglich der speziellen Herausforderungen des Systemdesigns sozio-technischer Anwendungen und den grundlegenden Erkenntnissen aus Abschnitt 3, werden für das zu entwickelnde Software-Artefakt die folgenden Anforderungen festgesetzt:

- Einfaches, intuitives Design der (Nutzer-) Schnittstellen,
- automatische Anpassung an die momentane Umgebung,
- Mobilität/Plattformunabhängigkeit,
- Unterstützung von reversen Auktionen mit kurzer bis sehr kurzer Laufzeit,
- Unterstützung der Product Brokering, Merchant Brokering und Negotiation-Phase des CBB-Modells,
- Transparenz bezüglich der ausgetauschten Daten

Um das Erkenntnisziel zu erreichen und diese Anforderungen zu erfüllen, wird ein Ansatz verfolgt, der die neuen Funktionen des Geolocated Browsing, die steigende Verbreitung von Smartphones, Status-Feeds wie aus Social Networks/Microblogs und die Vorgehensweise von ORAs kombiniert und so zu einem mobilen, pervasiven Mash-Up-System führt. Das adäquate Grundgerüst für ein solches System bietet die Event-Driven-Architecture (EDA) [DEF08], [TYP09]. Diese nutzt eine nachrichtengesteuerte Middleware (Message-oriented Middleware, MOM) als zentrales Element, welches auch als Event-Processor bezeichnet werden kann [DEF08]. Die involvierten Komponenten senden ein Ereignis eines speziellen Typs an die Middleware, wo sich andere Komponenten im Vorfeld bereits als Empfänger für spezielle Ereignistypen registriert haben. Sobald ein Ereignis ankommt, sendet die Middleware eine Nachricht an die für dieses Ereignis registrierten Komponenten, welche dann das Ereignis eigenverantwortlich übernehmen [VF11]. Übertragen auf das Nutzungsszenario dieses Artikels gestaltet sich die Interaktion wie folgt: Sobald ein Nutzer eine Anfrage nach einer speziellen Dienstleistung postet, wird diese an die Middleware geleitet. Diese greift auf einen Index der registrierten Anbieter für die geforderte Leistung/das geforderte Gut zurück und informiert diese via Narichtenprotokoll. Die Anbieter können dann ihre Gebote abgeben. Neben der Middleware, welche als Broker zwischen dem nachfragenden Konsumenten und den teilnehmenden Versorgern/Anbietern fungiert, benötigt das System Nutzerschnittstellen für Konsumenten. Diese werden mittels der Integration von APIs (Application Programming Interfaces) bestehender Social Software (Facebook, Twitter, Google+ etc.) und der Implementierung einer eigenen Oberfläche realisiert. Zum dritten muss die Kommunikation mit den Anbietern durch eine Client-seitige Anwendung sichergestellt werden, welche

die Nachrichten der Middleware empfängt und die Gebote an die entsprechenden Nachfragen postet.

## 4.2 System Design

Die im vorherigen Abschnitt beschriebenen Anforderungen benötigen ein mobiles und plattformunabhängiges Systemdesign. Dazu werden alle Komponenten (siehe Abbildung 3) als Web-basierte Anwendungen, basierend auf einer zweistufigen Client-Server-Architektur realisiert, um flexible und leichtgewichtige Client-Anwendungen zu ermöglichen, welche lediglich die Darstellungsebene präsentieren müssen, wogegen Anwendungslogik und Datenhaltung auf dem Server verbleiben [Fr99].

