Trends in Mobile Computing – From Mobile Phone to Context-Aware Service Platform

Thomas Strang
German Aerospace Center (DLR)
Oberpfaffenhofen, Germany
thomas.strang@dlr.de
Outlook:

- Motivation
- Implications of Mobility
- From Mobile Phone to Service Platform
  - Autonomy
  - Service Discovery and Execution
  - Context Awareness
- Contextual Interoperability
- Context Ontology
Motivation: Mobile Devices...
Mobile Devices!

- **multiple inhomogenous networks**
  - short range: IrDA, Bluetooth, Wireless LAN
  - wide area: (HS)CSD, GPRS, UMTS
  - often also no network / weak connectivity

- **resource limited (RLD)**
  - battery power
  - processing power (performance)
  - memory (volatile and non-volatile)

- **programmable, also by third parties**
  - WML script
  - Java2 Micro Edition (J2ME) / i-Mode
  - native code
Types of Mobility and its Implications

**What is different?**

- **User Mobility**
  - user interaction model

- **Device Mobility**
  - smaller, battery-driven devices
  - multiple inhomogeneous networks
  - often no network
  - position becomes parameter

- **Session Mobility**
  - issues in data distribution

- **Service Mobility**
  - (Code Mobility)
  - distributed lifecycle management
  - security is strong issue
What is Context (1/2)?

- **Definition of Context Information**
  - “A context information is any information which can be used to characterize the state of an entity concerning a specific aspect”
  - “An entity is a person a place or in general an object”
  - “An aspect is a classification, symbol or value-range, whose subsets are a superset of all reachable states”

- **Example:**
  - **Entity:** Telephone
  - **Aspect:** Call State
  - **Context Information:** specific state

```
{ { on-hook & idle },
{ on-hook & ringing },
{ off-hook } }
```

```
CI( entity=+4917912345,
aspect=callState ) =
{ off-hook }
```
What is Context (2/2) ?

Definition of Context

- “A context is the set of all context information characterizing the entities relevant for a specific task in their relevant aspects”
- “An entity is relevant for a specific task, if its state is characterized at least concerning one relevant aspect”
- “An aspect is relevant, if the state with respect to this aspect is accessed during a specific task or the state has any kind of influence on the task”

Example:

Task: traveller reaching plane using public transport

Relevant Entities:
- traveller
- carrier (taxi, train, bus...)
- infrastructure (roads...)
- plane to reach

Context: all CI of relevant aspects

- willingness to spend money
- duration/delay, cost
- delay
- time of departure + buffer
Context Awareness

Definition of Context Awareness:
▶ “A system is context aware if it uses any kind of context information before or during service provisioning or service usage”

Location Awareness is special issue of Context Awareness (but by far not the only one!)

Two main benefits from Context Awareness:
▶ Adaptation of services to changes in environment reduces amount of interaction with user
▶ Improvement of UI (particularly on small mobile devices)
A service is...

- The **concept of services** as approach for **solving complex tasks** in a **collaborative manner**.

  - A **service** is a **namable entity** which is **responsible for providing information or performing actions** with specific characteristics.

- A service w.r.t. this definition may be a **carrier service**, but is **not** restricted to be such.

- Services are **independently deployable** and need a **framework** for **discovery, transport** and **execution**.
Browser Approach vs. Service Platform Approach

- **WAP+WML/XHTML**
  - requires online connection
  - limited scripting possible
  - no storage, just caching
  - one network
  - just UI

- **Java**
  - works even offline (autonomy)
  - high variety by programming
  - file system, prefetching
  - multiple networks (cost metrics)
  - multi-party programming (enabled by late binding)
  - adaptivity to environment (context awareness)
  - extendable through Web Services
Mobile Service Platform based on MIDP

Optional Packages

Java 2 Enterprise Edition (J2EE)
Java 2 Standard Edition (J2SE)

Java Virtual Machine

Optional Packages

Personal Profile
Foundation Profile
MID Profile

CDC
CLDC

CVM
KVM

Java Card APIs

Remarkable specials (compared to other Java versions):

- VM: no float, no double, preverification required
- “closed” late binding -> “weak” mobile code
- HTTP is the only mandatory network protocol -> no sockets, no RMI
- No custom classloaders, no reflection, no object serialization
“Three Columns” of Architecture

- **Autonomy** through device local Service Registry

- **Context-aware Service Discovery**

- **Context-aware Service Execution**

---

**Mobile Application Suite**

- Data
- Service Module
- Service Module
- Service Module

**Invocation Interface**

**Search Interface**

**Register Interface**

**Sensor Processing**

**Context aware service matcher**

**Context aware offer filter**

**Registrations**

**Code Snippet DS**

**Network Communication Manager**

- GSM
- GPRS
- WLAN
- Bluetooth
- IrDA

---

**Signature Interop Specification**
- Service Offer
- Software Interface Signature
- List of Addressable Instances (URLs)
- Protocol Interop Specification (optional)
- Semantic Interop Specification (optional)
- Context Interop Specification
- Context Information Value Ranges (optional)
- Dependency Declarations (optional)
- Cost Indications (optional)
- (Code Snippets) (optional)

**Service Registry**

**Local Service Implementations**

**Service Mgt Application**

**Execution Handler**

**Discovery Handler**

**Network Adaptation**

---

**Invocaiton Interface**

---

**Service Offer**

---

**CoS feedback**

---

**Context Information Value Ranges**

---

**Dependency Declarations**

---

**Cost Indications**

---

**(Code Snippets)**

---
It is essential being able to handle weak connections and periods of no connection.

