
ContextFramework.KOM – Eine offene Middleware zur Integration heterogener Sensoren in eine Kontext-sensitive Kommunikationsplattform

Johannes Schmitt, Matthias Kropff, Andreas Reinhardt, Matthias Hollick

Fachgebiet Multimedia Kommunikation

Technische Universität Darmstadt, Merckstr. 25, 64283 Darmstadt

Die heutigen multimedialen Kommunikationstechnologien ermöglichen es, immer und überall über verschiedene Kommunikationskanäle erreichbar zu sein. Gerade für den gesellschaftlich relevanten Informations- und Wissensarbeiter wurden durch diese Technologien umfangreiche Effizienzsteigerungen prognostiziert. Um sich an die gestiegenen Transaktionsgeschwindigkeiten der aktuellen Informationsgesellschaft anzupassen, bietet daher der Arbeitsplatz eines moderner Informations- und Wissensarbeiter umfangreiche Technologien der *Computer-vermittelten Kommunikation* (CvK) wie Telefon-, Email-, Instant Messaging- oder Videotelefonie-Systeme.

Jüngste Untersuchungen über Technikstress am Arbeitsplatz zeigen jedoch, dass das erhöhte Kommunikations- und Konfigurationsvolumen sich stark negativ auf die Arbeitsproduktivität der Empfänger von Kommunikationsanfragen auswirken kann. Im Rahmen eines gemeinsamen Forschungsprojektes zwischen Universität, kleinem Unternehmen und großem Industriepartner zielen wir darauf ab, einen *Virtuellen Assistenten* zu entwickeln, der abhängig vom Kontext seiner Zielperson automatisiert die Filterung, Priorisierung und gegebenenfalls autonome Beantwortung von Kommunikationsanfragen ermöglicht. Um die Funktionalitäten des vorgeschlagenen Virtuellen Assistenten zu realisieren, ist das System auf umfangreiche Kontext-Informationen des Benutzers angewiesen. Informationen von rein statischen Sensoren, wie etwa Datenbank-Einträgen, bis hin zu hoch dynamischen Sensordaten, werden von unterschiedlichsten Endgeräten, wie z.B. dem Desktop-PC des Benutzers, seinem Mobiltelefon oder von integrierten Sensoren in seiner Arbeitsumgebung bereitgestellt. Da die Art und Anzahl der Sensoren äußerst heterogen ist, wurde eine Middleware entworfen und eingesetzt, die die verschiedensten Informationsquellen integriert und eine verlässliche Kommunikation bereitstellt. Die in einer Gruppenarbeit entwickelte Softwarearchitektur ermöglicht, dass Sensoren sich und ihre Fähigkeiten semantisch beschreiben und somit auch bislang unbekannte Sensoren nahtlos ins System eingebunden werden können.

Funktion der Software

Das ContextFramework.KOM besteht im Kern aus einer generischen, offenen Middleware, die es erlaubt, Sensoren beliebiger Art zur Laufzeit dynamisch anzubinden, um so automatisiert auf die verfügbaren Sensorinformationen zugreifen zu können. Eine ausgeklügelte Auswertelogik schließt aus den verfügbaren Informationen auf den aktuellen Kontext der im System registrierten Benutzer und kann dieses Wissen gezielt zur Unterstützung von Kommunikationsdiensten bereitstellen. Konkret wird dies realisiert, indem aus einem autonom erlernten Modell, das sowohl Sensorwerte als auch persönliche Präferenzen der Nutzer berücksichtigt, verschiedene höherwertige Informationen generiert werden, die sowohl zu Verbesserungen in der Telefonie als auch in der Kommunikation allgemein herangezogen werden können. Im exemplarisch betrachteten Fall der Kontext-bewussten Telefonie können diese Informationen etwa dahingehend eingesetzt werden, dass etwa bei längeren Beschäftigungen des Nutzers, die keine Unterbrechung erlauben, Telefone automatisch stumm geschaltet werden und Anrufenden eine entsprechende Mitteilung ausgegeben wird. Je nach Relation des Anrufers zum Angerufenen kann ebenfalls eine Meldung über die aktuelle Beschäftigung und den Zeitraum der erwarteten Wieder-Erreichbarkeit des Teilnehmers ausgegeben werden. Durch die Erfassung aller Kommunikationsmöglichkeiten eines Teilnehmers und der Berücksichtigung der im jeweiligen Moment verfügbaren Untermenge, wird eine erhöhte Zielführung von Anrufversuchen erreicht, sowohl durch Auswahl des aktuell am besten geeigneten Kommunikationsmediums, und ebenfalls durch die Unterstützung in der Findung eines idealen Anrufzeitpunkts.