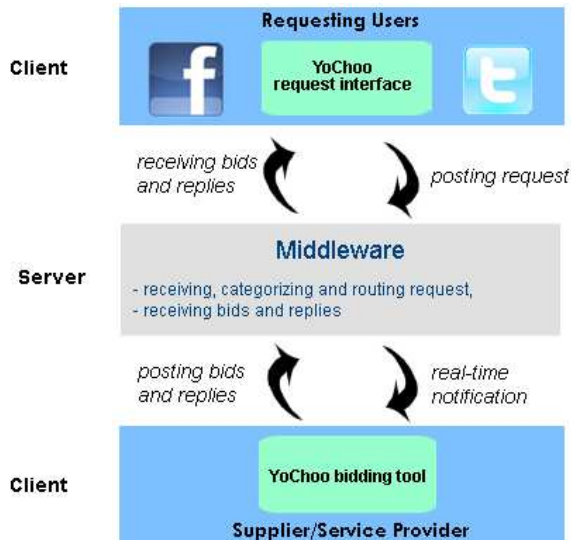


Abbildung 3: System Design.

Die MOM als zentraler Bestandteil der Systemdesigns wird in der Rolle des Servers agieren. Sie beinhaltet die beiden Ebenen der Datenhaltung (in relationalen Datenbanken), sowie die Datenverarbeitung und zusätzlich die Kommunikation mit den Front-End-Nutzeroberflächen der Clients. Von einer der Nutzerschnittstellen erhält sie die Anfrage eines Konsumenten, welche den Text der Anfrage selbst, die Nutzerdaten und die momentane geographische Position des Nutzers erhält. Im nächsten Schritt muss die Anfrage mittels enthaltener Schlüsselwörter klassifiziert werden, bevor aus der Datenbank der registrierten Versorger die passenden für die gesuchte Kategorie selektiert werden können. Die so bestimmten Empfänger werden dann via Real-Time-Nachrichten-Protokoll informiert. Diese Funktionalität wird mit Hilfe des "PubSubHubbub Protocol" [Da09] umgesetzt, welches das XML-basierte RSS-Protokoll zu einem nahezu verzugslosen System beschleunigt.



Die Anforderungen an die Systeme der Clients der Konsumenten werden vorsätzlich niedrig gehalten, damit sowohl die Mobilität, als auch die Plattformunabhängigkeit gewährleistet werden können. Die Client-Geräte müssen lediglich die Web-Front-End-Präsentation gewährleisten (Präsentationsebene), welche nur drei Funktionalitäten beinhaltet: Nutzer-Authentifizierung, ein Eingabeformular für die Anfrage und die Identifikation des momentanen geographischen Ortes des Nutzers durch den Smartphone-GPS-Sensor oder das HTML5-Geolocation Feature des Browsers [Po10]. Diese drei Informationen werden an die MOM gesendet, welche dann mit der Evaluierung und Kategorisierung beginnt (wie im vorherigen Absatz beschrieben).

Da sich die Nutzer-Interaktion an das Konzept von Status-Posts in Social Networks anlehnt, soll die Nutzeroberfläche auch in diese integriert werden. Dies wird durch das Nutzen der bereitgestellten APIs erreicht. Einige (wie beispielsweise die Facebook-API) erlauben das Sammeln genau der benötigten Daten: Nutzer-Authentifizierung, momentane Position und den Text des Status-Posts, welcher die Anfrage enthält. Die Client-Software für die Anbieter wird ebenfalls als browser-basierte Nutzeroberfläche entwickelt. Sie greift auf den Anfrage-Stream zurück, den die MOM für jede Kategorie generiert. Dazu muss die Nutzeroberfläche die momentan offenen Anfragen darstellen können, sowie ein Textformular für das Eingeben von Geboten bieten und den Updates der laufenden ORAs in Echtzeit folgen können.

## 5 Implementierung und beispielhafte Anwendung

Dieser Abschnitt präsentiert und beschreibt durch das Verfolgen eines beispielhaften Nutzungsszenarios jede Komponente des entwickelten Software-Artefaktes und demonstriert die Funktionalität der zuvor beschriebenen Implementierung. In den folgenden Absätzen wird das prototypenhaft implementiert Software-Artefakt als “YoChoo” (eine künstliche Kurzform von “You Choose”) bezeichnet – ein Platzhalter für die präzise Bezeichnung „Multi-Attribut-, Online-Mikro-revers-Auktionssystem für pervasive C2B-Koordination“.

