

Ubiquitous Computing (Ubiquitäre Informationstechnologien)

Vorlesung im WS 00/01

Hans-Werner Gellersen



Universität Karlsruhe
Institut für Telematik
Teleoperation Office

HWG 8-1

Einführung: Ubicomp-Netze

Typische Szenarien und Herausforderungen

- über Handy Geräte im Haus bedienen
 - heterogene Geräte und Netze (mobil/Heim)
 - kohärente Sicht auf Dienste im Heim
- Kamera sucht Drucker in fremder Umgebung
 - wie kann ein „geeigneter“ Drucker gefunden werden ?
 - wie unterhält man sich mit einem fremden Gerät ?

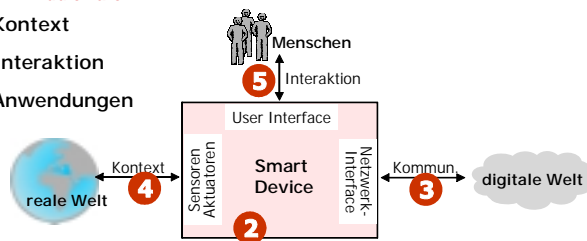
Wichtigste Herausforderung: Komplexität verbergen

- vor allem vor dem Anwender
- keine manuelle Installation / Konfiguration
- Abstraktionen für die Anwendungsentwicklung ⇔ Middleware

HWG 8-5

Ubiquitous Computing

- Vision/Grundlagen
- Smart Devices
- Vernetzung
 - Netzwerktechnologien
 - Middleware
- Kontext
- Interaktion
- Anwendungen



HWG 8-2

Middleware

Was ist Middleware ?

- „der Slash in Client/Server“
- Komponenten für Entwicklung und Einsatz verteilter Systeme
- angesiedelt zwischen Netzwerktechnologie(n) und Anwendung

Wofür ?

- Abstraktion von Netzwerkprogrammierung
- Interoperabilität: Zwischenschicht über unterschiedlichen Geräte- und Netzwerkplattformen
- Komponenten für allgemeine Aufgaben in verteilten Systemen: z.B. Name Service, Security Service,...

HWG 8-6

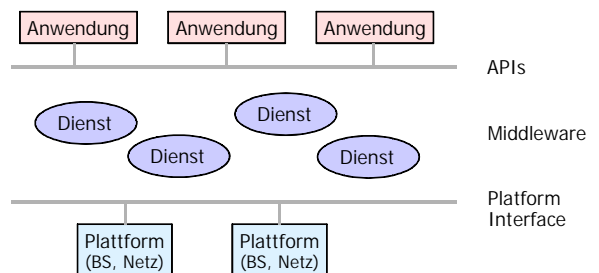
Middleware für UbiComp

- Einführung
- Service Paradigma
- Jini Service Infrastruktur
- JetSend Ad hoc Kommunikation

Gastvortrag von Lars Erik Holmquist,
Interactive Institute, Göteborg, S:
"Innovative Display Technologies"

HWG 8-3

Middleware



HWG 8-7

Einführung: Ubicomp-Netze

Netze für Ubiquitous Computing

- Diversifikation von Endgeräten: mobil, eingebettet, spezialisiert
- Mobilität: mobile Nutzer, mobile Geräte
- Allgegenwart: überall, insbesondere auch im Heimbereich
- Spontaneität: ad hoc Vernetzung von Geräten
- Konvergenz: Daten, Audio/Video, Steuerung

Entwicklungstrends

- Nutzung „alter Infrastrukturen“ und Schaffung neuer
 - trad. LANs, Funk-LANs, Plug&Play-Busse, Bluetooth, IrDA, Phonenumber, Powerline, ...
- Extrem heterogene Umgebungen: Geräte und Netze
- hohes Maß an Dynamik: Hot&Plug Play, mobile Netze, kurzlebige Verbindungen

HWG 8-4

Middleware

RPC: Remote Procedure Call (80er Jahre)

