



Workshop über
Selbstorganisierende, adaptive, kontextsensitive
verteilte Systeme
(SAKS 2010)

Nutzerintegration in die Anforderungserhebung für Ubiquitous
Computing Systeme

Axel Hoffmann, Holger Hoffmann und Jan Marco Leimeister

9 Pages

Nutzerintegration in die Anforderungserhebung für Ubiquitous Computing Systeme

Axel Hoffmann, Holger Hoffmann und Jan Marco Leimeister

Fachgebiet Wirtschaftsinformatik, Fachbereich
Wirtschaftswissenschaften, Universität Kassel

Abstract: Der Nutzer wird als die wichtigste Quelle für Anforderungen an technische Systeme gesehen. Die technikgetriebene Entwicklung ubiquitärer Systeme nimmt darauf zurzeit jedoch wenig Rücksicht. Wir beschreiben, wie der Nutzer nach bekannten Requirements Engineering Ansätzen besser in die Anforderungserhebung einbezogen werden kann und welche Probleme dabei auftreten. Dabei stehen die besonderen Eigenschaften ubiquitärer Systeme – wie die Tatsache, dass der Nutzer das System nicht mehr aktiv bedient, sondern durch Verwendung von Smart Devices steuert – Im Vordergrund. Ein weiterer Fokus bei der von uns beschriebenen Nutzerintegration liegt zudem auf der Tatsache, dass die Nutzer durch die Neuartigkeit ubiquitärer Systeme kaum eine Vorstellung von den sich bietenden Möglichkeiten für ubiquitäre Anwendungen haben und entsprechend anfänglich keine spezifischen Anforderungen nennen können.

Keywords: Anforderungserhebung, Requirements Engineering, Ubiquitous Computing, nutzergetriebene Entwicklung, User-Centered Design

1 Einleitung

Ubiquitous Computing schafft in der Vision von Weiser dem Nutzer ungeahnte Möglichkeiten [1]. IT-Unterstützung im Alltagsleben ohne die Fokussierung auf einen Computer bringt bei der Entwicklung neue Freiheitsgrade mit sich, die durch immer neue Entwicklungen im Hardwarebereich auch realisierbar werden. Die Entwicklung von Ubiquitous Computing Systemen findet jedoch weitestgehend unter Ausschluss späterer Nutzer statt, deren Anforderungen somit auch keine Beachtung finden.

Erfolgreiche UC Anwendungen müssen aber nicht nur technische Aspekte korrekt abbilden, sondern vor allem die Bedürfnisse der Nutzer erfüllen [2]. Daher ist es unerlässlich, die Nutzerbeteiligung an der Entwicklung zu erhöhen. Die Neuartigkeit ubiquitärer Systeme und die damit verbundene fehlende Vorstellungskraft der Nutzer, welche Möglichkeiten derartige Systeme haben, schränken den Nutzereinsatz im klassischen Requirements Engineering aber ein. Daher stellen wir im folgenden Beitrag eine systematische Lösung vor, welche die Innovationskraft größerer Gruppen, die Neuartigkeit der Technik und die Anforderungen der späteren Nutzer zu einem integrativen Ansatz für das Requirements Engineering für ubiquitäre Systeme verbindet.



2 Technikgetriebene Entwicklung

Die Vision des Ubiquitous Computing hat in der Vergangenheit die Erwartungen von Hard- und Softwareherstellern, Dienstleistern und Endkunden geschürt, in naher Zukunft bereits „intelligente“ Haushaltsgegenstände zu verkaufen und darauf basierende Services anbieten und nutzen zu können. Mit dem Vernetzen der Alltagswelt und den darin befindlichen Gegenständen lassen sich Möglichkeiten realisieren, die vorher undenkbar waren [3]. Immer neue technische Entwicklungen steigern die Leistung von eingebetteten Prozessoren, erhöhen die Bandbreite drahtloser Übertragungstechniken und verbessern die Genauigkeit von nutzbaren Sensoren.

Diese Entwicklungen im Hardwarebereich werden von Systemdesignern aufgegriffen und zu Lösungen kombiniert, welche die Fähigkeiten der Hardware ausloten und die Entwicklung weiter voran treiben. Dabei entstehen ubiquitäre Systeme, die in Übereinstimmung mit der Vision des Ubiquitous Computing das Potenzial haben, die Welt der IT zu revolutionieren [4].