Without network connectivity, facilities for
  ▪ Service disposition
  ▪ Service discovery & execution
  ▪ Service interaction interface (e.g. UI)

are required to run on the mobile device!

Low End Service Platform Implem. based on CLDC / MIDP
  ▶ Java based service mgt. application including Service Registry
  ▶ leaning towards Web Services based on XML & HTTP
Context-aware Service Discovery

- Hierarchical organized service discovery, starting at mobile

- **WSDL**
  - Signature Interop Specification
    - Software Interface Signature
    - List of Addressable Instances (URLs)
  - Protocol Interop Specification (optional)
  - Semantic Interop Specification (optional)
  - Context Interop Specificication
    - Context Information Value Ranges (optional)
    - Dependency Declarations (optional)
    - Cost Indications (optional)
    - [Code Snippets] (optional)

- **WSCSI**

- **DAML+S**

- **Context extension**

- **capacity limits + communication costs** of mobile device is big issue
  - push vs. pull

- easily expressed in XML
Distributed Mediator Operation: *spread & join*

- **Abstract** Service Invocation
  - using multiple links
  - cost mgt neccessary

- Makes *service execution context aware*

- **Advantages**: Multiple Result Merging/Selecting; Cascading; Update/Revocation
Ubiquitous Computing: Three main facets

- **context awareness**
  - benefit much from context awareness due to typically limited resources, display and keypad
  - particularly from mobility: location information

- **smart devices**
  - good context sensors

- **ad-hoc networking**
  - Network Situation (multiple network types, cost metrics etc.)
### Interoperability Levels

<table>
<thead>
<tr>
<th>Service Interoperability</th>
<th>Programming Language Interoperability (IDL)</th>
<th>Platform Interoperability (RPC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context Level</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Own context level** eases **separation of services** which are interoperable on classic levels, but not on context level (and vice versa)
  - Example: electronic public transport timetable service
- **context is not covered very well** in classic levels
- **enables thematic concentration** for correctness and integrity
Interoperability in Middleware

Middleware Task:

Interoperability Evaluation during Service Discovery

- service search (feed)
- service selection (feed back)

Interoperability Evaluation during Service Execution

- lifecycle monitoring and management
- notification about and adaptation to context changes
- inter-provider handover
MNM Service Model: Service View

Service View Diagram

Interoperability in Extended MNM S-Model

Context Ontology
Language
Design of Context Ontology Language (CoOL)

- XML & XML schema based
  - probably based also on RDF
- used to model
  - aspects defining valid context information
  - relevance conditions
  - actors and dependencies
- enables dynamic (at runtime) multi-party service interoperability checks at new context level
- **CoOL** is employed during service discovery and execution to negotiate and monitor context to enable context awareness of middleware and services
- Middleware uses CoOL statements to link distributed object-associated context information with context un-aware objects and services
Open issues: Current Investigations on CoOL

- **Pure XML schema vs. RDF / RDF Schema**
  - complexity of RDF/RDFS vs. resource limited devices
  - some drawbacks of RDF have already been identified e.g. in [Haustein2001] [Furche2001] [Indulska2003]
  - but significant body of work has been done with RDF

- **“Plugin Model” to DAML-S vs. Standalone Model**
  - DAML-S / DAML+OIL adds additional complexity

- **WebOnt (OWL) vs. minimum language**
  - OWL in very early state
Summary

- **Introduction to MIDP devices**

- **Autonomous Context-Aware Services for Smart Mobile Devices**
  - Service Discovery and Execution on the Mobile Device (through device local Service Registry as part of framework)
  - Context Information employed during discovery and execution
  - Concept using *reasonable* architecture elements (Web Services)

- **New context level** of service interoperability
  - advantages particularly in Ubiquitous Computing Environments

- **Clear definitions** what context is and what not
  - Introduction of the aspect
  - affiliated context awareness

- **Context extension** of the MNM service model
  - Identification of interoperability requirements

- **Outline** of Context Ontology Language (CoOL)
  - still some major open issues
Thank You!

Any Questions?

Dipl.-Inform. Thomas Strang  
mailto:thomas.strang@dlr.de  
http://www.dlr.de/kn/kn-s/strang

German Aerospace Center (DLR)  
Site Oberpfaffenhofen  
Institute of Communications and Navigation (KN-S)  
PO Box 1116  
82230 Wessling-Oberpfaffenhofen  
Tel: +49-8153-28-1354  
Fax: +49-8153-28-1871