Context-aware Communication Services

Research & Experiences

Manuel Görtz, Ralf Ackermann, Johannes Schmitt, Ralf Steinmetz

Multimedia Communications (KOM)
Technische Universität Darmstadt
Merckstr. 25 64283 Darmstadt

http://www.KOM.tu-darmstadt.de
Introduction
Motivation

Observation of Daily Communication

● irrelevant communication
  ◆ tele-marketing, surveys, out-of-office announcements

● annoying disruptions
  ◆ meetings, dinner, movie, theater, tennis match, etc.

● caller unaware – not knowing the availability of the callee
  ◆ lack of coordination leads to phone tag, missed opportunities, etc.

● interaction overload – decreasing latency increases junk
  ◆ bombarded by irrelevant communication versus noticing timely communication (like flight changes)

● device overload
  ◆ overwhelmed by managing and choosing the right channel
    ■ different media = different device = different address
Geek Solution

LASS MAL SEHEN ... ICH HABE MOBILTELEFON, PIEPER, PALMTOP, PERSONLICHEN TERMINKALENDER, DRAHTLOSES MODEM ...

JA, MAN KANN SAGEN, INGENIEURE WEIT UND BREIT BENEIDEN MICH. SIEHT GUT AUS ... WIRKLICH GUT ...

MEINE UMFASSENDE PERSONLICHE ELEKTRONIK MACHT MICH ZUM HERRSCHER ÜBER DIE SCHLECHTER AUSGERÜSTETEN INGENIEURE.

TECHNO-BILL!! SIEHT AUS, ALS HÄTTE JEMAND GERade EIN FAX BERECOMMEN.
What do we want?

- Filtering of incoming communication requests
  - handle incoming communication requests
  - redirect/terminate calls
  - apply e-mail filter like mechanisms
- Avoiding unnecessary communication
  - caller does not want to be disturbed
  - callee wants to save time for unsuccessful calls
- Customized services
  - services that fit user’s needs
  - develop & deploy mechanisms to create ‘own’ services
  - convenient and safe development and execution environment
  - “wizard”-like support
What do we have

Evolution of Communication Services

● POTS
   ♦ Basic Call
   ♦ alerting → phone rings
   ♦ user goes on hook/off hook

● ISDN/IN
   ♦ Supplementary Services
   ♦ calling party number can be displayed
   ♦ user may switch services on/off

● Mobile
   ♦ Multimedia Service
   ♦ callers categorized and attributed
     (ring tones, pictures)
   ♦ user manages profiles
Can we do better?

- **Secretariat**
  - hub of incoming communication
  - call handling based on
    - callee’s context
    - experience and knowledge
  - multi-modal interface to specify rules
  \[\Rightarrow\] communication is efficiently handled according to the context of the users

- **Human Face-to-Face Communication**
  - learned from early childhood
  - follows certain rules
  - ‘good’ feeling for right starting point
  \[\Rightarrow\] communication and interaction between humans always happen in a specific situation, a certain context, and in a particular environment

- **Context is the key concept**
Context-aware Communication Services
Efficient Communication

- **Context Sharing**
  - caller can avoid unnecessary “calls” (alertings)
  - shift decision to start call towards the caller

- **Context Filtering**
  - system avoids disturbing calls for callee
  - selects appropriate service to handle incoming calls
Context- Extended Service Model

- Concept
  - put user into focus
  - disappearing (technical) helper
  - context to adapt service control
  - context to trigger new services

```
media and/or signaling
```

```
Service Execution

Service Description

Service Enforcement
```

```
Context as parameter

Context Acquisition

Context Synthesis
```

Media and/or signaling
Context Definition

- **Context Definition**
  - Circumstances in which an event occurs [Dictionary Def]
  - A *Context* $\xi$ is an abstract and meaningful description of the relationship between objects and their environment. A context is a rich object consisting of *context features* and can be approximated by a characteristic function $\chi$. A *context label* $\lambda$ is assigned to each context.

- **Properties**
  - enabling effect
    - new class of services
    - disappearing from the user’s perception
  - automation
    - triggering of actions
  - reduction
    - of input and output
  
  $\implies$ for the system a context is just a label (e.g. data structure)
Introduction

Context-aware Services
- Efficient Communication
- Context-Extended Service Model
- Context Definition
- Context Spiral Model
- System Architecture
- Presence Information Data Format – PIDF

Session Initiation Protocol

Real World Experiences

Thank you!

---

Context Spiral Model

How to handle context

- **Acquisition**
  - sensing
  - data fusion
- **Synthesis**
  - feature extraction
  - aggregation
  - decision making
- **Dissemination**
  - store context (data)
  - distribute context (data)
- **Use**
  - apply context
System Architecture

- **Design recommendation**
  - divide context acquisition from context use
  - provide framework for program developer
  - provide communication mechanisms to distribute context

- **Components**
  - context aggregation network
  - context server
  - context-aware communication services
Exchanging context/presence information

- **Purpose of notation**
  - store all relevant information incl. history
  - provide interoperability to PIDF clients

- **Joint work with tzi (Ott, Kutschner)**

- **Added tags for context**
  - `<context>` element:
    - `<current-context>`
    - `<where>`, `<privacy>`
    - `<future-context>`, `<past-context>`
  - `<sensor>` element:
    - `<auth-class>`, `<owner>`, `<decay function>`
    - `<value>`, `<unit>`, `<type>`, `<dependability>`
Session Initiation Protocol
Application Level Signaling Protocol

- **Concept**
  - fast in the core, smart at the edges
  - horizontal integration (of protocols)
  - request/response transaction model
    - register users, setup, modify, terminate sessions
  - components
    - end-systems (UA), SIP proxies, Registrars