Im Gegensatz zu existierenden Lösungen, bei denen die Konfiguration von Einstellungen am Telefon, wie z.B. Weiterleitung oder Einspielung einer entsprechenden Ansage, manuell oder durch komplexe Regelwerke vorgenommen werden muss, handelt ContextFramework.KOM autonom und adaptiv, und berücksichtigt hierbei verfügbare Sensorwerte zur Erstellung eines Benutzerprofils und der deduktiven Ableitung von Regeln. Insgesamt zeichnet sich das System dadurch aus, dass es die Funktionalität eines *virtuellen Assistenten* realisiert, also eine Vielzahl von Tätigkeiten automatisiert, die ein Assistent heute im Bereich der Kommunikation erfüllt.

Praxisrelevanz

Heutige Unified Messaging Plattformen bieten neben der Integration verschiedener Kommunikationsmedien eine Vielzahl von Funktionalitäten zur vermeintlichen Vereinfachung des Kommunikationsmanagements. Es zeigt sich jedoch, dass aufgrund der hohen Komplexität und des Verwaltungsaufwands, Mehrwertdienste wie das sogenannte Call-Forwarding nur sehr eingeschränkt verwendet werden. Da ein Wissensarbeiter meist über verschiedene Kommunikationsendgeräte verfügt, beispielsweise Desktop-PC, Desktop-Telefon, Mobiltelefon und Notebook, resultiert hieraus ein sehr hoher manueller Konfigurations- und Verwaltungsaufwand. Jedes dieser Endgeräte kann in Kombination mit der verfügbaren Sensorik in

der Umgebung des Nutzers wertvolle Informationen zur Ermittlung dessen Kontext, z.B. Präsenz-Status, bereitstellen. In der Praxis ist dies jedoch aufgrund fehlender einheitlichen Schnittstellen nur unzureichend möglich. In einem ersten Schritt ermöglicht die hier vorgestellte Middleware die Kommunikation zwischen unterschiedlichsten Endgeräten. Hierdurch wird es in einem zweiten Schritt möglich, die heterogenen Informationsquellen zentral zu erfassen um Aufschluss über den Nutzer-Kontext zu erhalten. Dies ermöglicht in einem dritten Schritt eine automatisierte Steuerung und Parametrisierung von Kommunikationsdiensten, um den Verwaltungsaufwand zu minimieren. Zur Adaption des Systems an das Nutzerverhalten werden intelligente Lernverfahren angewendet, die die gewünschten Kommunikationsregeln erstellen. Hierzu wird ein nur minimales Feedback des Nutzers benötigt, das jedoch aufgrund der durch die Middleware entstandenen Vernetzung komfortabel von jedem Endgerät aus gegeben werden kann. Beispielsweise führt die mehrfache Wahl der Regel „Gesprächsweiterleitung auf Mailbox“ in einem gegebenen Kontext (Identität des Anrufers, Zeit, Ort, Beschäftigung des Angerufenen) dazu, dass das Kommunikationssystem bei nochmaligem Anruf sofort den Anrufer auf die Mailbox weiterleitet, ohne den Nutzer in seiner Tätigkeit zu unterbrechen. Hierdurch wird eine möglichst einfache und intuitive Nutzerführung realisiert.

Neuigkeit des Systems

Das System kombiniert eine Reihe von Technologien aus unterschiedlichen Bereichen zu einem Gesamtsystem mit vielfältigen Funktionen und Möglichkeiten. So werden im System unter anderem Ansätze und Technologien aus dem Bereich der Service-orientierten Architekturen umgesetzt und eingebunden. Die grundlegende Architektur des Systems ist Dienst-orientiert. Hierzu wurden die einzelnen Dienste im Rahmen des leichtgewichtigen Concierge OSGi Frameworks¹ implementiert. Die verteilt verfügbaren Informationsquellen wurden ebenfalls in Form von Diensten auf den Endgeräten umgesetzt und auf die jeweilige Plattform angepasst. Zur Auffindung dieser *Sensor-Dienste* wurden Technologien aus dem Bereich der *Service Lookup*-Protokolle herangezogen. Zur Kommunikation mit Sensor-Diensten, über unterschiedliche Plattformen und Netzwerke-Technologien hinweg kommen Technologien aus dem Bereich Middleware zum Einsatz. Hierzu wurde ein System aufgebaut, das Konnektoren einsetzt, die eine gemeinsame Schnittstelle zur Kommunikation auf der Ebene der Dienste bieten, auf der darunter liegenden Ebene jedoch eine auf das zugrunde liegende System angepasste Kommunikation ermöglichen. Es wurden hierbei unter anderem Konnektoren auf der Basis von *RMI*, *XML-RPC*, *Remote-OSGi*² umgesetzt.

