

Nutzung individueller Kontextinformationen zur Verbesserung von kollaborativen Arbeiten

Björn Joop, Tim Hussein, Jürgen Ziegler

Lehrstuhl für Interaktive Systeme und Interaktionsdesign
Abteilung für Informatik und angewandte Kognitionswissenschaften
Universität Duisburg-Essen
Lotharstrasse 65, 47057 Duisburg
{bjoern.joop, tim.hussein, juergen.ziegler}@uni-due.de

Abstract: Kollaborative Arbeitsprozesse werden entweder durch spezialisierte Arbeitsumgebungen gezielt unterstützt oder sind individualisierbar. Kontextbasierte Kollaborationsumgebungen können sich kontextabhängig auf die jeweilige Arbeitssituation einstellen und so dynamisch auf Änderungen der Prozesse oder der Benutzer reagieren. In diesem Beitrag stellen wir ein Kontext-Modell vor, welches sowohl individuelle Nutzerkontexte, wie auch gemeinsamen Kontext multipler Benutzer, erfassen, repräsentieren und als Basis für Adaptionseffekte nutzen kann.

1 Einleitung

Arbeit an schwach strukturierten kollaborativen Prozessen oder Aufgaben findet häufig entweder in Kollaborationsumgebungen, die eigens für diese Prozesse und Aufgaben erstellt wurden, oder in frei und offen gestalteten Umgebungen statt. Starre Umgebungen, die für einzelne Prozesse entwickelt wurden, schränken nicht nur den Verwendungszweck ein, sondern zwingen den Benutzer in der Regel, vorgegebene Arbeitsstrukturen zu erlernen, die sich nicht dynamisch an ändernde Anforderungen anpassen können. Frei gestaltete Arbeitsumgebungen hingegen können unerfahrene Benutzer durch zahlreiche Einstellungsmöglichkeiten überfordern, ermöglichen dafür Individualisierung der Arbeitsumgebung und -prozesse. Diese Individualisierung kann zu einer Überfrachtung der Arbeitsumgebung führen und unterstützt so nicht mehr das eigentliche Ziel: Die Unterstützung von kollaborativen Arbeitsprozessen.

Kontextbasierte Kollaborationsumgebungen können hier einen Mittelweg schaffen. In Abhängigkeit des aktuellen Kontexts können sowohl individuelle Anpassungen der Arbeitsumgebung, wie auch gruppenspezifische Änderungen der Kollaborationsprozesse durch das System für diese Umgebung erfolgen.

Im Folgenden werden wir unser Verständnis von Kontext erläutern und ein Kontext-Modell vorstellen, welches unserer Meinung nach den Anforderungen zur individuellen Unterstützung von Nutzern von Kollaborationsumgebung wie auch dynamischen Änderungen innerhalb des Kollaborationsprozesses gerecht wird.

2 Die Bedeutung von Kontext

Der Kontext eines Benutzers wird mit der Erfassung von Informationen über Ort, Zeit, die aktuelle Umgebung oder anderen sensorisch erfassbare Daten [SAW94, Ab97, De99, Ro05] gleichgesetzt. CSCW-Systeme [Ed04] betrachten Benutzer häufig als Konsumenten von Kontextinformationen. Im Ubiquitous Computing gilt das System als Konsument von Kontext [Do04].

In unserem Verständnis besteht ein Kontext-Modell aus drei Schichten, die im späteren Verlauf näher erläutert werden:

1. Modelliertes (Domänen-)Wissen unabhängig von Benutzer und Situation,
2. Informationen über die aktuellen Zustände in der physischen Welt, Arbeitsumgebung, Benutzerinformation, etc. sowie
3. Kontextualisierungsanweisungen, die den aktuellen Zustand in einen *kontextualisierten* Zustand überführen. Der kontextualisierte Zustand wird mithilfe eines Fokus über den aktuellen Zustand und Domänenwissen gebildet.

2.1 Domänenwissen-Schicht

Domainwissen repräsentiert sowohl das Domänen-Modell der Anwendung, wie auch Aspekte der physischen Umgebung, Benutzermodelle, Ressourcen und die digitale Arbeitsumgebung. Zu diesem Zweck werden Klassen von Individuen, Relationen und vordefinierte Relationstypen benutzt, um abstrakte Konzepte („Stadt“, „Person“, „Messenger“, „Personen leben in Städten“) wie auch konkrete Informationen („Al ist eine Person“, „Al lebt in Berlin“, „Al nutzt Skype“) zu modellieren (s. Abbildung 1). Das Domänenwissen kann unterschiedlich repräsentiert werden – z.B. in OWL¹ (vgl. Abbildung 2).