### 5.1 Fallbeispiel

Um dem Szenario zu folgen betrachten wir eine Person, welche einen Sightseeing-Trip für ein kommendes Wochenende plant. Sie nutzt ihren Facebook-Account, welcher mit der YoChoo-Facebook-App verbunden ist, um ihre Freunde zu informieren und automatische eine Anfrage zu stellen. Also schreibt sie:

*„@YoChoo On the last weekend of April, I'm going to +London again :D I just need a #hotel in +SoHo!“*

Mit Bezug auf das CBB-Modell ist das Bedürfnis also identifiziert (*Need Identification*) und mit diesem Post werden sowohl die Phasen *Product-* und *Merchant Brokering* abgedeckt.

Ein kürzlich eingeführtes Facebook-Feature ist das Anhängen der momentanen Position des postenden Nutzers an jeden Status-Post [Bi10]. Das „@YoChoo“ verbindet den Post automatisch mit der YoChoo-Facebook-App und macht diesen somit weiterverarbeitbar. Der Post liefert jede Art der geforderten Informationen: Der Nutzer kann über die durch die API übertragenen Nutzerinformationen bestimmt werden und das Keyword „#hotel“ bestimmt die Kategorie der zu informierenden Anbieter. Die momentane Position des Nutzers wird ebenfalls via API geliefert, wobei in diesem Fall der entscheidende Ort durch den +London-Tag vorgegeben ist. Das Zeichen „#“ führt einen Kategorisierungstag ein, wogegen ein „+“ den zu betrachtenden Ort kennzeichnet. Wird kein „+“-Tag spezifiziert, wird standardmäßig die momentane Nutzerposition übernommen. Die MOM evaluiert dieses Bündel an Informationen und sendet Echtzeit Benachrichtigungen an jeden Client, welcher sich zuvor für das entsprechende Tag (#hotel) registriert hat. Nun beginnt die *Negotiation*-Phase. Ein Hotel namens „SoHo Inn“ erhält Sekunden später die Anfrage unseres Fallbeispiels auf seinem YoChoo-Bieter-Interface, welches die aktiven Anfragen abhängig von der geographischen Entfernung (Nutzerbestimmt) filtert. Dazu sendet das Bieter-Interface die Keywords „London, SoHo“ an die Google-Maps-API, von der sie die Koordinaten dieses Stadtteils zurückerhält, mit denen anschließend die Entfernung zum Hotel „SoHo Inn“ berechnet wird. Dieses Hotel hat beispielsweise noch ein Einzelzimmer für das entsprechende Wochenende frei, weshalb der Concierge auf die Anfrage mit einem Gebot antwortet:

*„Stay at ours! You’ll just pay 60 GBP including a supreme continental breakfast!”*

Dieser Post wird nun an die MOM übertragen. Die Nutzerschnittstellen des Konsumenten (sowohl die proprietäre, als auch die in Social Software integrierten) werden sofort geupdatet. In dem betrachteten Fall wird das Gebot als Kommentar an die Anfrage des Nutzers angehängt. Dieser kann nun das Gebot akzeptieren oder auf weitere Gebote warten. In diesem Szenario antwortet das „City Hostel“ mit einem weiteren Gebot:

*“We offer a room for only 40 GBP! No breakfast though, but we have a couple of excellent bars with English breakfast around here!”*

Unser Tourist akzeptiert mit dem Post *„We have a deal, City Hostel!”* das zweite Gebot. Die weitere Abwicklung ist den Beteiligten selbst überlassen. Sie könnten nun die Telefonnummern oder E-Mail-Adressen austauschen um die verbleibenden Details zu klären. Dieses Beispiel hat gezeigt, wie die Integration von YoChoo in Social Networks aussieht. Das nun folgende Szenario beschreibt das proprietäre YoChoo Request-Interface.

