

Modellierung von Kontexten in kooperativen Umgebungen: Anspruch und Wirklichkeit

Tom Gross¹, Wolfgang Prinz²

¹ Fakultät Medien, Bauhaus-Universität Weimar

² Fraunhofer Institut für Angewandte IT, St. Augustin

Zusammenfassung

Um effektiv und effizient in Gruppen zusammenzuarbeiten, benötigen BenutzerInnen Informationen über Vorgänge und Zustände in der Gruppe. Neben der Erfassung, Verarbeitung und Darstellung der Aktivitätsinformationen ist deren Kontextualisierung ein zentrales Anliegen. Die adäquate Kontextualisierung gewährleistet, dass die BenutzerInnen in ihrer jeweiligen Arbeitssituation mit entsprechend ausgewählter und dargestellter Information versorgt werden können. In diesem Beitrag motivieren wir die Notwendigkeit der Modellierung von Kontexten in kooperativen Umgebungen, und wir gehen kurz auf bisherige Ansätze ein. Wir präsentieren ein einfaches, aber flexibles und generisches, Kontextmodell und berichten über eine Untersuchung über seine Umsetzbarkeit und Treffschärfe.

1 Einleitung

Um effektiv und effizient in Gruppen zusammenarbeiten zu können, benötigen BenutzerInnen Informationen über Vorgänge und Zustände in der Gruppe. Diese Informationen werden als Awareness (Dourish & Bellotti 1992; Schmidt 2002) bezeichnet, was oft als Gruppenwahrnehmung (Gross & Prinz 2000) oder Transparenz (Fuchs et al. 1996) übersetzt wird.

Für die Erfassung, Verarbeitung und Darstellung wurden in der Literatur zum einen anwendungsspezifische Lösungen wie beispielsweise ClearBoard (Ishii et al. 1994), zum anderen Infrastrukturen wie beispielsweise ELVIN (Fitzpatrick et al. 1999), NSTP (Patterson et al. 1996) oder NESSIE (Prinz 1999), welche wesentlich flexiblere Anwendbarkeit bieten, vorgeschlagen.

Neben der Erfassung, Verarbeitung und Darstellung von Informationen zur Gruppenwahrnehmung kommt der Kontextualisierung der Informationen eine zentrale Bedeutung zu. Nur

damit ist gewährleistet, dass die BenutzerInnen in ihrer jeweiligen Arbeitssituation mit entsprechend ausgewählter und dargestellter Information versorgt werden. An die adäquate Modellierung der Kontexte sind die folgenden Anforderungen zu stellen:

- genaue und wirklichkeitsnahe Modellierung, um eine scharfe Zuordnung von erfassten Ereignissen und Situationen des Alltags zu erreichen.
- einfache Modellierung und geringer Zusatzaufwand zur Anpassung des Modells an Veränderungen in der modellierten Wirklichkeit
- kein oder geringer Zusatzaufwand für die BenutzerInnen

In der Literatur wurden dazu bereits einige Ansätze vorgestellt; diese erfüllen in der Regel die obigen Anforderungen nur zum Teil.

Im AREA-System werden Situationen in einem Objekt-Relationen-Modelle beschrieben, das Relationen zwischen Objekten (einzelnen Personen, Artefakten oder Aggregationen wie etwa Gruppen von Personen) abbildet. Die BenutzerInnen können für einzelne Situationen spezifizieren, an welchen Ereignissen und Artefakten sie interessiert sind und wann und in welcher Intensität sie informiert werden möchten (Fuchs 1999; Sohlenkamp et al. 1998). Das AREA-Modell ist sehr genau und bringt keinen Zusatzaufwand für die BenutzerInnen, kann allerdings Ereignisse und Situationen nicht immer eindeutig zuordnen. Das einmal erstellte Modell ist nur schwer an in der Realität immer wieder auftretenden Veränderungen anpassbar.

Im Atmosphere-Modell werden Situationen als „Spheres“ bezeichnet und ähnlich modelliert wie in AREA. Die Besonderheit von Atmosphere ist, dass die BenutzerInnen ihre Aktionen auf Artefakten durch sogenannte „Contextors“ klassifizieren (Rittenbruch 1999). Atmosphere erlaubt eine detaillierte Modellierung und kann Ereignisse und Situationen eindeutig zuordnen. Allerdings ist auch hier das Modell nur schweranpassbar und es wird Zusatzaufwand von den BenutzerInnen gefordert.

