Kontextbezogene Kommunikation

Lars Geiger* Frank Dürr
Universität Stuttgart, Institut für Parallele und Verteilte Systeme
Universitätsstraße 38, 70569 Stuttgart
{lars.geiger, frank.duerr}@ipvs.uni-stuttgart.de

Zusammenfassung

Verbindet man die Position eines Benutzers mit weiteren Attributen wie beispielsweise Typ, Alter oder Status zu seinem Kontext, so ergeben sich daraus interessante neue Anwendungsmöglichkeiten. In diesem Artikel soll ein Verfahren vorgestellt werden, wie ein solcher Kontext benutzt werden kann, um Nachrichten an bestimmte Gruppen von Benutzern — bestimmt durch ihren Kontext — zu versenden. Ein solcher Kommunikationsmechanismus könnte beispielsweise eingesetzt werden, um Verkehrsnachrichten gezielt an alle Fahrzeuge innerhalb eines bestimmten Gebietes und mit einer bestimmten Fahrtrichtung zu übermitteln.

1 Einführung

Tragbare Navigationsgeräte, die den Benutzer mit zusätzlichen Informationen wie z.B. den nächstgelegenen Sehenswürdigkeiten oder Restaurants unterstützen, finden eine immer größere Verbreitung. Sie sind ein einfaches Beispiel für mobile kontextbezogene Anwendungen. Der DFG Sonderforschungsbereich 627 „Umgebungsmodelle für mobile kontextbezogene Systeme“ (Nexus) beschäftigt sich mit der Entwicklung globaler dynamischer Umgebungsmodelle. Solche Modelle dienen als Basis für verschiedenste kontextbezogene Anwendungen.

Im Rahmen dieser Forschung werden auch kontextbezogene Kommunikationsmechanismen betrachtet. Dabei geht es um die Untersuchung von Verfahren, Nachrichten an bestimmte Teilnehmer (die durch beliebige Attribute bestimmt werden) innerhalb eines bestimmten geografischen Bereichs zu versenden. Ein Beispiel für einen solchen Contextcast ist das Versenden von Verkehrsnachrichten: Fahrzeuge werden über das Attribut *type* = „vehicle“ identifiziert — Verkehrsmeldungen sind für Fußgänger eher weniger von Bedeutung. Gleichzeit schränkt eine Lokation den Empfangsbereich der Nachrichten auf die betroffenen Orte ein: ein Stau in Hamburg ist für einen Autofahrer in Stuttgart mit hoher Wahrscheinlichkeit uninteressant.

*Der Autor dankt der Deutschen Forschungsgemeinschaft (DFG) für die Unterstützung im Rahmen des Sonderforschungsbereichs 627 „Umgebungsmodelle für mobile kontextbezogene Systeme“

Da zu einer gegebenen Nachricht eine beliebige Anzahl Empfänger mit einem passenden Kontext existieren kann, handelt es sich hierbei um eine Multicast- bzw. 1:m-Kommunikation. Entsprechend ähneln die auftretenden Schwierigkeiten sehr denen anderer Multicast-Verfahren: Nachrichten müssen an alle Empfänger übermittelt werden, und zwar möglichst optimal hinsichtlich einer gegebenen Metrik (z.B. Verzögerung der einzelnen Pakete, Anzahl an Nachrichtenduplikaten, etc.). Ein Contextcast unterscheidet sich jedoch durch die flexiblere Adressierung und Gruppendefinition von anderen Verfahren wie IP-Multicast. Aus diesem Grund sind für eine Realisierung neue Konzepte notwendig.

2 Verwandte Arbeiten

Das vorgestellte Konzept eines Contextcasts ist eng verwandt sowohl mit Publish/Subscribe durch die Attribute der Empfänger, als auch mit Geocast durch die Ziel-Lokation einer Nachricht. Bei inhaltsbezogenem Publish/Subscribe können Teilnehmer anhand einer Reihe von Attributen die für sie relevanten Nachrichten selektiv empfangen bzw. andere Nachrichten vom System aussortieren lassen. Ein Netz von Brokern übermittelt die Benachrichtigungen von den Sendern (*publisher*) zu den interessierten Empfängern (*subscriber*). Beispiele für solche Systeme sind JEDI [4], REBECA [8], SIENA [3] oder auch Gryphon [1]. Speziell in den letzten Jahren wurden Methoden betrachtet, um solche Systeme für mobile Teilnehmer zu erweitern (z.B. für REBECA [5] oder SIENA [2]). Allerdings berücksichtigen diese Erweiterungen nur die Mobilität der Teilnehmer und die daraus resultierende wechselnde Konnektivität. Sie erlauben es jedoch nicht, die Lokation von Benutzern zur Übermittlung von Benachrichtigungen zu verwenden.