- Prozedurales Paradigma
- entfernter Prozeduraufruf analog zu lokalem Aufruf
- Code für die Kommunikation wird automatisch erzeugt

Objekt-orientierte Middleware (90er Jahre)

- Objekt-orientierte Programmierung
- Kommunikation mit entfernten Objekten über automatisch erzeugte lokale Proxies (z.B. „stubs“ in CORBA)
- Vermittlungsdienste (z.B. Object Broker)

Java Remote Method Invocation

- Java's Middleware für Methodenaufrufe von Objekten, die in verschiedenen VM ablaufen

HWG 8-8

Middleware für Ubicomp

Integration heterogener Geräte

- Gateways für kohärenten Zugang zu heterogenen Umgebungen
- Service Paradigma: dynamischer Verbund von Geräten ohne zentrale Komponente; spontane Bildung verteilter Systeme

Service Gateways

- Bündelung von Diensten über Gateways
- Residential Gateways: Verbindung zwischen Heimnetz und Außenwelt (= Internet)
 - bietet Geräten im Haus Zugriff auf Internet-Dienste
 - bietet externen Dienst Anbietern kohärenten Zugang zu Geräten/Infrastruktur im Haus (z.B. für Fernwartung, Sicherheitsüberwachung,...)
 - Administration durch Gateway Operator
 - Standardisierung: Open Service Gateway Initiative (OSGI)

HWG 8-9

Service Paradigma

Netzwerk-zentrisch

- „the network is the computer“
- Netzwerk = Hardware und Softwareinfrastruktur für Dienste
- Sichtweise: „Netzwerk, an das Geräte angeschlossen sind“ (statt „Geräte, die vernetzt werden“)
 - Netzwerk existiert immer, Geräte/Dienste sind transient
 - Komponenten und Kommunikationsbeziehungen kommen und gehen

Spontane Vernetzung

- Services finden sich in der offenen Netzwerkumgebung zu zeitweiligen Verbundsystemen zusammen
- müssen sich dazu nicht a priori kennen
- typisches Szenario: Client wacht auf und fragt nach Diensten in der lokalen Umgebung

HWG 8-13

Middleware für Ubicomp

- Einführung
- Service Paradigma
- Jini Service Infrastruktur
- JetSend Ad hoc Kommunikation

HWG 8-10

Service Paradigma

Spontane Vernetzung von Services

- wie werden Services aufeinander aufmerksam ?
- wie können bestimmte Services in einer fremden Umgebung gefunden werden ?
- wie verständigen sich Services, wenn sie sich gefunden haben ?

Infrastruktur für Service Discovery

- „Registry“: Verzeichnis/Vermittlung von Services
- Protokolle zum Registrieren und zum Anfragen von Services
- Protokolle für Client-Zugriff auf Service, und für die Nutzung von Services durch Clients
- z.B. Sun's Jini aufbauend auf Java/RMI, Microsoft's UPnP (Universal Plug & Play)
- z.B. HAVi (Home Audio/Video interoperability) aufbauend auf IEEE.1394 für Home Entertainment Dienste

HWG 8-14

Service Paradigma

Everything is a Service

- Geräte ebenso wie Software
- vgl. Objekt-orientierung: „everything“ is an object
- Services werden durch Interfaces definiert, über die sie ihre Funktionalität zur Verfügung stellen
- Services werden beschrieben durch Typ und Attribute
- Services können sich zu Systemen verbünden („federation“)

Beispiele für Services

- Kameram, Drucker, Fax, Scanner, Speicher, Rechenleistung, ...
- Türöffner, Beleuchtung, Alarmanlage, Stromzähler, ...
- Rechtschreibprüfung, Formatkonvertierung, ...
- Online Banking, Aktienhandel, ...
- Hotelführer, Stadtplan, ...

