

InfoSpace-Servicearchitektur eine Middleware für Ad hoc-Netzwerke

J. Brehm, C. Müller-Schloer
SRA, Universität Hannover
Appelstrasse 4,
30167 Hannover, Germany
Email: {brehm, cms}@sra.uni-hannover.de
Tel.: +49-511-76219730, Fax.: +49-511-76219733

R. Welge
Fachbereich Automatisierungstechnik, Teilbereich Informatik
Fachhochschule Nordostniedersachsen
Volgershall 1,
21339 Lüneburg
Email: welge@fhnon.de
Tel.: +49-4131-677462, Fax.: +49-4131-677300

1 Einführung

Moderne Infrastruktur-basierte Netzwerke weisen neben stationär angebundenen Netzwerkknoten zunehmend mobile Geräte auf, die über Funk-gestützte Technologien angebunden werden. Mit dem heute technisch ausgereiften drahtlosen Kommunikationsstandard WLAN, IEEE.802.11b sind komfortabel handzuhabende Datenkanäle verfügbar, die sich insbesondere für die Netzwerkintegration von mobilen Geräten mit hohen Verarbeitungskapazitäten wie Notebooks und PDAs (Personal Digital Assistant) anbieten. Ein wesentliches Charakteristikum von Infrastruktur-basierten Kommunikationskanälen besteht darin, dass deren Endpunkte in der Regel aus jeweils einem Zugangspunkt und einem mobilen Gerät bestehen.

Ein weiterer Trend im Bereich der drahtlosen Kommunikation ist in der Entstehung von sogenannten Ad hoc-Netzwerktechnologien zu erkennen. Hier geht man davon aus, dass jeweils zwei mobile Geräte mit einer begrenzten Reichweite bei einem Zusammentreffen eine Punkt-zu-Punkt-Verbindung (im folgenden als Peer-to-Peer-Verbindung bezeichnet) aufbauen. Dabei ist kein zentraler Zugangspunkt notwendig. Das hat zur Folge, dass dedizierte Rollen wie Client und Server möglicherweise noch für einzelne Applikationen, nicht aber mehr für die Geräte in ganzheitlicher Form gelten müssen. Stattdessen bietet jedes Gerät seine Ressourcen als Dienste innerhalb des entstandenen Ad hoc-Netzwerkes an, bzw. konsumiert Dienste anderer Geräte. Damit entsteht eine dynamische Struktur von Dienst Anbietern und Dienstabnehmern. Geht man davon aus, dass alle beteiligten Gerät zum einen mehrere Peer-to-Peer-Verbindungen aufbauen können und zum anderen nur eine begrenzte Zeit dem Ad hoc-Netzwerk angehören, so bedeutet ein erfolgreiches Netzwerkmanagement die Bewältigung von Problemen, die in Infrastruktur-basierten Netzwerken weniger eine Rolle spielen. Das sicherlich nicht neue Problem der vorher nicht kalkulierbaren erforderlichen Bandbreite der Kommunikationskanäle wird durch die Tatsache verschärft, dass deren Anzahl sich durch hinzukommende oder abgehende mobile Geräten zu jeder Zeit beliebig verändern kann. Darüber hinaus sind sowohl Dienstanbieter als auch Dienstkonsumenten von Natur aus unzuverlässig, was in Infrastruktur-basierten, insbesondere stationären Netzwerken eher als Fehlerfall einzustufen ist. Der heute verfügbare Mobilkommunikationsstandard Bluetooth [1, 2, 3] stellt eine Basis für die Realisierung von Ad hoc-Netzwerktechnologien dar.

Vor dem Hintergrund der in der Universalrechnertechnik verfügbaren Funk-gestützten Kommunikationskanäle ist derzeit der Trend im Embedded Systems-Bereich zu beobachten, oben genannte Technologien auch hier einzuführen. Dieser Trend wird insbesondere dadurch interessant, dass z.B. Bluetooth die Konvergenz von Daten-, Audio/Video-, und Steuerungsanwendungen auf Protokollebene unterstützt.