Der Tourist ist inzwischen in London und trifft sich mit alten Freunden in einer Bar. Während die Zeit verstreicht, wird ihm klar dass ein Taxi nötig ist, um zum Hotel zurück zu kehren. Über das Smartphone öffnet er die YoChoo-Website und schreibt (nach dem Log-In):

*„I need a #taxi to CityHostel in 5 mins!”*

Die momentane Position wird durch den geräteinternen GPS-Sensor determiniert. Die MOM sendet sofort eine Nachricht an alle Clients der Kategorie #taxi. Diese können nun

(wie oben beschrieben) ihr Gebot abgeben, welches dann im YoChoo Request-Interface dargestellt wird (siehe Abbildung 4).

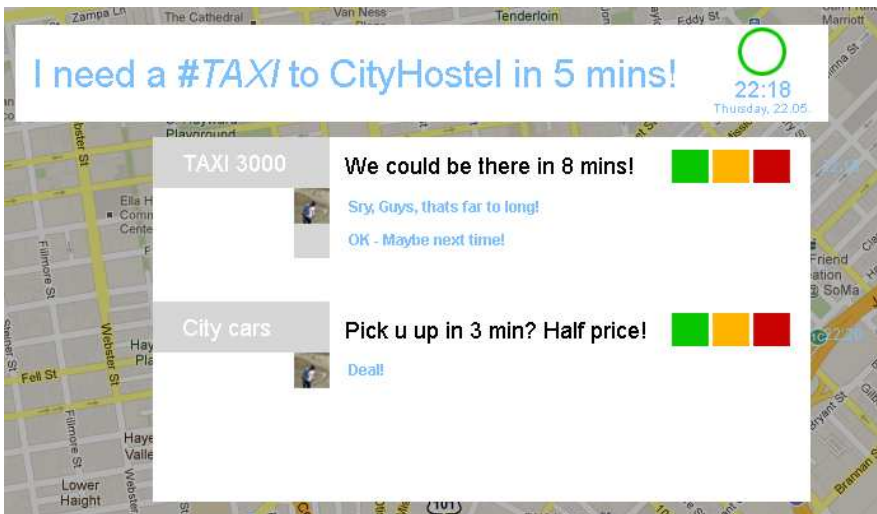


Abbildung 4: Darstellung der Anfragen und Gebote im Web-Interface.

Die Buttons in rot, gelb und grün haben die folgenden Funktionen: Der rote Button lehnt das Gebot ab, der gelbe signalisiert weiteren Verhandlungsbedarf und der grüne Button akzeptiert ein Gebot. Sobald der Nutzer einen davon betätigt, wird das Resultat auf der Bieter-Oberfläche dargestellt. Diese Funktionalität kann jedoch zum jetzigen Zeitpunkt nicht mit Social-Networks-APIs verwendet werden, da deren Funktionen limitiert sind.

Diese beiden Szenarios sind lediglich kleine Beispiele von möglichen Anwendungsfällen von YoChoo. Das Konzept der Mikro-ORAs im Alltagskontext scheint nahezu universell. Die technische Umsetzung der drei grundlegenden Komponenten wird nun in den folgenden Absätzen erklärt, bevor zusammenfassende und diskutierende Gedanken über dieses System den Artikel abschließen.

## 5.2 Das YoChoo Request Interface

Wie zuvor beschrieben, handelt es sich bei der Nutzerschnittstelle für Anfragen um eine Web-basierte Anwendung, welche mit jedem Browser – unabhängig von der jeweiligen Plattform – genutzt werden kann. Dadurch wird volle Mobilität und Interoperabilität gewährleistet. Der Quellcode ist eine Mischung aus HTML5, AJAX und PHP. Die Nutzung von HTML5 ist notwendig, um über die kürzlich eingeführte Geolocation API (Popescu, 2010) die Koordinaten der momentanen Position des Konsumenten zu empfangen. Die Nutzer autorisieren sich mittels Nutzernamen und Passwort, oder via Facebook- und OAuth-Accounts. Das zentrale Element des Designs des YoChoo-Request-Interface ist das Textfeld für die Eingabe der Anfrage. Die erkannte Position wird mittels einer großen Karte im Hintergrund des Textfeldes dargestellt, auf welcher der Nutzer bei Bedarf per Drag-and-Drop seine Position korrigieren kann. Der Nutzer tippt also seinen