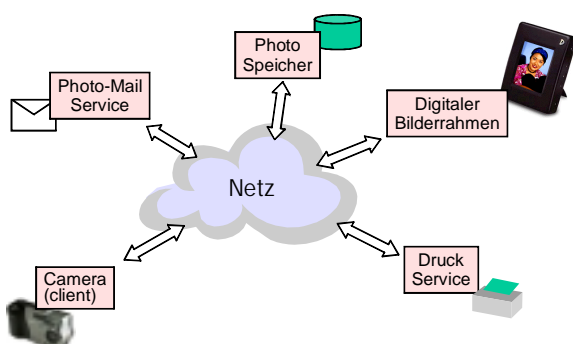
HWG 8-11

Middleware für Ubicomp

- Einführung
- Service Paradigma
- Jini Service Infrastruktur
- JetSend Ad hoc Kommunikation

HWG 8-15

Service Federation: Beispiel



HWG 8-12

Jini Service Infrastruktur

Hauptkomponenten

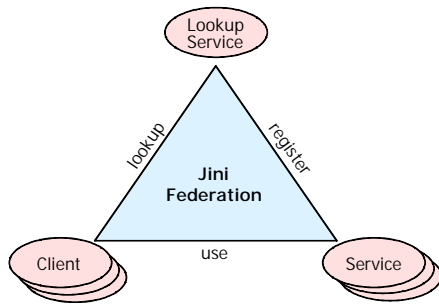
- Lookup Service (LUS): „Registry“ für Services
- Protokolle basierend auf TCP/UDP/IP
 - Discovery & Join, Lookup von Services
- Proxy Objekte
 - als lokale Vertreter für Services

Lookup Service

- Verzeichnis ähnlich RMI Registry
- Aufgabe: „Helpdesk“ für Services/Clients
 - Registrierung von Services, die angeboten werden
 - Verteilung von Diensten an anfragende Clients

HWG 8-16

Jini Lookup Service



HWG 8-17

Lookup: Service Matching

Service Template

- Service ID (Registration Number): kann angegeben werden, falls Dienst bereits bekannt ist (durch frühere Nutzung)
- Service Typ: definiert durch die Schnittstelle
- Attribute (sog. Entries), die den Service beschreiben

Service Matching

- Matching über Attribute: Mehrwert gegenüber traditionellem Naming Service: Service-Auswahl über beschreibende Merkmale
- aber nur exaktes Matching, kein „größer als“, keine Query-Sprache
 - z.B. Anfrage an „600dpi“ Drucker matcht nicht mit registriertem „1200dpi“ Drucker

HWG 8-21

Discovery: Finden eines LUS

Finden eines Lookup Services

- ... ohne a priori Kenntnis des Netzwerks
- Service Provider sucht LUS um Service anzumelden (register)
- Client sucht LUS um einen Service anzufragen (look up)

Discovery Protokoll

- Multicast-Anfrage an bekannten Port
- Lookup Service lauscht auf entsprechendem Port und antwortet mit Proxy Objekt (interface `ServiceRegistrar`)
 - Proxy Objekt wird in die anfragenden Service geladen
 - Kommunikation mit LUS dann über den Proxy
- weitere Discovery-Protokolle
 - Unicast: Service kann LUS direkt ansprechen, wenn er die IP-Adresse schon kennt
 - Multicast Announcement: LUS meldet sich per Multicast, z.B. nach Ausfall

HWG 8-18

Service Paradigma

Spontane Vernetzung von Services: Jini Konzepte

- wie werden Services aufeinander aufmerksam?
 - Jini: Lookup Services als Vermittlungsstelle, Registrierung von Services über Discovery & Join
- wie können bestimmte Services in einer fremden Umgebung gefunden werden?
 - Discovery von Lookup-Services als Help-Desk in fremder Umgebung
 - Lookup anhand von Service Templates, insbesondere auch anhand beschreibender Attribute
- wie verständigen sich Services, wenn sie sich gefunden haben?
 - Service Proxy wird in den anfragenden Client geladen
 - beliebiges Protokoll, kein spezifisches Aushandlungsverfahren