Die Modelle AETHER und MoMA sind ähnlich. Im AETHER-Modell werden Beziehungen zwischen Objekten in einem semantischen Netzwerk abgebildet (Sandor et al. 1997). Im Model of Modulated Awareness (MoMA) wird zusätzlich noch eine Reaktions-Diffusions-Metapher eingeführt, welche bewirkt, dass—ähnlich wie in der Physik—sich beim Aufeinandertreffen bzw. Kontakt zwischen zwei oder mehrere Entitäten, sich deren Status ändert. Gruppenwahrnehmung wird durch Felder erzeugt oder konsumiert (Simone & Bandini 1997). Für beide Modelle gilt bezogen auf die obigen Anforderungen dasselbe: sie sind sehr durchdacht und erlauben eine exakte Modellierung der Realität; es entsteht kein Zusatzaufwand seitens der BenutzerInnen. Die Zuordnung von Ereignissen und Situationen ist treffsicherer als bei AREA, aber schlechter als bei Atmosphere. Allerdings ist vor allem bei diesen komplexen Modellen und Mechanismen die laufende Anpassung sehr aufwendig.

Im nächsten Abschnitt präsentieren wir unseren eigenen Modellierungsansatz, welcher Informationen semantisch zu Kontexten zusammenfasst und die obigen Anforderungen erfüllt, sowie dessen technische Umsetzung in der Event and Notification Infrastruktur. Wir beschreiben eine analytische Untersuchung, diskutieren die Resultate und resümieren den Beitrag.

2 Modellierungsansatz und technische Umsetzung

Wir schlagen als zentrales Konstrukt für die Modellierung von Informationen zur Gruppenwahrnehmung Kontexte vor. Im Duden (1978) wird Kontext definiert als „(...zu: contexere = eng verknüpfen): ... der umgebende inhaltliche (Gedanken-, Sinn)zusammenhang, in dem eine Äußerung steht, u. der Sach- und Situationszusammenhang, aus dem heraus sie verstanden werden muss...“.

2.1 Gruppenwahrnehmung und Kontexte

Ein Kontext in kooperativen Anwendungen kann somit als die Umstände (z.B.: Ort oder Zeit), die miteinander verwoben sind (z.B.: durch chronologische Abfolge oder organisatorische Zusammenhänge), in denen etwas (z.B.: ein Benutzer, eine Gruppe, ein Artefakt) existiert oder sich etwas ereignet (z.B.: eine Handlung eines Benutzers oder einer Maschine). Kontexte können beispielsweise für Abteilungen in Organisationen oder für größere Projekte angelegt werden. Das Kontextmodell per se wurde bereits an anderer Stelle (Gross & Prinz 2000) ausführlich beschrieben und wird an dieser Stelle nur kurz zum Verständnis der nachfolgenden Studie zusammengefasst.

Im vorgeschlagenen Modell beschreiben wir einen Kontext mit den folgenden Attributen: Name: eine eindeutige Bezeichnung; Administrator: eine oder mehrere Personen, die den Kontext anlegt und pflegt; Mitglieder: BenutzerInnen, die für den Kontext von Bedeutung sind; Orte: elektronische und physische Orte, an denen sich kontextrelevante Dinge ereignen; Artefakte: elektronische Dokumente, die in einem Kontext verwendet werden; Anwendungen: Software-Applikationen, die in einem Kontext verwendet werden; Ereignisse: Vorgänge in einem Kontext; Zugriffskontrolllisten und Verbindungen zu anderen Kontexten: diese Attribute regulieren den Zugriff und die Verknüpfung.

2.2 Event and Notification Infrastruktur

Die Event und Notification Infrastruktur—kurz ENI—in die wir Kontexte integriert haben, bietet eine generische und leicht erweiterbare Infrastruktur zur Unterstützung der Gruppenwahrnehmung. Sie beinhaltet mächtige, aber einfache Mechanismen für die Generierung und Präsentation von Aktivitätsinformationen (Prinz 1999). ENI basiert auf einer Client-Server-Architektur inklusive Sensoren für die Erfassung von Informationen und Indikatoren für die Präsentation von Informationen zur Gruppenwahrnehmung.