Das Systems ist offen gegenüber neuen Sensoren und Technologien. So können im Nachhinein auch neue, unbekannte Sensoren am System angemeldet und genutzt werden. Sensoren und deren benötigte bzw. angebotene Information werden hierbei geeignet beschrieben, um neben der Relevanz eines Sensors für bestimmte Suchanfragen auch eine korrekte Verarbeitung und Nutzung der Informationen zu gewährleisten. Zur Beschreibung kommen sowohl Technologien aus dem Bereich der Schnittstellenbeschreibung von Diensten als auch Methoden zur semantischen Beschreibung von Diensten und Informationen zum Einsatz³.

Zur Bestimmung der Relevanz einer eventuell unbekanntem Informationsquelle zu einer Suchanfrage werden Aussagen über Beziehungen und Ähnlichkeiten zwischen Informationsquellen beziehungsweise Datentypen benötigt. Beziehungen zwischen Objekten werden im Allgemeinen durch *Ontologien* definiert. In vielen existierenden System wird eine fest vordefinierte Ontologie verwendet, die keine Erweiterungen zulässt. Um zur Erfüllung der Anforderung auch nachträglich unbekannte Informationsquellen aufzunehmen, wurden in *ContextFramework.KOM* Mechanismen zur Anpassung von Ontologien (über zusätzliche Elemente innerhalb der semantischen Beschreibung der Sensor-Dienste) integriert.

Die Parameter einer Suche über die Informationsquellen können je nach Anwendungsfall variieren. Eine Suche beinhaltet, neben den initialen Informationen (z.B. Name oder Nummer eines Teilnehmers), welche als Startpunkte für die Suche genutzt werden, auch die angestrebten Zielpunkte der Suche. Zielpunkte sind diejenigen Arten Informationen, welche für eine Entscheidung innerhalb des Anwendungsfalls benötigt werden (z.B. Aktivität oder Aufenthaltsort eines Teilnehmers). Die Suche erfolgt iterativ. Je nach gefundenen Informationen, können neue Informationsquellen relevant werden. So können beispielsweise über den ermittelten Aufenthaltsort auch die Informationsquellen, welche sich an diesem Ort befinden relevant werden. Neben der Möglichkeit die Zielpunkte eine Suche zu definieren, können auch weitere Parameter im Bereich *Dienstgüte* (QoS) angegeben werden. Es können beispielsweise neben der Vorgabe einer maximalen Suchdauer auch Anforderungen über die erforderliche *Informationsqualität* (QoI), wie etwa die Aktualität der Ergebnisse, übergeben werden. Die Suche ermöglicht die Definition von *Interessen*, also Möglichkeiten, bereits vor der eigentlichen Suchanfrage, die Such-Parameter zu spezifizieren. Ein *Interesse* wird dazu genutzt, vorab sicher zu stellen, dass die QoS-Anforderungen einer Suche erfüllt werden können. Hierzu werden *Prefetching*-Mechanismen innerhalb des Caches entsprechend eingestellt, um kritische Informationsquellen vorab anzufragen und die Daten mit ausreichender QoI vorzuhalten.

¹ <http://conciierge.sourceforge.net/>

² <http://r-osgi.sourceforge.net/>

³ basierend auf OWL-S - <http://www.w3.org/Submission/OWL-S/>

Zur Auswertung der gesammelten Informationen werden Verfahren aus dem Bereich maschineller Lernverfahren eingesetzt. Um zu vermeiden, dass ein Nutzer die Regeln zur Auswertung der Informationen selbst definieren muss, wird ein adaptives, lernfähiges Auswerteverfahren genutzt. Dieses Verfahren stützt sich auf das *Feedback* des Nutzers. Mittels Feedback bestimmt der Nutzer das gewünschte Resultat zum Zeitpunkt X. Das System ermittelt daraufhin mögliche Abhängigkeiten und Muster zwischen dem gewünschten Resultat und den gesammelten Informationen zum Zeitpunkt X. Auf diesen Mustern aufbauend wird das Auswertemodell angepasst. Das System ermöglicht die Nutzung und den direkten Vergleich verschiedener existierender Lernverfahren. Hinzu kommt die Umsetzung eines eigenen, speziell auf die Anforderungen innerhalb des Telefonie-Szenarios angepassten, Lernverfahrens.

Das System bietet die Möglichkeit neue Informationsquellen mit geringem Aufwand zu integrieren. Ein minimal konzipierter Installer ermittelt die zugrunde liegende Plattform und lädt entsprechende Sensor-Dienste und Konfigurationsdateien vom einem Repository nach. Das System beinhaltet einen eigenen leichtgewichtigen Webserver, welcher den Basis-Funktionen von *Tomcat* nachempfunden wurde. Über diesen Webserver kann jeder Dienst eingesehen, genutzt und konfiguriert werden.