Auf der anderen Seite ermöglichen Geocast Systeme, wie beispielsweise das auf einem geometrischen Lokationsmodell basierende GEO [7], die Adressierung von Teilnehmern anhand ihrer Position. Allerdings unterstützen existierende geografische Routingverfahren keine weiteren Attribute, mit der die Menge der Empfänger innerhalb der Ziel-Lokation weiter eingeschränkt werden kann.

Daneben existiert bereits ein weiteres Kommunikationssystem, das die Bezeichnung „ContextCast“ trägt [6]. Allerdings unterstützt dieses System nur die Adressierung und Übermittlung von Nachrichten in einem globalen Kontextraum $G \subset \mathbb{Z}^n$. Dieser stellt ein n-dimensionales Koordinatensystem dar und ist somit eher eine Variante eines Geocast Systems. Zusätzliche Attribute sind nicht vorgesehen bzw. müssten durch eine Abbildung der Attribute in einzelne Dimensionen von G nachgebildet werden.

3 Kontextbezogenes Routing

Durch die enge Verwandtschaft zu Publish/Subscribe Systemen liegt beim Entwurf die Verwendung entsprechender Verfahren aus diesem Bereich nahe. Gleichzeitig unterscheiden sich Lokationen in ihrer Semantik aber deutlich von den üblichen Attributen in inhaltsbezogenen Publish/Subscribe Systemen. Dies wirft eine Reihe neuer Fragen und Probleme auf. Im folgenden soll ein erster Entwurf vorgestellt werden. Da eine Veränderung der IP-Infrastruktur und eine Erweiterung um zusätzliche Routing-

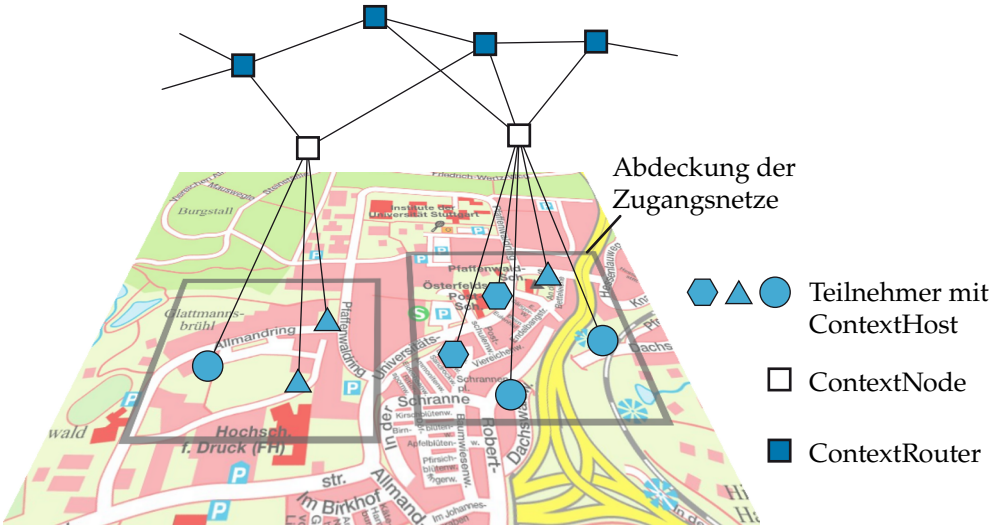

Abbildung 1: Schematische Darstellung des Contextcast Systems

Verfahren aus technischen und wirtschaftlichen Gründen eher schwer durchzuführen ist, wurde für den Entwurf des Contextcast Systems ein Overlay Netz konzipiert, das die Nachrichten auf Anwendungsebene zu den Empfängern übermittelt.