Sensoren sind mit menschlichen Akteuren, gemeinsamen Arbeitsbereichen, den darin enthaltenen Dokumenten und anderen Kooperationswerkzeugen assoziiert. Sie erfassen Ereignisse und leiten diese an den zentralen ENI-Server weiter. Der ENI-Server verarbeitet die Ereignisse, bereitet sie auf und leitet sie an die Indikatoren weiter. Die Indikatoren stellen Informationen für die interessierten und berechtigten BenutzerInnen dar.

Die Kommunikation zwischen den Indikatoren und Sensoren einerseits und dem ENI-Server andererseits kann entweder direkt über ein Common Gateway Interface (CGI) erfolgen, oder indirekt über einen ENI-Client. Der Vorteil des Clients ist, dass Informationen vom ENI-Server auch über die Application Programming Interfaces (API) aktiv an die Indikatoren geschickt werden können (Push-Prinzip). Im Fall des CGI muss die Initiative aus technischen Gründen von den Indikatoren ausgehen—d.h. diese fragen in regelmäßigen Abständen beim Server nach Veränderungen und stellen diese dann gegebenenfalls dar. Der große Vorteil von CGI ist allerdings, dass über diese Schnittstelle fast jede beliebige Anwendung mit dem ENI-Server kommunizieren kann—einzige Anforderung an die Anwendung ist, dass sie über HTTP und CGI verfügen muss.

2.3 Kontexte in ENI

Da die einzelnen Ereignisse in ENI bereits als Attribut-Wert-Paare vorliegen, ist es nahe liegend und für die Verarbeitung am einfachsten, Kontexte auch als Attribut-Wert-Paare darzustellen. Die bereits oben genannten Attribute werden dann in entsprechende Attribute, die für jeden Kontext festgelegt werden, abgebildet.

Kontexte werden über die in 2.1 identifizierten Attribute beschrieben. Jedes Attribut kann durch beliebig viele Werte beschrieben werden; d.h. dass es für manche Attribute auch keine Werte geben muss, für andere Attribute kann es eine Liste von Werten geben.

Über die Attribute und ihre Werte erfolgt dann die entsprechende Zuordnung von Ereignissen zu Kontexten (siehe Abbildung 1). Und zwar wird zunächst für jedes Ereignis, bei dessen Auftreten ermittelt, welchem Entstehungskontext es zugeordnet werden kann (1). Dies erfolgt über den Vergleich der Ereignisattribute mit den Kontextattributen. Anschließend werden für jeden Benutzer alle ausgelösten Ereignisse analysiert (2). So kann jedem Benutzer ein aktueller Arbeitskontext zugewiesen werden. Im Idealfall kann eine eindeutige Zuordnung ermittelt werden. In vielen Fällen ist nur eine Zuordnung zu verschiedenen Kontexten

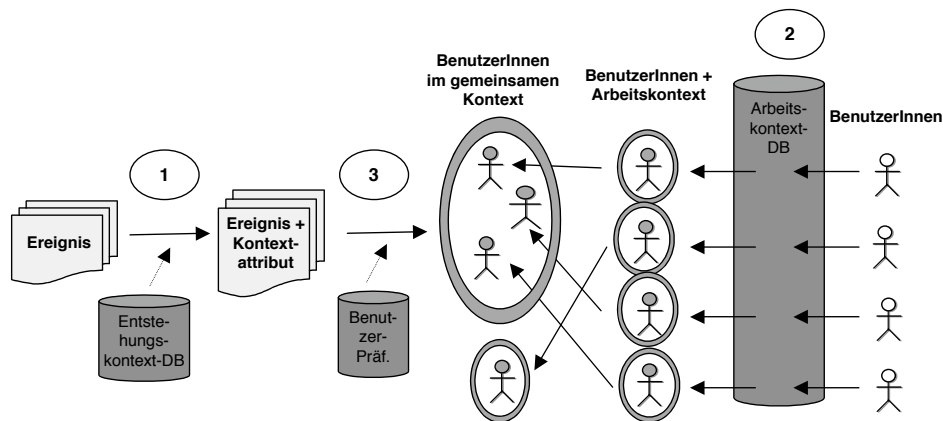


Abbildung 1. Ereignis- und Kontextverarbeitung.

mit unterschiedlichen Wahrscheinlichkeiten möglich.