Jedoch ist die Entwicklung dieser Systeme bisher rein technikzentriert. Sie demonstrieren die schöne neue Welt, zeigen Möglichkeiten der Technik und versetzen die Betrachter in Staunen. Die Bedarfe der Endkunden finden dabei in den vorgestellten Szenarien wenig Beachtung [5]. Entsprechend ist fraglich, ob es einen Markt für die neue Technik geben wird oder die Entwickler weiter wie bisher, also an den Anforderungen der Nutzer vorbei, entwickeln?

3 Nutzergetriebene Entwicklung

Die Entwicklung von technischen Systemen ist in der Regel kein Selbstzweck. So sollten auch die Systeme des Ubiquitous Computing dafür ausgelegt sein, die Anforderungen der Nutzer zu erfüllen und ihnen eine bestmögliche Unterstützung innerhalb des Funktionsumfangs zu bieten. Viele Ansätze der Produktentwicklung stellen deshalb den Anwender in den Mittelpunkt der Anforderungserhebung (User-Centered Design [6]) oder erklären ihn zum Designer (Open Design [7]) und binden ihn so aktiv in die Entwicklung mit ein [8, 9].

Die Idee, den Anwender als Designer für das System einzusetzen, welches er später nutzen möchte, basiert dabei auf einem einfachen Gedanken: Wer könnte besser beurteilen, was der Nutzer braucht, als der Nutzer selbst. Bei einem derartigen Vorgehen müssen jedoch Probleme berücksichtigt werden [10]:

- Es gibt nicht *den einen* Benutzer. Ein kommerzielles System zielt in der Regel auf mehrere spätere Benutzergruppen ab [11] und es ist nicht möglich, jeden potenziellen Nutzer das System bewerten zu lassen.
- Wie werden möglichst repräsentative Nutzer für die Anforderungserhebung ermittelt? In der von der Wirtschaftsinformatik geprägten Forschung findet sich hierzu der Ansatz, Fokusgruppen [12] zu nutzen und Lead User zu befragen.
- Es können nie die Anforderungen aller Benutzer berücksichtigt und vor allem umgesetzt werden, da verschiedene Nutzer komplementäre oder konfliktäre Anforderungen an ein System stellen werden.
- Auch bei Anforderungen, die grundsätzlich miteinander vereinbar sind, bleibt die Frage, wie diese möglichst stringent in einem System vereinigt werden können.

- Ebenso muss geklärt werden, wie Lösungsinformationen, also Ansätze, wie der Benutzer eine Anforderung umsetzen würde, mit in den Erstellungsprozess integriert werden können.

Ein Problem im Zusammenhang mit Ubiquitous Computing ist gerade die Neuartigkeit der Systeme. Die Vision von Weiser ist sehr abstrakt [1]. Konkrete Beispiele beschränken sich auf Umsetzung in Forschungseinrichtungen. Der breiten Öffentlichkeit sind der Begriff und das dahinterliegende Paradigma kaum bekannt. Wird der Nutzer an der Entwicklung beteiligt oder in die Mitte der Anforderungserhebung gestellt, steht er vor der Herausforderung, gleichzeitig die Systeme zu begreifen, ihre Regeln zu verstehen und innerhalb dieser Regeln Anforderungen für ein neues System zu sammeln. Er ist also durch die Neuartigkeit überfordert [11].



Abbildung 1: Personal Computing – Mobile Computing – Ubiquitous Computing

Den meisten Nutzern ist dieser Spagat möglich, solange sie mit der Art des Systems vertraut sind und sich die Interaktion mit dem System vorstellen können. Als Beispiel ist es relativ einfach zu erfassen, welche Anforderungen sie an eine Textverarbeitung haben, weil sie bereits mit der Art des Systems vertraut sind. Sie kennen die Regeln der Interaktion, sind gewohnt, dass sie Informationen über Tastatur und Maus (in seltenen Fällen per Mikrofon) eingeben und ihnen das Ergebnis auf einem Bildschirm präsentiert wird. Das Design eines Ubiquitous Computing Systems unterscheidet sich jedoch völlig von den bisherigen Interaktionskonzepten (Abbildung 1). Der Anwender soll Anforderungen an ein System verbalisieren, dessen Nutzung für ihn völlig implizit sein soll, also von ihm keine Aufmerksamkeit fordert [13]. Dieser Abstraktionsgrad wird die meisten Anwender überfordern.