Anfragetext in das Textfeld, kennzeichnet die Kategorie-Schlüsselwörter mittels „#“ und zu betrachtende Orte (abweichend von der momentanen Position) mit einem „+“. Die Anfrage wird mit dem Betätigen der Enter-Taste übernommen. Mittels PHP wird dieser Text nun an die MOM weitergegeben. An dieser Stelle sei jedoch darauf hingewiesen, dass PHP kein Multithreading unterstützt und bei einer flächendeckenden Implementierung auf eine entsprechende alternative serverseitige Sprache ausgewichen werden sollte. Durch das Zurückgreifen auf AJAX wird das Text-Feld nicht mehr editierbar und insensitiv, ohne die Seite neu zu laden, ebenso werden sofort die empfangenen Gebote inklusive Nutzernamen der Bieter unterhalb des Textfeldes angezeigt (siehe Abbildung 4). Ein Klick auf das jeweilige Logo/Avatar zeigt dann weitere Informationen zu dem Bieter an, wie beispielsweise die Reputation und Bewertungen vergangener Transaktionen.

### 5.3 Die Message-Oriented Middleware

Die Rolle des Servers in der Systemarchitektur wird durch die MOM ausgefüllt. Diese erhält die geposteten Anfragen aus dem YoChoo-Request-Interface oder aus den APIs der Social Networks. Die Hauptaufgaben der MOM sind Datenhaltung (mittels SQL-Datenbanken) und die Datenverarbeitung. Sie verwaltet die Informationen über jeden registrierten Nutzer (Konsument sowie Anbieter), erzeugt Datenbankeinträge für jede neue Anfrage und die dazugehörigen Gebote und Kommentare. Dies wird durch serverseitige PHP-Skripte in Verbindung mit SQL-Elementen realisiert. Diese greifen auf die Google Maps-API zurück, um Namen von Orten in geographische Koordinaten umzuwandeln. Darüber hinaus evaluieren diese Skripte die gekennzeichneten Schlüsselwörter („#“- und „+“-Tags) und wählen die kongruente Anbieterkategorie. Sobald diese determiniert wurde, generieren die Skripte einen neuen XML-Eintrag in dem RSS-Feed für diese Kategorie, welcher die aus der Anfrage entnommenen Informationen enthält. Zusätzlich verfügt die MOM über eine XML-RPC-Schnittstelle (XML Remote Procedure Call), um Gebote und Kommentare des Bidding-Tools der Bieter zu empfangen.

### 5.4 Das „YoChoo Bidding Tool“

Das Bidding-Tool (Abbildung 5) wurde mithilfe der gleichen Komponenten implementiert wie das Request-Interface: HTML5, AJAX und PHP. Es nutzt den kategoriespezifischen RSS-Feed um alle aktiven Anfragen für die jeweilige Kategorie des Anbieters anzuzeigen. Zudem nutzt auch das Bidding-Tool die Google-Maps-API, um die Distanz der Anfragen zum Standort des Anbieters zu berechnen. Es erlaubt dem Nutzer zudem einen Radius-Filter auf die darzustellenden Anfragen anzuwenden. Die Anbieter können dann eine der aufgelisteten Anfragen auswählen und via Textfeld ihr Gebot beziehungsweise ihren Kommentar posten. Die eingegebenen Informationen werden mittels XML-RPC an die MOM gesendet. Die Kombination eines RSS-basierten Protokolls für eingehende Nachrichten mit XML-RPC für ausgehende Nachrichten ermöglicht eine komplett XML-basierte Interaktion mit der MOM und somit komplette Plattformunabhängigkeit und Erweiterbarkeit und Integrierbarkeit für angeschlossene Systeme.