Das System kann in mehreren Modi ausgeführt werden. Neben der Möglichkeit die Dienste innerhalb eines OSGi-Frameworks zu starten, kann das System auch direkt, ohne OSGi, genutzt werden. Ein eigener Scheduler, welcher durchgängig im System genutzt wurde, ermöglicht es das System auch in simulierter Zeit und Umgebung zu starten. Hierbei kann ebenfalls über einen speziellen Konnektor das Verhalten der Informationsquellen zur Realzeit aufgenommen und zur simulierten Zeit wiedergegeben werden.

Die Nutzung des Systems wurde exemplarisch für das Anrufmanagement innerhalb eines VoIP-Systems umgesetzt, es kann jedoch auch für andere Zwecke der Kontext-bewussten Kommunikation genutzt werden. Das System erbringt seine Dienste weitgehend autonom. Neben der vereinfachten Prozedur zur Installation der Sensor-Dienste, braucht ein Teilnehmer nur noch Feedback zu geben, um das Verhalten des Systems an seine Wünsche anzupassen. Um auch diesen Schritt zu vereinfachen, existieren verschiedene Möglichkeiten des Nutzerfeedbacks: Web-basiertes Feedback, Feedback per Telefoneingabe und Symbolleiste am PC, aber z.B. auch die Möglichkeit, Feedback über eine Bluetooth-fähige Armbanduhr zu geben. Zur Visualisierung der vorhandenen Informationsquellen innerhalb einer bestimmten Lokation, wurde ein weiterer Dienst entwickelt, der diese Daten auf Google Earth zur Verfügung stellt.

Systemvoraussetzungen

Um Umrüstkosten an Rechnern und mobilen Endgeräten zu vermeiden, wurde bei der Entwicklung des Systems konsequent Wert auf die Einsetzbarkeit bestehender Geräte und Technologien gelegt. Insbesondere wurde durch eine einfache Installationsroutine dafür Sorge getragen, dass der administrative Aufwand und damit die Betriebskosten gering bleiben.

Zur Erfassung der Informationen, die herangezogen werden, um beispielsweise Schlüsse auf den aktuellen Kontext einer Person ziehen zu können, dienen primär existierende Endgeräte mit Zugang ins Internet, u.a. Mobiltelefone, Desktop-Rechner, oder auch diverse Linux-basierte eingebettete Systeme. Neben dem Einsatz effizienter Java-basierter Kommunikation lassen sich hierbei auch textuelle Übertragungsprotokolle wie etwa *XML-RPC* einsetzen. Somit können neben Plattformen, die virtuelle Maschinen für Java zur Verfügung stellen, auch mobile Endgeräte wie etwa Windows Mobile-Telefone eingesetzt werden.

Der Such-, Aggregations- und Auswertedienst kann zentral auf einem einzelnen Sensor-Verzeichnis-Server initiiert werden, alternativ aber auch in Form mehrerer Verzeichnisse umgesetzt werden. In beiden Fällen jedoch ist ein dedizierter Server mit erhöhter Rechenleistung nötig, um Suchanfragen und Auswerte-Algorithmen zeitnah ausführen zu können. Neben der Anzahl der angemeldeten Sensoren hat hierbei ebenfalls die Komplexität der Auswerte-Logik einen Einfluss.

Verfügbarkeit und Zusammenfassung

Das System wurde mit allen zuvor beschriebenen Funktionalitäten und Eigenschaften umgesetzt und wird stetig um weitere Funktionen erweitert. Als Proof-of-Concept sind zudem eine Reihe von Sensor-Diensten implementiert worden und stehen für Tests und Demonstrationen zur Verfügung. Hierzu zählen unter anderem Sensor-Dienste zur Nutzung von Adressbuch und Kalendereinträgen, Daten über die Bluetooth- und WLAN-Umgebung, Tastatur- und Maus-Aktivität, gestartete Anwendungen, sowie Erkennung von Bewegung und Aktivitätsbereichen durch eine Webcam. Zudem steht ein Gateway zur transparenten Anbindung drahtloser Sensornetze zur Verfügung. Im Rahmen dieser Sensornetze sind weitere Sensoren zur Lokationsbestimmung, Bewegungserkennung und Bestimmung von Nachbarschaftsbeziehungen umgesetzt worden.

Die Demonstration des Systems beinhaltet die Nutzung und Auswertung dieser Informationsquellen zur Steuerung des Anrufmanagements innerhalb eines VoIP-basierten Kommunikations-Szenarios.