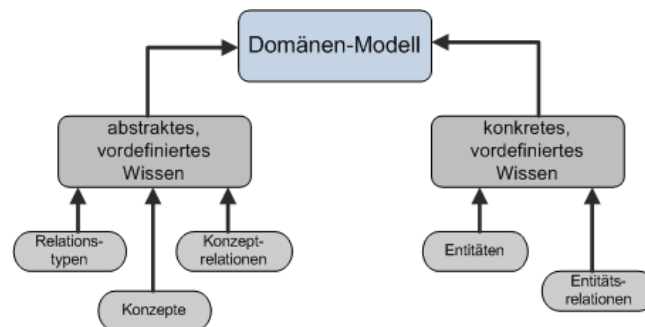


Abbildung 1: Das Domänenwissen ist im Domänen-Modell als abstraktes oder konkretes, vordefiniertes Wissen gespeichert.

¹ Web Ontology Language: <http://www.w3c.org/2004/OWL>

2.2 Zustands-Schicht

Die Zustandsschicht repräsentiert die aktuellen situationsabhängigen Informationen. Neben konkreten, vordefinierten Informationen aus dem Domänen-Modell beinhaltet die Zustandsschicht dynamisch veränderbare Informationen und Eigenschaften, die jedoch vordefinierten Klassen, Axiomen und Eigenschaftstypen des Domänen-Modells entsprechen müssen.

Wir verstehen den Zustand als Graphen $G_s = \{V, E, \sigma\}$ mit Knoten $v \in V$ als Individuen und Klassen, $e \in E$ als Relationen. Die Funktion σ entspricht einem Wahrheitswert (und findet häufig Verwendung in Fuzzy Logik), die zu einem Zeitpunkt T die Stärke einer Relation definiert. $\sigma: E \times T \rightarrow [0,1]$

G_s repräsentiert den aktuellen Zustand und kann durch einen RDF-Graph abgebildet werden. Abbildung 2 zeigt einen Zustand mit Individuen, Klassen und Relationen.

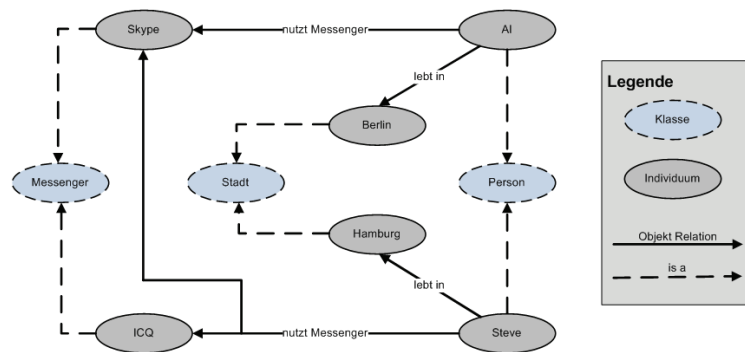


Abbildung 2: Ein Zustand mit Individuen (vollständige Ellipse), Klassen (gestrichelt) und Relationen (Linien).

Der Zustand zeigt zwei Personen, Tim und AI, die Städte, in denen sie leben, und ihre präferierten Messenger (hier: Skype und ICQ). Änderungen des Zustands, z.B. die Nutzung eines anderen Messengers, können von Sensoren erfasst werden und zusätzlich neues Wissen mithilfe des Domänen-Modells inferiert werden.

2.3 Kontextualisierungs-Schicht

Die Kontextualisierungsschicht beinhaltet Techniken (z.B. regelbasiert), mit deren Hilfe eine Teilmenge des Zustands als relevant in Bezug auf einen vorgegebenen Fokus markiert wird. Ein Fokus ist eine Teilmenge des aktuellen Zustands, der im Zentrum der Aufmerksamkeit des Systems oder des Benutzers liegt.

Kontextualisierungstechniken in Form von Regeln können beispielsweise durch einfache IF-THEN-Anweisungen dargestellt werden, die ausdrücken, was unter welchen Umständen als wichtig deklariert werden muss. So wird, falls zwei Personen miteinander online kommunizieren wollen, ein gemeinsam verwendeter Messenger (Skype, ICQ, etc.) als wichtig erachtet. In einer an SPARQL² angelehnten Notation wäre dies:

² SPARQL Protocol and RDF Query Language, W3C recommendation, <http://www.w3.org/TR/rdf-sparql-query>

IF (?person1 nutzt_messenger ?messenger) AND (?person2 nutzt_messenger ?messenger) THEN ?messenger

Es können auch zahlreiche weitere Techniken zur Kontextualisierung genutzt werden. Nach unserer Erfahrung [HZ08] bietet Spreading Activation [CL75] einen guten Ansatz um Mithilfe gewichteter Kanten die Kontextualisierung durchzuführen. Hierbei werden ein oder mehrere Knoten des Graphen als Ausgangsknoten mit einer initialen Ladung versehen, die sich entlang der Kanten an benachbarte Knoten weiter ausbreitet bis eine Abbruchbedingung eintritt. Eine mögliche, typische Abbruchbedingung ist beispielsweise eine maximale Entfernung zu den Ausgangsknoten. Auf diese Weise können zu einem oder mehreren vorselektierten Informationen im Fokus, verwandte Informationen im Graphen als Kontext markiert werden.