- **Protocol extensions**
  - subscribe/notify event package
  - SIP for instant messaging
  - SIP security extensions
  - QoS extensions
Demand for new services

● Distinction of provider
  ♦ argument for customers to change provider
  ♦ and technology (PSTN → IP Telephony)

● Public Switched Telephone Network
  ♦ service provided by the network
  ♦ rather closed group for standardization

● IP Telephony
  ♦ service intelligence in the end-systems
  ♦ open standards
  ♦ users have access to the network

● Two principle services classes (here)
  ♦ end-system services
  ♦ 3rd party call control services running on a server
Call Processing Language (CPL) [RFC 2824]

- 3rd party call control
  - XML-based script language
- Simple, extensible, not Turing-complete language
  - no loops, variables, recursion
  - no execution of external programs
- Call processing action represented as Directed Acyclic Graph
  - bounded and predictable
  - finite memory and time

Components

- switches
  - address, time, string, priority, ...
- actions/subactions
  - re-direct, proxy, reject, ...
CPL Example

<clp>
  <incoming>
    <address-switch field="origin" subfield="host">
      <address subdomain-of="kom.eu">
        <location url="sip:mgoertz@kom.eu">
          <proxy timeout="10">
            <busy> <sub ref="voicemail" /> </busy>
            <noanswer> <sub ref="voicemail" /> </noanswer>
            <failure> <sub ref="voicemail" /> </failure>
          </proxy>
        </location>
      </address>
      <otherwise>
        <sub ref="voicemail" />
      </otherwise>
    </address-switch>
  </incoming>
</clp>

Address Switch
  field: origin
  subfield: host
  subdomain-of: kom.eu
  otherwise
  location
  url: sip:mgoertz@voicemail.kom.eu

proxy
timeout: 10s

Voicemail
  location
  url: sip:mgoertz@voicemail.kom.eu

redirect
Providing methods to develop context-aware communication services

- New Elements
  - Context-Lookup
    - provide context to CPL-Engine
    - query Context Server
  - Context-Notify
    - context is shared on request
    - send context using NOTIFY-Messages
  - Context-Switch
    - select next path depending on current context
  - Answer-Switch
    - select appropriate next path depending on current context
- implemented in VOCAL and SER CPL-Engines
CPL Editor

Creation of Services by script writing – not user friendly → GUI
Real World Experiences
Implementation / VOCAL

SIP suite from VOVIDA
  ● Open source SIP Server / (www.vovida.org)
  ● “all in one” packet with multiple server-processes:
    ▶ Provisioning (maintenance / management)
    ▶ Marshall (Message processing / conversion)
    ▶ Authentification
    ▶ Redirect / Registrar
    ▶ Call Detail Record (Billing)
    ▶ Heartbeat Server
    ▶ Policy Server
CPL Engine VOCAL

Properties
- executing on incoming/outgoing calls
- every feature has its own process/port
- transform CPL script to feature FSMs at first call

Provisioning Server
- CPL A -> port X
- CPL A -> port X...
- user A -> feature Y
- user B -> feature X, Y...

Feature Server
- start-features
- start()
- call - user A
- call(Y)
- port X -> CPL A
- port Y -> CPL B
- port Z -> ...

Thank you!
Implementation / SER

Sip Express Router – SER
- Open Source SIP server
  [www.iptel.org/ser/](http://www.iptel.org/ser/)
  - lightweight basic installation:
    - only Redirect / Registrar
- Modular / Plugins
  - authentication
  - MySQL
  - web interface
  - CPL
  - 3rd party modules
CPL Engine SER

- CPL script parsed after upload
- binary CPL script stored in external MySQL db
- script can be executed during incoming/outgoing call

INVITE  UPLOAD(CPL)

SER

CPL module

Load CPL

Execute

MySQL

CPL

Parser

...
Self-learning mechanisms

● **Drawbacks**
  ♦ writing rules takes time
  ♦ rules become outdated and must be modified/replaced

● **Automatic adaption of rules**
  ♦ provides user friendly handling
  ♦ user intervention: only by “feedback”

● **Dynamic evaluation model building**
  ♦ based on a set of user feedbacks and the related sensor values in this situation
  ♦ Methods: Bayesian net / neural net / fuzzy logic / decision trees

Self organizing sensor evaluation

● self description (type/location/relation) for sensors
● automatic discovery, selection, query of sensors
● provides scalability
Summary

- Communication
  - has become ubiquitous
  - demand to handle communication

- Services
  - distinction between different providers
  - might become driving force of IP Telephony
  - demand for customized services

- Context
  - provides information to make communication more efficient
  - allows to build customized and adaptive services
  - support whole chain from sensors to context representation

- Call Processing Language – CPL
  - provides mechanism to build and deploy safe services
  - have been enhanced to consider context information
  - changes implemented in wide spread CPL-engines
Demo Setup – Context-aware Call Diversion

LOCATION TRACKING

CALL HANDLING

Scenario:
- Caller "office" or "meeting"
- If caller is in room A, divert call to room A
- If caller is in room B, divert call to room B

Diagram:
- Room A
- Room B
- Badge
- IR sensor
- Radio
- Base station
- Call handling
- Caller
- "office" or "meeting"
- CPL
- SER
- Evaluation
- Sensor server

Implementation:
- Vocal
- CPL Engine Vocal
- SER
- CPL Engine SER

The next step: Demo Setup - Context-aware Call Diversion

Thank you!

Manuel Görts, February 28, 2005
Thank you!