Um die BenutzerInnen nun in der aktuellen Situation mit adäquater Information versorgen zu können, werden die BenutzerInnen nur über jene Ereignisse von anderen Kooperationspartnern informiert, die aus einem Kontext stammen in dem sie auch gerade tätig sind. Dabei können die BenutzerInnen selbst ihre Präferenzen bzgl. Kontext und zeitlicher Ausgestaltung angeben (3). Bezüglich des Kontexts können die BenutzerInnen für jeden eigenen Tätigkeitskontext festlegen, aus welchen anderen Kontexten Informationen angezeigt werden sollen. So können BenutzerInnen festlegen, dass sie nur mit Informationen aus demselben Kontext versorgt werden möchten; es kann aber sinnvoll sein, dass man in einem bestimmten Tätigkeitskontext auch aktuell über Tätigkeiten anderer aus einem ähnlichen Projektkontext informiert wird. Bezüglich der zeitlichen Ausgestaltung können die BenutzerInnen angeben, wann sie informiert werden möchten. Hier sind verschiedene Einstellungen möglich (z.B.: sofortige Benachrichtigung, Benachrichtigung in festen Zeitintervallen, Benachrichtigung entsprechend des Alters von Ereignissen).

Abbildung 2 zeigt die Architektur von ENI mit den entsprechenden Erweiterungen, die für die Umsetzung der Kontexte benötigt werden. Das Kontextmodul ist dabei für die Analyse der einzelnen Ereignisse und für die Zuordnung zu Kontexten zuständig und das Situationsmodul ist für die Analyse der Ereignisse der einzelnen Benutzer zuständig.

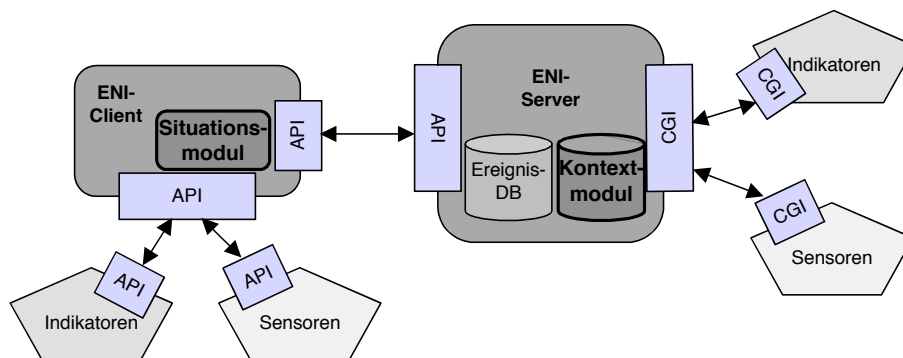


Abbildung 2. Event and Notification Infrastruktur mit Kontexterweiterungen.

3 Analyse des Modells

In diesem Abschnitt beschreiben wir die Anwendung des Kontextmodells in einem gemeinsamen Arbeitsbereich sowie das Ergebnis einer Analyse des Kontextmodells.

3.1 Methode—Kontexte und gemeinsame Arbeitsbereiche

Kooperative Arbeit wird oft über gemeinsame Arbeitsbereiche abgewickelt (Pankoke-Babatz & Syri 1997). In diesen gemeinsamen Arbeitsbereichen haben i.d.R. eine Gruppe von MitarbeiterInnen Zugriff auf eine Menge von Dokumenten. Oft werden solche Arbeitsbereiche für gemeinsame Projekte eingerichtet. Wenn man nun einen Kontext für ein solches Projekt anlegen möchte, dann sind viele Werte für die Attribute, die zur Beschreibung des entsprechenden Kontexts in ENI benötigt werden, bereits vorhanden (z.B.: Kontextmitglieder, Kontextdokumente). Umgekehrt fällt in einer solchen Umgebung die Zuordnung von einzelnen Handlungen von Akteuren zu Arbeitsbereichen bzw. Kontexten leichter.