Abbildung 1 zeigt eine schematische Darstellung des Entwurfs: die so genannten ContextRouter leiten Nachrichten entlang der etablierten Pfade zu den ContextNode genannten Zugangsknoten weiter. Die Aufgabe der ContextNodes ist die Verwaltung der angemeldeten Teilnehmer und ihrer Kontexte, sowie die Übermittlung von Nachrichten innerhalb der Zugangsnetze. Jedem ContextNode ist dazu eine Lokation zugewiesen, die er verwaltet. Diese Lokation ändert sich — verglichen mit den Attributen der Teilnehmer — eher selten. Auf den teilnehmenden Systemen wird ein ContextHost genannter Prozess ausgeführt. Er stellt die Schnittstelle zu den Anwendungen bereit und verwaltet z.B. die aktuelle exakte Position und die Attribute des Teilnehmers. Außerdem filtert er die Nachrichten abschließend, bevor sie an die Anwendungen übergeben werden. Damit wird verhindert, dass überflüssig übermittelte Nachrichten an Anwendungen ausgeliefert werden.

Damit die ContextRouter eine Nachricht weiterleiten können, benötigen sie die Information, welche Empfänger bzw. Objekte mit welchen Kontexten (Lokation und Attribute) über eine bestimmte Verbindung erreichbar sind. Die am Routing beteiligten Knoten tauschen dazu periodisch über ein Routingprotokoll Informationen aus. Dabei erhalten alle Router ein globales Wissen, insbesondere ist bekannt, für welche Lokation(en) ein ContextNode zuständig ist. Jeder Router kann daraus den kürzesten Pfad zu allen ContextNodes und damit auch zu allen Lokationen berechnen. Außerdem müssen auch die Attribute, die den Kontext der Teilnehmer ausmachen, den Routern bekannt sein. Dazu propagiert jeder ContextNode die Attributmengen der Kontexte seiner Teilnehmer durch das gesamte Netz. Jeder Router kann durch die bereits bekannte Information über die kürzesten Pfade überprüfen, ob Attribute über die kürzeste Route zum Sender empfangen wurde. Ist das der Fall, wird für die Attribute und die Verbindung ein zugehöriger Forwarding-Eintrag angelegt.

Empfängt ein Router nun eine neue Contextcast Nachricht, so kann er für jede seiner Verbindungen überprüfen, ob darüber ein Teil der Ziel-Lokation sowie Teilnehmer mit passenden Attributen erreicht werden können. Über alle Verbindungen, auf die das zutrifft, leitet er die Nachricht an den jeweils nächsten Hop weiter. Dadurch wird jede Nachricht entlang eines Multicast Baums aus kürzesten Pfaden bis zu den zuständigen ContextNodes weitergeleitet.

Da prinzipiell jeder Knoten auch Sender sein kann, ist es notwendig, dass die Empfänger ihre Kontexte zu allen anderen Knoten propagieren und dadurch entsprechende Routen etablieren. Dadurch wird jedoch auch eine vergleichsweise hohe Netzlast erzeugt. Aus diesem Grund trennt der Entwurf die Benutzerkontexte in die Lokation, die über die Zuständigkeit der ContextNodes im Netz (statisch) approximiert wird, und die (dynamischen) Attribute der Benutzer. Würden Benutzerlokation und seine Attribute dagegen integriert behandelt, so würde die Feststellung gleicher oder enthaltener Kontexte ($K_1 \subseteq K_2$) sehr erschwert, da praktisch jede Benutzerlokation unterschiedlich ist. Durch diesen Ansatz wird es möglich, die Anzahl der Attribute durch Techniken aus Publish/Subscribe Systemen (wie *cover* und *merge*, s. Abschnitt 5) zu reduzieren. Andererseits führt diese Entscheidung aber auch eine Unschärfe ein, da nicht mehr allen Routern exakt bekannt ist, welche Benutzerattribute in welchen Gebieten existieren.

4 Ergebnisse

Der vorgestellte Entwurf wurde zur Bewertung als Prototyp implementiert. Zum Vergleich wurde ein zweiter Prototyp geschrieben, der die Lokation sowie weitere Kontextattribute integriert betrachtet und gemeinsam durch das Netz propagiert.

Erste Ergebnisse zeigen, dass durch die entstehende Unschärfe beim ersten Prototypen die Anzahl der Multicast-Nachrichtenkopien um rund 50% ansteigt, wobei dieser Nachteil mit steigender Teilnehmerzahl geringer wird. Den größten Anteil an der Netzlast machen in den Simulationen jedoch die Nachrichten aus, mit denen die Kontexte propagiert und Routen etabliert werden. Der Vorteil des Entwurfs ist deshalb, dass zum einen die Kontexte keine Lokation mehr enthalten und zum anderen die Menge der Attribute der Kontexte einfacher mittels existierender Verfahren aus Publish/Subscribe Systemen verringert werden kann (s. Abschnitt 5). Die Auswirkung dieser Verbesserung muss allerdings erst noch in zukünftigen Simulationen untersucht werden.