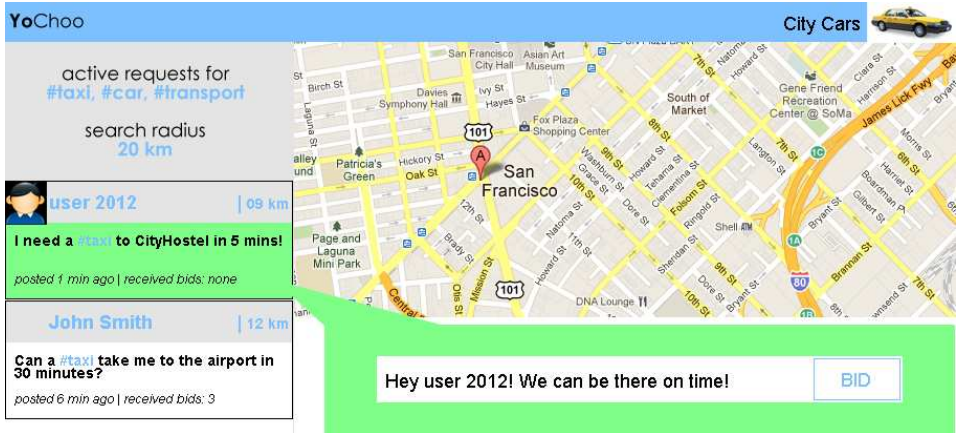


Abbildung 5: Das Browser-basierte „Bidding Tool“.

## 6. Schlussbetrachtung

Die Evaluation des entwickelten und implementierten Software-Artefaktes gegen die in Abschnitt 4.1 aufgestellten Anforderungen zeigt, dass das Erkenntnisziel erreicht wurde: Das Schnittstellendesign wurde einfach und intuitiv gestaltet, da der Nutzer lediglich seine Anfrage in natürlicher Sprache eintippen braucht. Die einzige formale Restriktion ist das Setzen von Tags zur Kennzeichnung von Schlüsselwörtern. Das geforderte automatische Anpassen an die Umgebung wurde erreicht, indem das System automatisch die momentane Position der Nutzer erkennt und weiterverarbeitet. Mobilität und Plattformunabhängigkeit wurde durch die komplett Web-basierte Architektur und XML-basierte Informationsübertragung sichergestellt. Es wird lediglich ein Browser für die Nutzung von YoChoo benötigt – ganz egal auf welchem System dieser betrieben wird. Des Weiteren wurden die Barrieren für die Nutzung des Systems dadurch niedrig gehalten, dass Nutzer sich auch mit bestehenden Facebook- oder OAuth-Accounts anmelden können. Da YoChoo lediglich als Plattform zum Zusammenführen von Nachfrage und Angebot konzipiert wurde, benötigt es sehr wenige Informationen vom Nutzer und bietet somit ein hohes Maß an Transparenz.

Dennoch ist die momentane Version von YoChoo noch in einer frühen Entwicklungsphase. Die Anwendung in der Praxis hat bereits verbesserungswürdige Punkte aufgezeigt. Dem Nutzer zu erlauben, die Anfrage in natürlicher Sprache zu stellen, reduziert zwar die Zugangsbarrieren und erhöht die Intuitivität, jedoch wird dadurch eine mögliche Automatisierung auf Seiten der Anbieter stark erschwert. In der aktuellen Version geht das System davon aus, dass die Interaktion auf Konsumenten- als auch Anbieterseite durch Menschen durchgeführt wird. Gerade im Kontext größerer Unternehmen (z. B. eine Anfrage an Fluggesellschaften für Flüge) scheint es unwahrscheinlich, dass jede Anfrage von einem Mitarbeiter manuell bearbeitet wird. Die künftige Entwicklungsarbeit sollte sich daher auf die Implementierung automatisierter Ressourcen fokussieren, jedoch unter der Prämisse, die Restriktionen an die Formatierung der Anfrage möglichst