Wir haben daher für die Anwendung des Kontextmodells ein solches System zur Unterstützung von gemeinsamen Arbeitsbereichen herangezogen, und zwar BSCW—ein Web-basiertes System zur Unterstützung von gemeinsamen Arbeitsbereichen (Appelt 1999). Obgleich es sich bei BSCW um ein sehr spezifisches System handelt, glauben wir, dass aufgrund der großen Ähnlichkeit zu anderen Systemen zur Unterstützung von gemeinsamen Arbeitsbereichen, die Ergebnisse auch für andere Systeme zutreffen.

Wir haben für das BSCW System Sensoren entwickelt, die für jede Benutzer-Aktivität in BSCW ein Ereignis generieren und dieses an den ENI-Server schicken. Abbildung 3 zeigt als Beispiel ein von BSCW erzeugtes und an den ENI-Server gesendetes Ereignisses. Dieses Ereignis enthält alle von BSCW generierten Attribute; in anderen Anwendungen können zum einen verschiedene, zum anderen eine andere Anzahl von Attributen verwendet werden.

```
<EVENT>
  <ATTRIBUTE type="sensor" value="BSCW"/>
  <ATTRIBUTE type="event-originator" value="Schmitz"/>
  <ATTRIBUTE type="operation" value="ReadEvent"/>
  <ATTRIBUTE type="artefact" value="weiser_cacm.pdf"/>
  <ATTRIBUTE type="bscw-object-id" value="135578"/>
  <ATTRIBUTE type="folder" value="Related Work"/>
  <ATTRIBUTE type="bscw-folder-id" value="132978"/>
  <ATTRIBUTE type="bscw-class" value="Document"/>
  <ATTRIBUTE type="bscw-content" value="application/pdf"/>
  <ATTRIBUTE type="acl" value="Schmitz, Müller, Klein"/>
  <ATTRIBUTE type="date" value="2003-02-03 09:09:45"/>
  <ATTRIBUTE type="expires" value="2003-02-04 09:09:45"/>
</EVENT>
```

Abbildung 3. Beispiel eines von BSCW erzeugten Ereignisses.

Das Attribut sensor gibt die Anwendung an, welche das Ereignis erfasst hat—dies ist bei der Anwendung von BSCW immer BSCW. Das Attribut event-originator enthält den Login-Namen des Benutzers, der die Aktion (Attribut operation) auf einem bestimmten Dokument (Attribut artefakt) durchgeführt hat. die Attribute bscw-object-id und bscw-folder-id können zusätzliche Informationen über das Elternverzeichnis des jeweiligen Dokuments enthalten.

Die Attribute `bscw-class` und `bscw-content` geben weitere Informationen über den Dokumenttyp. Das Attribut `acl` enthält eine Liste von Benutzern (Login-Namen), welche auf dieses Ereignis zugreifen bzw. über dieses Ereignis informiert werden dürfen. Schließlich gibt das Attribut `date` Informationen über den Entstehungszeitpunkt und das Attribut `expires` Informationen über die Gültigkeit des Ereignisses.

3.2 Ergebnisse

In unserer Untersuchung haben wir die Verwendung von BSCW über einen Zeitraum von 18 Monaten beobachtet und dabei insgesamt ungefähr 16.000 Ereignisse erfasst. Alle Ereignisse haben eine Struktur wie oben beschrieben. Insgesamt wurden auf dem untersuchten BSCW-Server elf verschiedene Projekte über entsprechende Arbeitsbereiche abgewickelt. Die Arbeitsbereiche enthielten zwischen 25 und 1.200 Objekte, wobei die meisten im Bereich von 100 Objekten lagen. Die Anzahl der Mitglieder pro Arbeitsbereich lag zwischen acht und zwölf, mit einer Ausnahme von 30 Mitgliedern für einen Arbeitsbereich, der für eine Seminarveranstaltung an einer Universität verwendet wurde. Im Folgenden diskutieren wir zuerst mögliche Attributvergleiche und reflektieren dann die Anwendbarkeit. Dabei ist zu klären, welche Attribute und von welchen Werten für die jeweiligen Attribute Vergleiche für eine adäquate Unterstützung Sinn machen.