4 Partizipative Entwicklung & Prototyping

Ein umfassendes Verständnis der Anforderungen von Benutzern eines umzusetzenden Systems ist für dessen erfolgreiche Einführung unerlässlich. Um ein solches Verständnis aufzubauen, ist es besonders bei anwendernahen Systemen wichtig, die Anforderungen direkt im späteren Nutzungskontext zu erfassen [14, 15, 16, 17, 18].

Die Forschungsbereiche der partizipativen Gestaltung und der Mensch-Maschine Interaktion haben Ansätze entwickelt, die es ermöglichen, gemeinsam mit verschiedenen Beteiligten an der Erstellung von Lösungen zu arbeiten [19]. Durch die Einbeziehung von Endkunden in den Entwicklungsprozess ergeben sich dabei zahlreiche Vorteile.

Nutzer kennen oft die Möglichkeiten verfügbarer Technologie nicht, vor allem, wenn diese, wie das Ubiquitous Computing, ein neues Paradigma verfolgt. Entwicklern ist dagegen der Kontext für die spätere Nutzung nicht bekannt. Mit der partizipativen Entwicklung wird die bestehende Wissenslücke zwischen Nutzer und Entwickler kleiner [20]. Gleichzeitig verschmilzt die Beteiligung von Nutzer und Entwickler im gesamten Produktlebenszyklus. Der Nutzer erhält das Produkt nicht erst, wenn die Entwicklung abgeschlossen ist [21]. Endanwender und Entwickler können so gemeinsam bereits früh im Entwicklungsprozess, und oftmals über mehrere Iterationen, Anforderungen an das zu erstellende System erheben und analysieren [15].

Das Prototyping von Anwendungen – also die teilweise Umsetzung von Elementen der Anwendung, die nutzbar und damit auch evaluierbar ist – stellt einen möglichen partizipativen Ansatz zur Anforderungserhebung dar [15, 16]. Dabei wird das finale System iterativ anhand zahlreicher Prototypen, die Zwischenschritte auf dem Weg zum gewünschten Endergebnis darstellen, aufgebaut. So werden wenige, initial erhobene oder aus festgelegten Rahmenbedingungen abgeleitete Anforderungen (bzw. deren Umsetzung) in jeder Iterationsstufe evaluiert, verfeinert und um weitere Anforderungen ergänzt. Auf diese Weise entsteht systematisch eine Vorstellung vom System bei den beteiligten Anwendern, mit deren Hilfe sie neue Anforderungen an das System erschließen können. Diese Art der gemeinsamen Arbeit wird von [15] im „Framework der kooperativen Zusammenarbeit“ (Abbildung 1) theoretisch beschrieben.

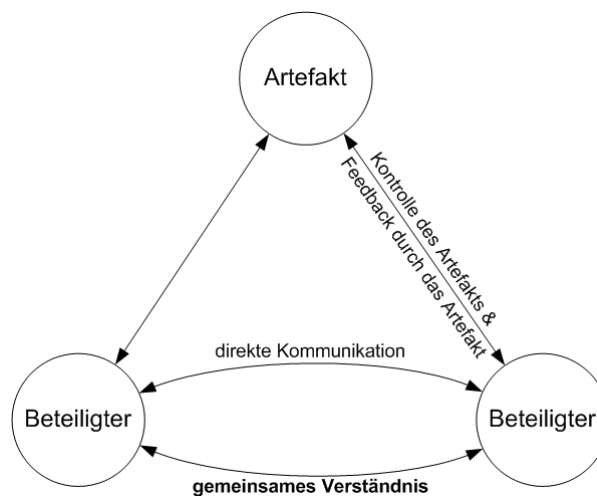


Abbildung 2: Framework der kooperativen Zusammenarbeit nach [15]