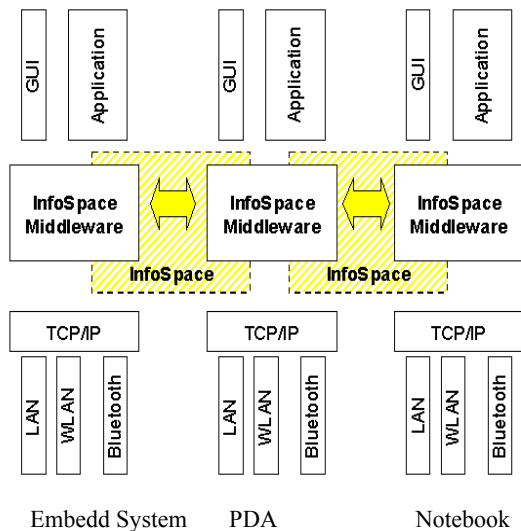


Bild 1: Einordnung der InfoSpace Servicearchitektur

Eben dieser Konvergenzaspekt aber auch die derzeit entstehenden und noch nicht gelösten Probleme hinsichtlich des sicheren Managements von per definitionem „unsicheren“ Netzwerktechnologien, den Ad hoc-Netzwerken, haben uns dazu motiviert, die plattformübergreifende *InfoSpace-Servicearchitektur*, eine sogenannte Middleware, zu entwickeln. Eine Middleware ist eine Softwarekomponente, die während der Ausführung verteilter Systeme zwischen Netzwerktechnologien wie TCP/IP und beliebigen Anwendungen angeordnet ist. Bild 1 zeigt die Einordnung der Middleware. Die *InfoSpace-Servicearchitektur* kapselt Protokollstacks durch übergeordnete Dienste, die für Aufgaben in verteilten Systemen unbedingt notwendig sind. Wir verfolgen mit unserem

Projekt das Ziel, Embedded Systems-Plattformen auf der einen Seite und PDAs sowie Notebooks auf der anderen Seite auf Middlewareebene konvergieren zu lassen.

2 Unsere Vorstellung von zukünftigen Netzwerken: Ubiquitous Computing

Ein neues, hier verwendetes Netzwerkkonzept wird mit dem Begriff „Ubiquitous Computing“ umschrieben. Die wesentlichen Grundgedanken lassen sich wie folgt zusammenfassen:

„Ubiquity“: Der Mensch bewegt sich in einer Umgebung, die allgegenwärtige Computer beinhaltet. Jeder Computer ist ein aus Hard- und Software bestehender Netzwerkknoten, im folgenden als „Gerät“ bezeichnet. Die Nutzung dieses dynamischen, massiv verteilten Systems ist implizit und unbewusst. Jedes Gerät ist zu jeder Zeit an jedem Ort innerhalb eines begrenzten Kommunikationsraums verfügbar.

Dynamische Netzwerkstrukturen: In klassischen Netzwerksystemen werden Geräte und Kommunikationspfade als stationär betrachtet. Dabei liegt der Schwerpunkt der Betrachtung auf den Geräten, die über Kommunikationspfade vernetzt werden. Innerhalb des „UbiComp“-Gedankens sind Geräte mobil und damit lediglich temporär verfügbar. Mit dem Eintreten von Geräten in einen Kommunikationsraum, im folgenden einfach als Raum bezeichnet, werden Kommunikationspfade eingerichtet bzw. nach deren Austreten aus dem Raum wieder aufgelöst. Moderne Netzwerkkonzepte müssen damit insbesondere eine dynamisch handzuhabende Netzwerkstruktur berücksichtigen.

Dienste (Service): Geräte definieren sich über Dienste, die sie mittels geschichteter Kommunikationsprotokolle offerieren. Damit attributieren Dienste ein Gerät und ergeben zusammen das Geräteprofil. Dienste unterliegen hinsichtlich Kennung und Eigenschaften einer fest vereinbarten Struktur, damit Systeme (Netzwerke) entstehen können.

Spontane (Ad hoc-) Vernetzung: Dienste zweier Geräte (Peers) können sich in einem dynamischen Netzwerk zu einem System verbinden. Man spricht auch von der Einrichtung eines Peer-to-Peer-Links. Eigenschaften von Ad hoc-Netzwerken sind üblicherweise die Nutzung drahtloser Kommunikationswege, eine begrenzte Übertragungsbandbreite und eine im Vergleich zu den Kommunikationszeiten des Netzwerks langsame Bewegung der Geräte im Raum. Typisches Szenario: Ein PDA wird in einen Raum, z.B. in eine Funkzelle, hineinbewegt und soll die Verfügbarkeit eines Druckers feststellen.

Interoperabilität und Plattformunabhängigkeit: Das Gerätespektrum reicht vom Datenbankserver bis hin zu intelligenten Sensoren und Aktoren (Electronic Dust). In die Betrachtungen sind eine Vielzahl Programmiersprachen, Transportprotokolle, Betriebssystem- und Hardwareplattformen einzubeziehen. Eine Spezifikation der Kommunikationsmechanismen mittels XML erscheint sinnvoll.

Kontextsensitivität: Die Einführung von Kontextinformationen ermöglicht eine Vorselektion von Netzwerkzugehörigkeiten und Informationsinhalten.