5 Zusammenfassung und Ausblick

Der vorgestellte Kommunikationsmechanismus ermöglicht eine effiziente Nachrichtenübermittlung anhand von Empfängerkontexten entlang eines Baumes aus kürzesten Pfaden. Allerdings zeigte sich bei der Evaluierung des Prototypen auch, dass die per Flooding übermittelten Kontexte das Netz nicht unerheblich belasten. In zukünftigen Arbeiten ist deshalb das Ziel, diese Netzlast zu verringern und das System entsprechend besser skalierbar zu gestalten. Dazu kommen verschiedene Ansätze in Frage: Durch ein zweistufiges hierarchisches Routing könnte das Propagieren der Kontexte auf kleinere, lokale Systeme beschränkt werden. Für die obere Ebene bietet sich dabei intuitiv ein

Routing anhand der Ziel-Lokation(en) an, eventuell können auch weitere Attribute für das übergeordnete Routing identifiziert und verwendet werden.

Eine weitere Möglichkeit besteht in der Reduktion der Kontexte, die übermittelt werden müssen. Für Publish/Subscribe Systeme sind zu diesem Zweck die Verfahren *cover* und *merge* bereits gut erforscht. Eine entsprechende Methode kann dazu verwendet werden, von mehreren Kontexten nur den allgemeinsten an die anderen Router zu propagieren. Dies ist auch der Grund für die Entscheidung, die Lokation von den restlichen Attributen getrennt im Rahmen des Routingprotokolls an die ContextRouter zu übermitteln: ein Kontext K_2 kann nur dann allgemeiner sein als ein Kontext K_1, wenn für die Lokationen der beiden Teilnehmer gilt: $L_1 \subseteq L_2$.

Eine weitere Idee, die ebenfalls aus dem Gebiet der Publish/Subscribe Systeme stammt, ist die Verwendung sogenannter *Advertisments*. Bisher wird durch einen Broadcast ein Pfad zwischen jedem Empfänger und jedem potenziellen Sender etabliert. Falls jedoch nur ein kleiner Teil der Teilnehmer tatsächlich Nachrichten senden möchte, so könnte durch *Advertisements* bekannt gemacht werden, wie diese Nachrichten adressiert sind. Entsprechend müssten die Kontexte der Teilnehmer anschließend nur noch zu den Sendern, zu deren Nachrichten sie passen, übermittelt und auf dem Weg eine entsprechende Route etabliert werden. Besonders, wenn die Zahl der Sender entsprechend klein ist, wird so das Problem des Floodings weg von den zahlreichen Empfängern zu einigen wenigen Sendern verschoben und die Netzlast so weiter reduziert.

Literatur

[1] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems. In *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (ICDCS'99)*, pages 262–272, 1999.

[2] M. Caporuscio, P. Inverardi, and P. Pelliccione. Formal Analysis of Clients Mobility in the Siena Publish/Subscribe Middleware, mar 2002.

[3] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Interfaces and Algorithms for a Wide-Area Event Notification Service, jul 2000.

[4] G. Cugola, E. D. Nitto, and A. Fuggetta. The JEDI Event-Based Infrastructure and Its Application to the Development of the OPSS WFMS. *IEEE Trans. Softw. Eng.*, 27(9):827–850, 2001.

[5] L. Fiege, F. C. Gärtner, O. Kasten, and A. Zeidler. Supporting Mobility in Content-Based Publish/Subscribe Middleware. In M. Endler and D. C. Schmidt, editors, *Middleware*, volume 2672 of *Lecture Notes in Computer Science*, pages 103–122. Springer, 2003.

[6] D. Heutelbeck. Context Spaces — Self-Structuring Distributed Networks for Contextual Messaging and Resource Discovery. *Lecture Notes in Computer Science*, 2519:248–265, 2002.

[7] T. Imielinski and J. C. Navas. GPS-based geographic addressing, routing, and resource discovery. *Commun. ACM*, 42(4):86–92, 1999.

[8] G. Mühl. *Large-Scale Content-Based Publish/Subscribe Systems*. PhD thesis, TU Darmstadt, 2002.