Kernaussage dieser Theorie ist, dass „Artefakte“ – also vom Menschen geschaffene Objekte wie z.B. Prototypen von Ubiquitous Computing Anwendungen – unterschiedlichen Prozessbeteiligten neben der direkten Kommunikation eine weitere Möglichkeit bieten, ein gemeinsames Verständnis über einen Sachverhalt aufzubauen. Auf diese Weise ist es demnach möglich, dass die Entwickler Ubiquitärer Systeme durch Nutzung von (realitätsnahen) Prototypen die Anforderungen der Kunden als primäre Anspruchsgruppe besser verstehen und Kunden ihre Anforderungen leichter kommunizieren können. Zudem lassen sich durch die Beobachtung der Nutzung der Prototypen im Rahmen von Pilotprojekten und Feldexperimenten in der späteren Zielumgebung durch Kundengruppen auch solche

Anforderungen erfassen, die von Kunden nicht direkt kommuniziert werden (konnten) [22, 23, 24].

5 Vorgelagerte Vision als erster Schritt des Requirements Engineering

Das Prototyping zu nutzen, um den späteren Nutzern eine Vorstellung des zu entwickelnden System zu schaffen und dadurch konkretere Anforderungen zu bekommen, wurde für das Ubiquitous Computing schon mehrfach erkannt [5, 25]. Der vorgestellte iterative Ansatz des Prototyping unterstützt hierbei die Entwickler und Benutzer ein gemeinsames Verständnis der Anwendung, und damit der Anforderungen an die Anwendung zu entwickeln. Jedoch müssen auch hier initial Anforderungen an den ersten Prototypen des Entwicklungsprozesses erfasst werden, der dann anschließend zur sinnvollen Integration der Nutzer eingesetzt werden kann.

Der klassische Prozess des Requirements Engineering der Softwareentwicklung beginnt mit der Anforderungsgewinnung [26]. In dieser Phase sollen bereits die (endgültigen) Anforderungen von Nutzern und anderen Stakeholdern gesammelt werden. Wie beschrieben ist aber eine Anforderungserhebung für Ubiquitous Computing Systeme nur sinnvoll, wenn es gelingt, dem Nutzer die neuen Möglichkeiten deutlich und nachvollziehbar zu machen.

Für eine erste Vorstellung des umzusetzenden Ubiquitous Computing Systems benötigt die Anforderungserhebung dementsprechend eine Phase, die im Vergleich zum klassischen Requirements Engineering dem Prozess vorgelagert ist. In dieser Phase müssen die Bedürfnisse des Benutzers erfasst und darauf basierend Ideen und nicht-technische Anforderungen an das System gesammelt werden [27]. Diese dienen dann als Grundlage, an der sich eine erste Umsetzung orientieren kann.

Für diese Szenarien ist darauf zu achten, dass die Entwicklung sich nicht nur an technischen Gegebenheiten orientiert und somit die technikgetriebene Entwicklung alle anderen Sichtweisen in den Hintergrund drängt. Vielmehr ist darauf zu achten, dass, auch ohne die konkrete Einbeziehung der Parteien, relevante Interessen von Stakeholdern gewahrt werden [28].

Für die frühe Gestaltung des Systems bieten sich Methoden aus dem Bereich Ideation [29] oder Collaboration Engineering [30] an. Mit deren Hilfe können Ideen produziert, erweitert und ausgestaltet werden. Dabei liegen die Stärken der Einbeziehung von Gruppentechniken vor allem auch in einem heterogenen Teilnehmerfeld, wobei die Beteiligung potenzieller Nutzer auf keinen Fall ausgeschlossen werden soll. Der Fokus liegt hier aber nicht darauf, das System später an die Anforderungen der Nutzer zu optimieren, sondern erst einmal einen Lösungsraum für das umzusetzende System zu erstellen. Dafür ist vor allem beim Ubiquitous Computing die Verinnerlichung der Vision durch die Teilnehmer entscheidend.

Die eigentliche Partizipation der Nutzer beginnt dann in den späteren Phasen, wenn es gelingt, diesen, zum Beispiel durch Prototypen, eine Vorstellung vom System zu vermitteln.