3 Basistechnologien

Als wesentliche Basiskonzepte für Middlewaresysteme müssen Spezifikationen wie JINI, JXTA und Tuple Spaces betrachtet werden.

3.1 Jini

JINI [5,7] ist eine im Jahr 1999 durch SUN veröffentlichte Netzwerktechnologie, die den Aufbau von Ad-hoc Netzwerken zwischen verschiedenen Geräten erlaubt. Geräte bieten Dienste für andere Geräte über Jini-Charakteristiken. Jini basiert auf Java und ist daher für kleine Embedded Systems schwer implementierbar.

3.2 Tuple Spaces

Dem von D. Gelernter vorgeschlagenen Konzept der Tuple Spaces [4] liegt eine Shared Memory Idee zugrunde. Der LINDA Tuple Spaces realisiert die Idee des globalen Kommunikationspuffers für asynchrone und parallel arbeitende Anwendungen. Datenaustausch und Interprozesssynchronisation können als wesentliche Aufgaben genannt werden. Aus dem LINDA Tuple Space Konzept wurden im wesentlichen zwei Projekte abgeleitet: TSPACES von IBM [8] und JAVASPACES[7,5]. JAVASPACES ist Bestandteil des JINI-Projekts. Das Shared Memory Konzept der *InfoSpace-Servicearchitektur* basiert auf dem Konzept der Tuple Spaces.

3.3 JXTA

JXTA [6] wurde von der Firma SUN als Forschungsprojekt zur Entwicklung von modernen Peer-to-Peer Netzwerkfunktionen begonnen. Die Ergebnisse des Projekts wurden im Jahr 2001 veröffentlicht. JXTA ist lediglich eine Spezifikation von insgesamt sechs Protokollen und damit keine Implementierung. Die JXTA-Protokolle unterstützen die im Rahmen dieses Projekts innerhalb der *InfoSpace-Servicearchitektur* definierten Dienste bei der Selbstorganisation und Autokonfiguration von Ad hoc-Netzwerks.

Das Peer Discovery Protocol (PDP) ermöglicht sowohl die Veröffentlichung als auch die Suche von Ressourcen über sogenannte Advertisements. Ein Advertisement (Dt. Ausschreibung) ist ein programmiersprachen-neutrales XML-Dokument, das eine Netzwerkressource (Daten, Dienste, Devices, ...) repräsentiert. Das Peer Resolver Protocol (PRP) ist ein auf dem PDP basierendes Abfrageprotokoll. Das Peer Information Protocol (PIP) dient der Abfrage von Statusinformation wie Zustand, Laufzeitparameter und weitere Eigenschaften entfernter Geräte. Das Peer Membership Protocol (PMP) dient der Selbstorganisation von Geräten in Gruppen. Das Pipe Binding Protocol (PBP) ermöglicht die Instanzierung von virtuellen Kommunikationskanälen zwischen einem oder mehreren Geräten. Das Peer Endpoint Protocol (PEP) ist ein Mechanismus zur Feststellung von Routen, d.h. Sprünge über Geräte, die Nachrichten auf dem Weg zum Ziel durchlaufen können.

Die JXTA-Protokolle finden als Basis für die InfoSpace-Servicearchitektur Verwendung.

4 Die InfoSpace-Servicearchitektur

Die Grundidee der *InfoSpace-Servicearchitektur* liegt in der Bereitstellung eines objektorientierten verteilten gemeinsam geteilten Speichers, den „InfoSpace“. Jeweils alle Geräte, die zuvor ein Ad hoc-Netzwerk gebildet

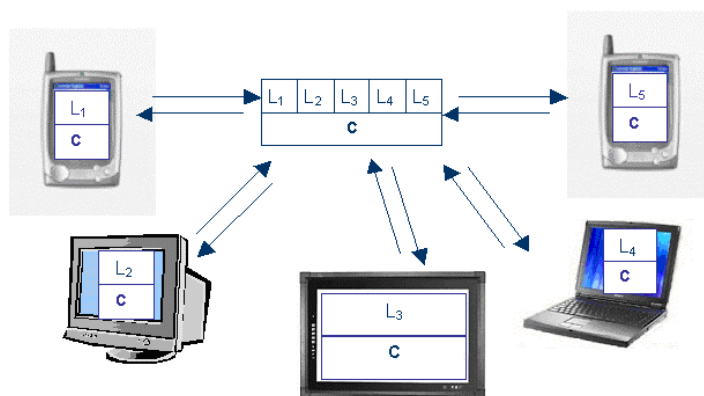


Bild 2: Bildung eines InfoSpaces durch mehrerer Geräte

Charakteristikum ein und denselben InfoSpace auf. Zur Definition einer Zusammengehörigkeit führen wir Kontexte ein, die einen InfoSpace beschreiben. Die kontextsensitive Behandlung von Infospaces ersetzt die klassi-