Der kontextualisierte Zustand wird somit als Kontext zu einem vorgegebenen Fokus bezeichnet und kann optional durch zusätzliche Filterung oder Gewichtung der Informationen weiter interpretiert und so für Adaptationszwecke genutzt werden.

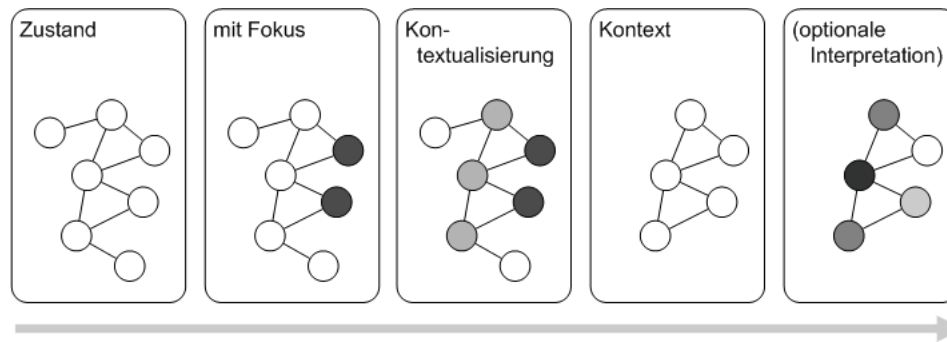


Abbildung 3: Schritt 1-4 zeigen den Vorgang zur Ermittlung des kontextualisierten Zustands als Filterprozess. Als optionaler Schritt folgt die Interpretation des kontextualisierten Zustands.

Abbildung 3 zeigt einen schematischen Vorgang zur Ermittlung eines Kontexts zu einem gegebenen Fokus. Der (optional) interpretierte Kontext kann z.B. für Adaptationszwecke eingesetzt werden, welche wiederum Veränderung im Zustandsgraphen und in der Arbeitsumgebung bewirken.

3 Kontext-basierte Adaption

Mithilfe des kontextualisierten Zustands werden zunächst alle Individuen und Klassen identifiziert, die für Adaptationszwecke von Bedeutung sind. Durch die Verwendung von Adaptionenregeln können globale und für Nutzer spezifische Adaptionen definiert und durchgeführt werden. Jede kontextbasierte Adaption ist abhängig von der Arbeitsumgebung. In [Ha09] wurde das Kontextmodell anhand von vier Szenarien evaluiert: Co-Recommendation, Co-Location, Co-Dependency und Co-Access.

So kann, falls viele Nutzer eines Systems ihre jeweilig zugewiesenen Aufgaben bearbeiten und es zu einer Konfliktsituation kommt, das System aufgrund der Benutzerkontexte ermitteln, welche Art von Kommunikation erwünscht ist und diese dann auch gezielt vermitteln. Ebenso kann das System auf mögliche Konflikte hinweisen, wie wenn beispielsweise eine wichtige Teilaufgabe zuvor noch erledigt werden muss. Solche unerfüllte Teilaufgaben werden im Arbeitskontext ermittelt und geeigneten Benutzern, die sich z.B. durch besondere Qualifikationen oder Interessen auszeichnen, als wichtig markiert.

Mithilfe dieses Kontext-Modells lassen sich so auch Awareness-Funktionalitäten innerhalb von Anwendungen oder Kollaborationsumgebungen realisieren, indem beispielsweise der Kontext einer gemeinsam genutzten Datei ermittelt, und – gemäß Adaptationsregeln – entsprechende Awareness-Informationen angezeigt werden. Der Nutzer eines Systems kann so auf neue, für seinen Kontext wichtige, Dokumente oder Benutzer hingewiesen werden, oder einfach nur im Fall eines gleichzeitigen Zugriffs auf ein Dokument durch mehrere Benutzer mit unterstützenden Informationen versorgt werden.

Adaptionsmöglichkeiten sind unbegrenzt und können – trotz Verarbeitung desselben Kontextes – für jede Umgebung oder Anwendung und ihre Zwecke angepasst werden.