6 Fazit und Ausblick

Die Anforderungserhebung an Ubiquitäre Systeme weist zahlreiche Herausforderungen auf, die mit dem klassischen Requirements Engineering nicht abgedeckt werden können. Gerade durch die Neuartigkeit des Paradigmas sind die initiale Anforderungserhebung und die Einbeziehung des späteren Nutzers nicht problemlos möglich. Hier muss für die Anforderungserhebung im Ubiquitous Computing ein Vorgehen gefunden werden, das diese Herausforderungen adressiert und die damit verbundenen Probleme lösen kann. Einen ersten Schritt in diese Richtung machen partizipative Ansätze, die den Benutzer mit Hilfe von iterativ erstellten Prototypen immer näher an das spätere System heranzuführen und so sukzessive ein gemeinsames Verständnis von Entwickler und Anwender fördern. Um diese partizipativen Prozesse beginnen zu können ist es jedoch notwendig, zunächst die Bedürfnisse des Benutzers zu verstehen, um initiale Konzepte für eine Unterstützung durch ein ubiquitäres System entwickeln zu können. Diese dienen dann konsequent als Grundlage für die systematische Entwicklung eines derartigen Systems unter Beteiligung der späteren Nutzer.

Für die weitere Forschung zur Analyse von Anforderungen an ubiquitäre Systeme bedeutet dies konkret, dass zunächst Methoden und Techniken aus verschiedenen technischen und sozial relevanten Disziplinen für die jeweiligen Phasen der Anforderungserhebung identifiziert und auf ihre Eignung für die Entwicklung ubiquitärer Systeme evaluiert werden. Anschließend sind geeignete Methoden zu einem handhabbaren Vorgehen zur Anforderungserhebung ubiquitärer Systeme zu integrieren.

Neben dem Nutzer existieren noch weitere Quellen für Anforderungen im Ubiquitous Computing, z.B. um die Sozialverträglichkeit und Rechtskonformität der Anwendungen garantieren zu können. Diese gilt es vollständig zu identifizieren, damit eine umfassende Anforderungserhebung möglich wird. Zudem muss, wie gewöhnlich bei der Einführung neuer Technologien, das Problem des Vertrauens der Nutzer in die neue Technologie hinzu. Hier müssen jene Eigenschaften des Systems identifiziert werden, die das Vertrauen der potenziellen Nutzer in ubiquitäre Systeme unterstützen, um diese später gezielt gestalten zu können.

Acknowledgements: Die Forschungsarbeit wurde unterstützt durch das VENUS-Projekt. VENUS ist ein Forschungsprojekt der Universität Kassel, gefördert durch das Land Hessen als Teil der Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz (LOEWE). Weitere Informationen unter: www.iteg.uni-kassel.de/venus

Bibliography

- [1] Weiser, M.: The computer for the 21st century. SIGMOBILE Mobile Computing and Communications Review. 3(3): S. 3-11, 1999.
- [2] Baier, G. and M. Rothensee: Die Akzeptanz zukünftiger Ubiquitous Computing Anwendungen. In Hrsg.: A.M. Heinecke, H. Paul: Mensch und Computer 2006: Mensch und Computer im StrukturWandel. Oldenburg Verlag, München, 2006.
- [3] Mattern, F.: Ubiquitous Computing: Schlaue Alltagsgegenstände – Die Vision von der