Das vorgestellte Kontextmodell erfordert die Spezifikation der folgenden Attribute: `context-app`, `context-member` oder `context-location`, `context-event`, oder `context-artefact`. In unserer Untersuchung haben alle Ereignisse BSCW als Anwendung; daher kann dieses Attribut nicht für eine Zuordnung von Ereignissen—welche eine Unterscheidung von Ereignissen erfordert—verwendet werden. Es bleiben die folgenden Vergleichsmöglichkeiten:

- a) die Attribute `event-originator` und `context-member`
- b) die Attribute `artefact` und `context-artefact`
- c) die Attribute `event-attributes` und `context-event`

Zu a) Der Vergleich der beteiligten BenutzerInnen konnte in unserer Untersuchung nicht zu einer trennscharfen Unterscheidung verwendet werden. Der Grund dafür lag in der Tatsache, dass die meisten Arbeitsbereiche (sieben von elf Arbeitsbereichen) ähnliche Mitgliederlisten (mit einem Unterschied von zwei bis drei BenutzerInnen) hatten, und daher die Unterscheidung über `context-member` nicht zielführend war. Eine eindeutige Unterscheidung bzw. Zuordnung war daher nur in weniger als 20 Prozent der Fälle möglich.

Wir untersuchten bzgl. der beteiligten BenutzerInnen ergänzend auch die in der `acl` enthaltenen BenutzerInnen und verglichen diese mit `context-member`. Hier hatten nur drei der elf Arbeitsbereiche große und nicht hinreichend unterscheidbare Ähnlichkeiten.

Zu b) Vergleicht man die vorkommenden Dokumente, so ergibt sich eine eindeutige Zuordenbarkeit. Allerdings ist dieser Vergleich sehr aufwendig und daher langsam: es muss der Dokumentname des entsprechenden Ereignisses mit den Namen aller Dokumente aus allen Arbeitsbereichen verglichen werden. Entsprechend muss auch die Kontextbeschreibung alle

Dokumentenamen aus den gemeinsam genutzten Arbeitsbereichen enthalten. Damit entsteht in der Kontextbeschreibung eine nahezu redundante Abbildung der Arbeitsbereiche. Auch dies ist nicht gewünscht.

Zu c) Wir haben daher nach einem skalierbaren Maß gesucht und dazu das Elternverzeichnis, respektive das Attribut `bscw-folder-id`, genutzt. In der Kontextbeschreibung erfolgt daher die Beschreibung der `context-artefact` durch die Elternverzeichnisse der Dokumente. Entsprechend erfolgt die Abbildung eines Ereignisses in einen Kontext über den Vergleich des Elternverzeichnisses. Damit ist zwar noch immer eine Redundanz verbunden. Da die Anzahl der Verzeichnisse im Vergleich zur Anzahl der Dokumente nur 13 Prozent ausmacht, ist dieser Ansatz wesentlich skalierbarer.

Damit der Abgleich zwischen der Kontextbeschreibung und den sich dynamisch ändernden Arbeitsbereichen nicht manuell erfolgen muss, ist ein automatisches Verfahren notwendig. Dies basiert ebenfalls auf einer Auswertung der eingehenden Ereignisse. Wenn ein neues Verzeichnis in einem Arbeitsbereich eingerichtet wird, schickt BSCW ein Ereignis mit entsprechenden Informationen an den ENI-Server. Das Kontextmodul im ENI-Server bildet dieses Ereignis dann über den Vergleich der Elternverzeichnisse in den entsprechenden Kontext ab und erweitert zusätzlich die Kontextbeschreibung um den Verzeichniseintrag. Damit wird die Kontextbeschreibung automatisch an die neue Konstruktion des Arbeitsbereichs angepasst. Bei Ereignissen, die die Löschung eines Verzeichnisses anzeigen, wird die Kontextbeschreibung ebenfalls angepasst.

3.3 Diskussion

Die obigen Vergleichsmöglichkeiten von Attributen, respektive deren Werten, zeigt, dass die Anwendung des Kontextmodells einer vorhergehenden gründlichen Analyse der generierten Daten bedarf.