niedrig zu halten. Wie in Abschnitt 2.1 beschrieben, stehen pervasive Computersysteme Herausforderungen gegenüber, die über das bloße Design und die Implementierung hinausgehen: Sie benötigen die Akzeptanz der potentiellen Nutzergruppen, da sie zu Veränderungen in alltäglichen Situationen führen und zudem konform zu gesetzlichen Regelungen sein müssen (beispielsweise nicht-verhandelbare Taxipreise in vielen Ländern). Diese Hürden betreffen auch YoChoo. Es setzt voraus, dass die Nutzer bereit sind ihre aktuellen Bedürfnisse in Verbindung mit der geographischen Position zu teilen. Jedoch geht es weit darüber hinaus: YoChoo fordert einen Paradigmenwechsel auf beiden Seiten – Konsument und Anbieter. Bis jetzt haben Anbieter ihre Dienstleistungen angeboten und die Konsumenten mussten diese finden und auf sie zukommen (B2C). YoChoo wendet die Richtung der alltäglichen Marktkoordinierung von einer B2C- zu einer C2B-Philosophie, mit dem Ziel Preise und Leistungen effektiver zu bestimmen. Da Social Software generell von einer bestimmten Mindestanzahl an Nutzern abhängig ist um zu funktionieren [BGT08], vertraut der hier vorgestellte Ansatz auf die Partizipation der Anbieter. Diese könnten jedoch Vorbehalte gegenüber einem System für offene und dynamische Preisfindung haben. [HSF11] schlagen in diesem Zusammenhang ein Framework für die Entwicklung und Realisierung von pervasiven Systemen vor, welches für die weiteren Schritte in Betracht gezogen werden kann.

Ein neu entwickeltes Softwaresystem kann nur durch die Nutzung in der Praxis effektiv evaluiert und verbessert werden. Praktische Erfahrungen erzeugen grundlegende Rückschlüsse bezüglich der Nutzbarkeit und des Potentials eines Systems. Aus der Forschungsperspektive wird der nächste Schritt die Nutzung von „Focus Groups“ für die Evaluation und Verbesserung des Artefaktes sein [THB10] – jedoch immer unter Berücksichtigung der speziellen Herausforderungen, die die Umsetzung eines pervasiven Computersystems mit sich bringt.

## Literaturverzeichnis

- [BGT08] Back, A., Gronau, N., Tochtermann, K. 2008. Web 2.0 in der Unternehmenspraxis. Oldenbourg Wissenschaftsverlag, München.
- [BH94] Barney, J. B., Hansen, M. H. 1994. Trustworthiness as a Source of Competitive Advantage. *Strategic Management Journal*, (15), 175-190.
- [Bi10] Bilton, N. 2010. Facebook Will Allow Users to Share Location. *The New York Times: Bits*. <http://www.bits.blogs.nytimes.com/2010/03/09/facebook-will-allow-users-to-share-location/>.
- [CJ08] Charki, M. H., Josserand, E. 2008. Online Reverse Auctions and the Dynamics of Trust. *Journal of Management Information Systems*, 24 (4), 175-197.
- [CR09] Caniels, M.C.J., Van Raaij, E.M. 2009. Do All Suppliers Dislike Electronic Reverse Auctions?. *Journal of Purchasing and Supply Management*, 15(1), 12-23.
- [Da09] Dash, A. 2009. The Pushbutton Web: Realtime Becomes Real. <http://dashes.com/ani/2009/07/the-pushbutton-web-realtime-becomes-real.html>.
- [DEF08] Dunkel, J., Eberhart, A., Fischer, S., Kleiner, C., Koschel, A. 2008. Systemarchitekturen für verteilte Anwendungen. Hanser, München.
- [Fr99] Fraternali, P. 1999. Tools and Approaches for Developing Data-Intensive Web Applications: A Survey. *ACM Computing Surveys*, 31 (3), 227-263.
- [GMM98] Guttman, R. H., Moukas, A. G., Maes, P. 1998. Agent-mediated Electronic Commerce: A Survey. *The Knowledge Engineering Review* 13 (2), 147-159.