HWG 8-22

Join: Registrieren eines Services

Join Protokoll

- Service Provider hat einen Proxy des LUS für die Kommunikation empfangen
- Provider registriert über den Proxy seinen Service: `register()`
- Provider übergibt dabei dem LUS
 - den eigenen Service Proxy
 - Attribute, die den Dienst beschreiben (z.B. „600 dpi“, „version 21.1“, ...)
- Service tritt mit dem Join in den Jini-Verbund ein
 - Provider kann nun über den LUS gefunden und von anderen Services genutzt werden

HWG 8-19

Middleware für Ubicomp

- Einführung
- Service Paradigma
- Jini Service Infrastruktur
- JetSend Ad hoc Kommunikation

HWG 8-23

Lookup: Suchen von Services

Lookup Protokoll

- Client sucht bestimmten Service
- kennt LUS und verfügt über Proxy für die Kommunikation (via Discovery Protokoll)
- sendet Anfrage an LUS in Form eines „Service Template“
 - ID, Typ, Attribute
- LUS antwortet mit keinem/einem/mehreren passenden Services
 - ggf. Auswahl im Client
- Client erhält vom LUS Proxy des vermittelten Services
- Client nutzt Proxy für direkte Kommunikation mit dem Provider
 - beliebiges Protokoll
 - Proxy: Gateway zu Service-Funktionalität beim Provider
 - Proxy kann aber auch (Teil der) Service-Funktionalität implementieren, d.h. Ausführung beim Client

HWG 8-20

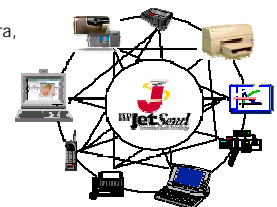
JetSend Ad hoc Kommunikation

Spontane Vernetzung von Services

- wie verständigen sich Services, wenn sie sich gefunden haben?

JetSend (HP, 1997)

- Universelles Kommunikationsprotokoll für Informationsgeräte
 - beliebige Geräte: Drucker, Kamera, Scanner, Projektor, ...
 - Aushandlung von Formaten zum Austausch von Information
- Punkt-zu-Punkt
 - geht von direkter logischer Verbindung zw. Geräten aus
 - Abstraktion vom Netzwerk (Medium, Verbindung, Transport)



HWG 8-24

JetSend Ad hoc Kommunikation

JetSend Protokoll

- Geräte- und gerätetypunabhängig
 - Im Protokoll und in den Geräten befinden sich keine Informationen über andere Geräte/Typen
 - Beispiel: JetSend-Kamera kann Information mit beliebigem JetSend-Drucker austauschen ohne den Druckertyp zu kennen
- Aushandlung der Informationsbeschreibung
 - Geräte verhandeln über unterstützte Codierungen mit dem Ziel, den größten gemeinsamen Nenner zu finden
- Hauptkonzepte
 - Surface: Gegenstand des Informationsaustauschs
 - Surface Interaction: das Protokoll für Informationsaustausch
 - E-Material: Austauschformat für Surface Interaction

HWG 8-25

Beispiel: JetSend E-Material

Level	Attribute	Value
	vEncoding	vImage
vImage	vSize	(576000,756000)
vImage	vOpacity	255
vImage	vColorSpace	vSRGB vGray Alternativen
vImage.vSRGB	vPixelFormat	24
vImage.vSRGB.24	vResolution	(150,150)
vImage.vSRGB.24.(150,150)	vPixels	(1200,1575)
vImage.vSRGB.24.(150,150)	vCompression	vJPEG
vImage.vGray	vPixelFormat	1
vImage.vGray.1	vResolution	(300,300)
vImage.vGray.1.(300,300)	vPixels	(2400,3150)
vImage.vGray.1.(300,300)	vCompression	vRLE