Im einfachsten Fall analysiert die das Ereignis generierende Anwendung den Sinnzusammenhang der Ereignisentstehung und liefert entsprechende Kontextinformationen mit. Dies ist allerdings eine wenig flexible Lösung, weil die Kontextmodelle dann in allen Anwendungen gespeichert bzw. ständig aktualisiert werden müssten. Anstelle einer Redundanz in der zentralen Kontextbeschreibung erzielt man eine hohe Redundanz durch die Verteilung der Kontextbeschreibung auf alle involvierten Sensoren. Außerdem erlaubt dieser Ansatz keine benutzerspezifische Kontextverarbeitung, in welcher auf Präferenzen einzelner BenutzerInnen oder Gruppen Rücksicht genommen wird. Dies kann nur bei einer zentralen Auswertung erfolgen. Der gewählte Ansatz mit im ENI-Server zentral gehaltenen und administrierten Kontextbeschreibungen bietet daher mehr Flexibilität und leichtere Handhabbarkeit.

Darüber hinaus ist es für die klare und wenigdeutige Zuordnung von Ereignissen zu Kontexten wichtig, dass sowohl so viel wie möglich Information über jedes Ereignis erfasst als über Kontexte gespeichert werden. Da die manuelle Verwaltung dieser detaillierten Informationen sehr mühsam und kaum zu bewerkstelligen ist, ist eine automatische Erfassung der Ereignisse und die automatische Aktualisierung der Kontextbeschreibungen wie oben beschrieben unerlässlich.

4 Zusammenfassung

In diesem Beitrag haben wir ein Konzept für die Modellierung von Kontexten zur verbesserten Information der BenutzerInnen präsentiert und dessen Anwendung mit einem System zur Unterstützung von gemeinsamen Arbeitsbereichen beschrieben. Wir haben über eine analytische Untersuchung berichtet und deren Resultate diskutiert.

Die Untersuchung zeigt, dass eine generische Abbildung von Ereignissen über die Beschreibung von Kontextmustern nicht unmittelbar möglich ist. Agiert ein Benutzer in sehr disjunkten Gruppen, dann kann eine Zuordnung über die Gruppenmitgliedschaft erfolgen. Ist dies, wie in unserem Beispiel, nicht der Fall, dann müssen feingranularere Unterscheidungsmerkmale genutzt werden. Als Konsequenz daraus erhöht sich die Redundanz zwischen der Kontextbeschreibung und den Arbeitsmaterialien. Die Untersuchung zeigt, dass das Problem in der Realisierung kontextsensitiven Anwendungen in der Identifikation von Ereigniseigenschaften besteht, die eine trennscharfe Zuordnung ermöglichen. Diese Eigenschaften sind jedoch nicht nur anwendungsabhängig sondern auch benutzer- und nutzungsabhängig. Bei unsere Lösung wird eine automatische Aktualisierung der Kontextbeschreibung durch einen Ansatz erreicht, bei dem das Kontextmodul aus den eingehenden Ereignissen lernt und so für eine automatische Aktualisierung sorgt. Zusätzlich müssen jedoch benutzerspezifische Anpassungen vorgenommen werden. Es sind daher weitere Arbeiten notwendig, um den BenutzerInnen einfache Werkzeuge zu bieten, mit denen Sie Kontext individuell beschreiben und anpassen können.

Die Skalierbarkeit unseres Modellierungsansatzes konnte bei diesen relativ großen Daten gezeigt werden. Im der aktuellen Version von BSCW wurden die Mechanismen für die Ereignisverarbeitung weiter verbessert und optimiert, sodass auch für wesentlich größere Datenmengen kein Skalierprobleme auftreten würden.

Literaturverzeichnis

- Appelt, W. WWW Based Collaboration with the BSCW System. In SOFSEM'99 (November 26 - December 4, Milovy, Czech Republic). Springer Lecture Notes in Computer Science 1725, 1999. pp. 66-78.
- Dourish, P. and Bellotti, V. Awareness and Coordination in Shared Workspaces. In CSCW '92 - Sharing Perspectives (Toronto, Canada). ACM Press, 1992. pp. 107-114.
- Duden. Das grosse Woerterbuch der deutschen Sprache. Bibliographisches Institut AG, Mannheim, 1978.
- Fitzpatrick, G., Mansfield, T., Kaplan, S., Arnold, D., Phelps, T. and Segall, B. Augmenting the Workaday World with Elvin. In ECSCW'99: Sixth Conference on Computer Supported Cooperative Work (Copenhagen). Kluwer Academic Publishers, 1999. pp. 431-450.