- [GR11] Gartner Research, 2011. Market Share: Mobile Communication Devices by Region and Country, 3Q11. <http://www.gartner.com/resId=1847315>.
- [HSF11] Hoffmann, A., Söllner, M., Fehr, A., Hoffmann, H., Leimeister, J. M. 2011. Towards an Approach for Developing socio-technical Ubiquitous Computing Applications. Informatik 2011 - Informatik schafft Communities. Beiträge der 41. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Berlin, Germany.
- [Ja02] Jap, S. D. 2002. Online Reverse Auctions: Issues, Themes, and Prospects for the Future. *Journal of the Academy of Marketing Science*. 30 (4), 506-525.
- [Ja07] Jap, S. D. 2007. The Impact of Online Reverse Auction Design on Buyer-Supplier Relationships. *Journal of Marketing*, 71(1), 146-150.
- [Kl04] Klemperer, P. 2004. Auctions: theory and practice. Princeton University Press, Princeton.
- [La03] Langheinrich, M., 2003. When Trust Does Not Compute - The Role of Trust in Ubiquitous Computing. Proceedings of Privacy Workshops of Ubicomp'03.
- [MJM08] Mithas, S., Jones, J. L., Mitchell, W. 2008. Buyer Intention to Use Internet-Enabled Reverse Auctions: The Role of Asset Specificity, Product Specialization, and Non-Contractibility. *MIS Quarterly*, 32 (4), 705-724.
- [MS02] Mabert, V. A., Skeels, J. A. 2002. Internet Reverse Auctions: Valuable Tool in Experienced Hands. *Business Horizons*, 45 (4), 70-76.
- [Or05] O'Reilly, T. 2005. What is Web 2.0?: Design patterns and business models for the next generation of software. <http://www.oreillynet.com/pub/a/oreilly/2005/09/30/what-is-web-20.html>.
- [Po10] Popescu, A. (2010). Geolocation API specification, <http://dev.w3.org/geo/api/spec-source.html>.
- [PSV03] Pinker, E. J., Seidmann, A., Vakrat, Y. (2003). Managing Online Auctions: Current Business and Research Issues. *Management Science*, 49 (11), 1457-1484.
- [SC03] Smeltzer, L. R., Carr, A. S. 2003. Electronic Reverse Auctions: Promises, Risks and Conditions for Success. *Industrial Marketing Management*, 32 (6), 481-488.
- [Sh10] Shin, D., 2010. Ubiquitous Computing Acceptance Model: End User Concern About Security, Privacy and Risk. *International Journal of Mobile Communications*, 8 (2), 169-186.
- [THB10] Tremblay, C. M., Hevner, A. R. Berndt, D. J. 2010. Focus Groups for Artifact Refinement and Evaluation in Design Research. *Communications of the Association for Information Systems*, 26 (27).
- [TYP09] Taylor, H., Yochem, A., Phillips, L., Martinez, F. 2009. Event-Driven Architecture – How SOA Enables the Real-Time Enterprise. Pearson Education, Boston.
- [VF11] Vogt, S., Fink, A. 2011. Using Status Feeds for Peer Production by Coordinating Non-predictable Business Processes. *BPM 2011 Workshops, Part I, LNBIP 99*, 253–265.
- [WB96] Weiser, M., Brown, J. S. (1996). Designing calm technology. *Power Grid Journal*, 1 (1), 75-85.
- [We91] Weiser, M. 1991. The Computer for the 21st Century. *Scientific American* 265 (3). 94-104.
- [Wu01] Wurman, P. R. 2001. Dynamic Pricing in the Virtual Marketplace. *IEEE Internet Computing* 5 (2), 36-42.