sche Authentisierung. Erst nach der Feststellung der Übereinstimmung aller Kontextparameter können zwei und mehr Geräte einen gemeinsamen InfoSpace bilden. Die *InfoSpaces-Servicearchitektur* wird durch Dienste repräsentiert. Neben den für die Verwaltung eines InfoSpace notwendigen Diensten ergänzen weitere, auf den JXTA-Protokollen basierende, das Konzept hinsichtlich der erforderlichen Ad hoc-Netzwerkfunktionalität. Auf eine detaillierte Erklärung aller Dienste wird in diesem Rahmen verzichtet.

Bild 2 zeigt die Bildung eines InfoSpaces, der in diesem Beispiel fünf physikalische „Local Spaces“ L_i zu einem logischen „Common Space“ C zusammenfasst und damit allen beteiligten Geräten zur Verfügung steht. Der InfoSpace entsteht an einem beliebigen Ort durch das Zusammentreffen beliebiger kontextgleicher Geräte. Der Common Space C besteht aus Beiträgen (Objekte) aller beteiligter Systeme. Jedes dieser Systeme ist damit gleichermaßen Client und Server, d.h. es nimmt nur für jeweils einzelne Objekte die Rolle eines Servers ein und ist ansonsten Client. Die Objekte von C werden von jedem Gerätebenutzer aus einem sogenannten Local Space L_i in C verschoben und damit veröffentlicht.

Es muss noch einmal verdeutlicht werden, dass abweichend von klassischen Ansätzen der objektbasierte globale Speicher C nicht durch eine zentrale Komponente innerhalb des verteilten Systems sondern durch die Interaktion der Middlewarekomponenten aller beteiligter Systeme, den InfoSpace Services, bereitgestellt wird.

5 Erste Implementierungen

Bis zum heutigen Tag wurde eine Java-gestützte Implementierung für MS-Windows 2000 entwickelt. Eine C-basierte Implementierung für C167-Mikrocontroller der Fa. Infineon ist in Arbeit. Eine weitere Implementierung für MS-Windows Pocket PC V. 3.0 Geräte ist in Planung.

6 Zusammenfassung und Ausblick

In dieser Kurzfassung wurde die *InfoSpace-Servicearchitektur* vorgestellt. Diese als Middleware einzustufende Software übernimmt das Management von Ad hoc-Netzwerken, deren Betrieb hinsichtlich einer geforderten Selbstorganisation und Autokonfiguration erhebliche Probleme aufwirft. Die *InfoSpace-Servicearchitektur* ermöglicht dynamisch einem Ad hoc-Netzwerk beitretenden bzw. dieses verlassenden Geräten einen sicheren und robusten Umgang mit konzeptbedingt unzuverlässigen Ressourcen. Diese Eigenschaften werden durch eine vertretbare Anzahl Dienste zum einen für das Ad hoc-Netzwerkmanagement und zum anderen für die Verwaltung von InfoSpaces, einem „Object Oriented Distributed Shared Memory“-Ansatz, erreicht. Die *InfoSpace-Servicearchitektur* weist als unter Schnittstelle die TCP/IP-Umgebung auf. Sie ist nach oben hin für jegliche Anwendungen offen. Bestehende, anerkannte Konzepte wie Tuple Spaces und JXTA werden aufgegriffen. Nach der Fertigstellung von drei Referenzimplementierungen für Notebooks, PDAs und eine „Embedded“-Plattform wird eine wesentliche Phase folgen, die sich mit der Realisierung von Szenarien befasst. Als Beispiele können InfoKiosk-Applikationen, Steuerungsanwendungen und drahtlose Betriebsdatenerfassungsanwendungen genannt werden.

7 Literatur

1. Fallstudie: Anbindung eines Bluetooth-Device, Elektronik 11/2002, S.66-69, WEKA-Fachzeitschriften-Verlag
2. Bluetooth Specification 1.1, www.bluetooth.com
3. Bluetooth module, www.ericsson.com
4. David Gelernter: Parallel Programming in LINDA, Technical Report 359, Yale University Department of Computer Science, Jan. 1985
5. <http://www.jini.org>
6. <http://www.jxta.org>
7. W. Keith Edwards: Core Jini, Prentice Hall, Dezember 2000, 2nd Edition
8. <http://www.almaden.ibm.com/cs/TSpaces/>