- Fuchs, L. AREA: A Cross-Application Notification Service for Groupware. In Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work - ECSCW'99 (Sept. 12-16, Copenhagen, Denmark). Kluwer Academic Publishers, Dordrecht, NL, 1999. pp. 61-80.
- Fuchs, L., Sohlenkamp, M., Genau, A., Kahler, H., Pfeifer, A. and Wulf, V. Transparenz in kooperativen Prozessen: Der Ereignisdienst in POLITeam. In Deutsche Computer Supported Cooperative Work 1996 - DCSCW'96 (Sept. 30-Oct. 2, Stuttgart-Hohenheim, Germany). Springer-Verlag, Berlin, 1996. pp. 3-16.
- Gross, T. and Prinz, W. Gruppenwahrnehmung im Kontext. In Verteiltes Arbeiten - Arbeit der Zukunft, Tagungsband der Deutschen Computer Supported Cooperative Work Tagung - DCSCW 2000 (Sept. 11-13, Munich, Germany). Teubner, Stuttgart, 2000. pp. 115-126.
- Ishii, H., Kobayashi, M. and Arita, K. Iterative Design of Seamless Collaboration Media. Communications of the ACM 37, 8 (Aug. 1994). pp. 83-97.
- Pankoke-Babatz, U. and Syri, A. Collaborative Workspaces for Time Deferred Electronic Cooperation. In GROUP'97: International ACM SIGGROUP Conference on Supporting Group Work (Phoenix, AZ). ACM Press, 1997. pp. 187-196.
- Patterson, J.F., Day, M. and Kucan, J. Notification Servers for Synchronous Groupware. In Conference on Computer Supported Cooperative Work (CSCW'96) (Boston, MA.). ACM Press, 1996. pp. 122-129.
- Prinz, W. NESSIE: An Awareness Environment for Cooperative Settings. In Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work - ECSCW'99 (Sept. 12-16, Copenhagen, Denmark). Kluwer Academic Publishers, Dordrecht, NL, 1999. pp. 391-410.
- Rittenbruch, M. Atmosphere: Towards Context-Selective Awareness Mechanisms. In Human-Computer Interaction: Communication, Cooperation and Application Design - HCI'1999 (Aug. 22-26, 1999, Munich, Germany). Lawrence Erlbaum, Hillsdale, NJ, 1999.
- Sandor, O., Bogdan, C. and Bowers, J. Aether: An Awareness Engine for CSCW. In Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work - ECSCW'97 (Sept. 7-11, Lancaster, UK). Kluwer Academic Publishers, Dordrecht, NL, 1997. pp. 221-236.
- Schmidt, K. The Problem with Awareness: Introductory Remarks on Awareness in CSCW. Computer Supported Cooperative Work: The Journal of Collaborative Computing (Kluwer Academic Publ., Dordrecht) 11, 3-4 (2002). pp. 285-298.
- Simone, C. and Bandini, S. Compositional Features for Promoting Awareness within and across Cooperative Applications. In Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: The Integration Challenge - Group'97 (Nov. 16-19, Phoenix, AZ). ACM, N.Y., 1997. pp. 358-367.
- Sohlenkamp, M., Prinz, W. and Fuchs, L. POLIAwaC--Design und Evaluierung des POLITeam Awareness-Clients. In Groupware und Organisatorische Innovation, Tagungsband der Deutschen Computer Supported Cooperative Work Tagung - DCSCW'98 (Sept. 28-30, Dortmund, Germany). Springer-Verlag, Berlin, 1998. pp. 181-194.

Kontaktinformationen

Prof. Dr. Tom Gross, tom.gross(at)medien.uni-weimar.de, T. 04363/58-3733

Prof. Dr. Wolfgang Prinz, wolfgang.prinz(at)fit.fraunhofer.de, T. 02241/14-2717