4 Diskussion

Obwohl zahlreiche Modelle und Möglichkeiten existieren, Kontext in einem System zu modellieren [CK00, CFJ03, Wa04, SL04], bietet keiner dieser Ansätze die Möglichkeit, gezielt relevante Kontextaspekte durch eine Anfrage (Fokus) zu ermitteln. Das vorgestellte Kontext-Modell hat den Vorteil, dass über dieselben Zustände und dasselbe vormodellierte Domain-Wissen sowohl der individuelle Kontext eines jeden Nutzers, der Kontext des kollaborativen Prozesses wie auch der Kontext des aktuellen Systemzustands zu ermitteln ist.

Durch die Verwendung eines Fokus und frei editierbarer oder vordefinierter Kontextualisierungstechniken ermöglicht dieses Modell die Filterung des Zustands auf notwendige Aspekte, die eine Aussage über relevante Kontextinformationen ermöglichen. Hierbei ist es unerheblich, ob ein einzelner oder mehrere Nutzer des Systems im Fokus enthalten sind, oder aber ob der Kontext zu bestimmten (Teil-)Aufgaben ermittelt werden muss. In allen Fällen können die gleichen Techniken genutzt werden – wenngleich sich Kontextualisierungsregeln und Adaptation unterscheiden.

Abhängig vom aktuellen Zustand können global definierte Kontextualisierungs- und Adaptationsregeln die Arbeitsumgebung gezielt für alle Benutzer beeinflussen. Mit denselben Mitteln ist es möglich, den Kontext eines einzelnen Benutzers oder ganzer Benutzergruppen gezielt zu bestimmen um so direkt auf Bedürfnisse und Interessen der Benutzer einzugehen. Letztlich eignet sich das vorstellte Kontext-Modell auch für den Einsatz in Recommender-Systemen, wie bereits in [HZ08] gezeigt wurde.

Obwohl erste kontextbasierte, kollaborative Arbeitsumgebungen bereits existieren, verfolgen existierende Ansätze entweder die Individualisierung der Arbeitsumgebung an die Bedürfnisse (den „Kontext“) der Benutzer oder aggregieren Informationen aller Benutzer in einen einzigen Kontext, der zumeist aufgrund der Informationsvielfalt schwer zu interpretieren ist und so wiederum zu unvorhersehbaren Anpassungen der Arbeitsumgebung führen kann. Bei dem vorgestellten Ansatz treten derartige Schwierigkeiten nicht auf.

Danksagung

Diese Arbeit ist im Rahmen des von der Deutschen Forschungsgemeinschaft geförderten Projekts CONTIci entstanden.

Literaturverzeichnis

- [Ab97] Abowd, G. D. et. al.: Cyberguide: a mobile context-aware tour guide, *Wireless Networks* Nr. 5 Vol. 3, 1997, pp. 421-433.
- [CFJ03] Chen, H. , Finin, T., Joshi, A.: Using OWL in a Pervasive Computing Broker, *Workshop on Ontologies in Agent Systems*, 2003.
- [CK00] Chen, G., Kotz, D.: A survey of contextaware mobile computing research, *Dartmouth College Technical Report*, 2000.
- [CL75] Collins, A. M., Loftus, E. F.: A spreading-activation theory of semantic processing. *Psychological Review*, 82:407–425, 1975
- [De99] Dey, A. K. et. al.: Towards a Better Understanding of Context and Context-Awareness, in *Proceedings of the 1st int. symposium on Handheld and Ubiquitous Computing*, 1999.
- [Do04] Dourish, P.: What we talk about when we talk about context, *Personal Ubiquitous Computing* Nr. 1 Vol 8 (2004), pp. 19-30.
- [Ed05] Edwards, W. K.: Putting Computing in Context: An Infrastructure to support extensible context-enhance collaborative Applications, *ACM Trans. Computer-Human Interactions* Nr. 4 Vol. 12 (2005), pp. 446-474.
- [Ha09] Haake, J., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J.: Context Modeling for Adaptive Collaboration. *Technischer Bericht 2009-02*, Universität Duisburg-Essen, 2009
- [HZ08] Hussein, T., Ziegler J.: Adapting web sites by spreading activation in ontologies, *Int. Workshop on Recommendation and Collaboration*, Ed. L. Bergmann et al., 2008.
- [Ro05] Roth, J.: *Mobile Computing* (dpunkt.verlag, 2nd. Edition, 2005).
- [SAW94] Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications, (IEEE) *Workshop on Mobile Computing Systems and Applications*, 1994.
- [SL04] Strang, T. , Linnhoff-Popien, C.: A Context Modeling Survey, *Workshop on Advanced Context Modelling, Reasoning and Management*, Sixth International Conference on Ubiquitous Computing, 2004.
- [Wa04] Wang, X. H. et. al.: Ontology Based Context Modeling and reasoning using OWL, in *Proceedings of the Second IEEE Annual conference on Pervasive Computing and Communications Workshops*, 2004.