abhängig von Auswahl

Level: Ebene, auf die sich Attribut-Wert-Paar beziehen

HWG 8-29

JetSend Ad hoc Kommunikation

Surface

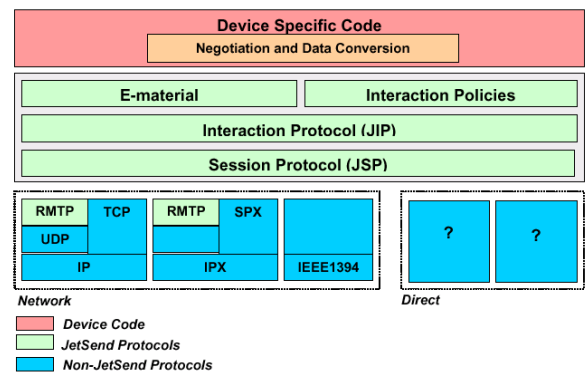
- „Oberfläche“: darstellungsnahe Information, z.B. Bilder, Dokumente, Statusmeldungen
- bestehen aus Beschreibung und Inhalt

Surface Interaction

- Austausch von Surfaces zwischen Geräten
 - Original-Surface wird als *Expression* („Ausdruck“) bezeichnet
 - ein empfangendes Gerät erhält eine *Impression* („Eindruck“)
- Aushandlung von Inhalts-Codierungen
 - beim Erzeugen einer Impression wird nur die Beschreibung gesendet, die eine Liste alternativer Codierungen enthält
 - der Empfänger wählt anhand der Beschreibung eine geeignete Codierung, in der der Inhalt angefordert wird

HWG 8-26

JetSend Architektur



HWG 8-30

JetSend Ad hoc Kommunikation

Surface Interaction Messages

SurfaceRequestMsg

- Anforderung einer Surface (identifiziert durch einen Namen)

SurfaceMsg

- Erzeugen einer Impression, Senden der Beschreibung

ContentRequestMsg / ContentReplyMsg

- Übertragung des Inhalts

DescriptionRequestMsg / DescriptionReplyMsg

- Übertragung weiterer Beschreibungen

SurfaceChangeMsg

- Benachrichtigung über Änderungen einer Surface

SurfaceReleaseMsg

- Verbindung zwischen Expression und Impression aufheben

HWG 8-27

JetSend Zusammenfassung

Informationsaustausch garantiert

- alle JetSend-Geräte unterstützen zumindest eine Default-Codierung ihrer Geräteklasse (Image, Video, Audio, ...)

Interoperabilität

- Peer-to-Peer Informationsaustausch zwischen Geräten ohne vorherige Kenntnis voneinander

Modularität

- Separate Entwicklung von Systemkomponenten (Hi-fi Modell)
- Protokoll und Gerätefunktionalität sind getrennt
 - Protokoll definiert „Rohr“ durch das Information fließt
 - Protokoll beeinflusst Funktionalität nicht

HWG 8-31

JetSend Ad hoc Kommunikation

E-Material

- Format für Surface Messages
 - definiert Format, in dem Information (Beschreibung und Inhalt) zwischen Geräten ausgetauscht wird
 - aber nicht wie Geräte diese Information verarb./speichern
- Beschreibung von Codierungsalternativen
 - Tabelle mit Attribut-Wert-Paaren
 - jede Codierung wird mit Standardattributen assoziiert
 - einige Attribute können Alternativen beinhalten
 - Hierarchische Beschreibung ermöglicht geschachtelte Auswahl aus Alternativen

HWG 8-28

Ubiquitous Computing

1 Vision/Grundlagen

2 Smart Devices

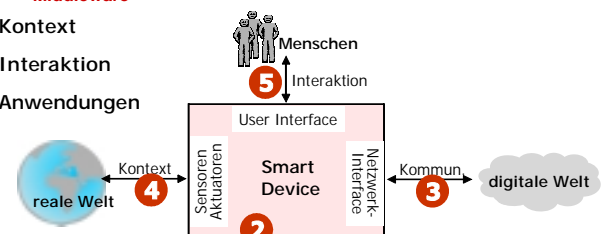
3 Vernetzung

↳ Netzwerktechnologien
Middleware

4 Kontext

5 Interaktion

6 Anwendungen



HWG 8-32