- Informatisierung des Alltags. Bulletin SEV/VSE. 2004(19): S. 9-13, 2004.
- [4] Want, R.: An Introduction to Ubiquitous Computing. In Hrsg.: J. Krumm: Ubiquitous computing fundamentals, CRC Press, Boca Raton, 2009.
 - [5] Pichler, M.: A Novel View on Requirements Engineering for Ubiquitous Computing: The Innovation Perspective. Dissertation, Linz, 2007.
 - [6] Norman, D.A.: The design of everyday things. MIT Press, London, 1999.
 - [7] Vallance, R., S. Kiani and S. Nayfeh: Open Design of Manufacturing Equipment. in Proceedings of the CIRP 1st International Conference on Agile and Reconfigurable Manufacturing. Ann Arbor, 2001.
 - [8] Hoffmann, H., J.M. Leimeister and H. Krcmar: A Framework for Developing Personalizeable Mobile Services in Automobiles. Proceedings of the Fifteenth European Conference on Information Systems. St. Gallen, 2007.
 - [9] Hoffmann, H., J.M. Leimeister and H. Krcmar: Kundenintegration bei der Erstellung bedarfsgerechter automotiv Software. In Hrsg.: H. Heilmann: Humane Nutzung der Informationstechnologie, Akademische Verlagsgesellschaft AKA GmbH, Heidelberg, 2010.
 - [10] He, J. and W.R. King: The Role of User Participation in Information Systems Development: Implications from a Meta-Analysis. Journal of Management Information Systems. 25(1): S. 301-331, 2008.
 - [11] Ebb, B.R.W.: The role of users in interactive systems design: when computers are theatre, do we want the audience to write the script? Behaviour and Information Technology. 15(2): S. 76-83, 1996.
 - [12] Bruseberg, A. and D. McDonagh-Philp: Focus groups to support the industrial/product designer: a review based on current literature and designers' feedback. Applied Ergonomics. 33(1): S. 27-38, 2002.
 - [13] Fleisch, E., F. Mattern and S. Billinger: Betriebswirtschaftliche Applikationen des Ubiquitous Computing: Beispiele, Bausteine und Nutzenpotenziale. HMD Praxis der Wirtschaftsinformatik. 229: S. 5-15, 2003.
 - [14] Beyer, H. and K. Holtzblatt: Contextual design: defining customer-centered systems. Morgan Kaufmann Publ., San Francisco, 1998.
 - [15] Dix, A., J. Finley, G. Abowd and B. Russel: Human-Computer Interaction. Pearson Prentice Hall, Harlow, 2004.
 - [16] Sharp, H., Y. Rogers, and J. Preece: Interaction Design. Wiley, Chichester, 2007.
 - [17] Hoffmann, H., J. Föhling, J.M. Leimeister and H. Krcmar: Kundenintegration in die Innovationsprozesse bei hybriden Produkten - eine Bestandsaufnahme. In: Informatik 2009 - Im Focus das Leben. Lübeck, 2009.
 - [18] Hoffmann, H., J.M. Leimeister, and H. Krcmar: Tool Support for the Participatory Design of End User Oriented Applications in the Automobile. Proceedings of the 18th European Conference on Information Systems. Pretoria, 2010.
 - [19] Ahrens, G.: Das Erfassen und Handhaben von Produkthanforderungen - methodische Voraussetzungen und Anwendung in der Praxis. Dissertation, Berlin, 2000.
 - [20] Foltz, C., S. Killich, M. Wolf, L. Schmidt and H. Luczak: Task and Information

- Modeling for Cooperative Work. In: Systems, Social and Internationalization Design Aspects of Human-Computer Interaction - Proceedings of the HCI International. New Orleans, 2001.
- [21] Mambrey, P. and V. Pipek: Enhancing Participatory Design by Multiple Communication Channels. In Hrsg.: H.-J. Bullinger and J. Ziegler: Human-Computer Interaction: Communication, Cooperation, and Application Design, S. 387-391, Lawrence Erlbaum Ass., London, 1999.
- [22] Szyperski, N.: Zur wissensprogrammatischen und forschungsstrategischen Orientierung der Betriebswirtschaft. Zeitschrift für betriebswirtschaftliche Forschung. 23: S. 261-282, 1971.
- [23] Witte, E.: Feldexperimente als Innovationstest - Die Pilotprojekte zu neuen Medien. Zeitschrift für betriebswirtschaftliche Forschung. 49(5): S. 419-436, 1997.
- [24] Schwabe, G. and H. Krcmar: Piloting a Socio-technical Innovation. Proceedings of the 8th European Conference on Information Systems ECIS 2000. Vienna, 2000.
- [25] Michahelles, F.: Innovative application development for ubiquitous and wearable computing. Hartung-Gorre, Konstanz, 2005.
- [26] Kotonya, G. and I. Sommerville: Requirements engineering: Processes and techniques. Wiley, Chichester, 1998.
- [27] Pohl, K.: Requirements engineering: Grundlagen, Prinzipien, Techniken. dpunkt-Verl., Heidelberg, 2008.
- [28] Sommerville, I. and P. Sawyer: Viewpoints: principles, problems and a practical approach to requirements engineering. Annals of Software Engineering. 3: S. 101-130, 1997.
- [29] Knoll, S.W. and G. Horton: Changing the Perspective: Improving Generate thinkLets for Ideation. Proceedings of the Hawaii International Conference on System Sciences 2010 (HICSS). Koloa, Kauai, Hawaii, 2010.
- [30] Briggs, R.O., G.-J.D. Vreede, and J. Jay F. Nunamaker: Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems. Journal of Management Information Systems. 19(4): S. 31-64, 2003.