Motivation of this paper

• Mobile devices have limited resources
  – Bandwidth, memory, CPU, display, battery
• Mobile devices are exposed to variation of context
  – Bandwidth, connectivity, neighbors, resources
• In particular disconnection happens frequently

• However:
Motivation of this paper (2)

- Middleware for traditional systems is monolithic
- It works based on the black box approach
  - Hides details from applications
- Mobile applications need to be context aware and adaptive to changing context!
- Example: adapt to stream video in black and white (from colour) if the bandwidth drops

Transparency vs Awareness

- Application transparent approach
  - Middleware provides best effort adaptation to context changes (context hidden)
- Sometimes however the application is in the best position to make decisions on operating conditions and adaptation strategy
  - This is often the case in mobile computing, so awareness is good
Content of the Paper

- MobiPADS is a middleware which supports active deployment of context aware services for mobile computing
  - Assume a cellular network with a backbone and some wireless linked nodes which are mobile
- Services (mobilets) are composed to offer augmented services to applications
- MobiPADS uses REFLECTION to achieve adaptation

Reflective Middleware

- Reflection is the ability to monitor and change its computation and its semantics
- Reflective Middleware is able to model itself and auto manipulate its behaviour through *introspection and interception*
- *Introspection* is the ability to observe its own state
- *Interception* is the ability to modify it
Middleware Adaptation

- Interservice adaptation (middleware level)
  - Changes in context trigger service composition at middleware level
  - Eg: drop of bandwidth and increased error rate -> adaptation example: content transcoding of images from colour to grayscale and addition of a local service to perform local error control over wireless link

Middleware Adaptation (2)

- Intraservice adaptation (middleware level)
  - Service objects can react to context events (they subscribe and are notified about the event)
  - Service object can be programmed to behave different after receiving an event notification
  - Eg, a transcoding service object can be programmed to receive event on bandwidth. When bandwidth change, behaviour changes: high-> full colour, medium-> low colour, low-> gray scale
Middleware Adaptation (3)

- Application adaptation (application level)
  - Application can react to changes in context
  - It subscribes and is notified about context events
  - Application can choose how to change its logic in order to adapt
  - Eg, video streaming application can reduce frame transmission rate when bandwidth drops

System Overview

- MobiPADS server: located on the backbone
- MobiPADS client: located on the mobile device
- One server can support many clients
- Clients provide the services to applications for adaptation
Client/Server Interaction

System Framework (Server and Client)
The framework

- System components provide services:
  - Deployment, reconfiguration, management of services
- Service chain: mobilets are composed to offer aggregated services. Mobilets are pairs: one on server and one on client. They cooperate to provide a service.
- Composite events are used to monitor context
- Metaobjects reflect the configuration of the composite events and service chain and adaptation policies
- Middleware and application can inspect and reconfigure events and service chain

Events

- Mobilets and applications can register for events and be notified when something happens
- Possible Events:
  - CPU load level
  - RAM free space
  - Storage free space
  - Battery level
  - Client count
  - Network delay
  - Network error rate
  - Network disconnect
  - Network handoff
System Profiling

- Regulate adaptation through XML profiles
- Event
  \[ \text{HighCPUAvailable} = (\text{CPUclockrate}=400 \text{ and } \text{CPUloading} < 50\%) \text{ or } (\text{CPUclockrate}=200 \text{ and } \text{CPUloading} < 10\%) \text{ and } \text{Powerwarning}=false \]
  
- Events can be composed

Reconfiguration

- When change happens (event is triggered):
  
  - Adapt the service chain (the middleware)
  - Communicate the change to the mobilets which react internally
  - Communicate to application which reacts
Reconfiguration (2)

<table>
<thead>
<tr>
<th>MobiletService ChainMobile Application</th>
<th>Reconfiguration entity -&gt; Adaptation Initiator</th>
<th>Mobile Application</th>
<th>Service Chain</th>
<th>Mobilet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Application</td>
<td>Adjust its behaviour</td>
<td>Using profiles applications can specify service configurations in different contexts</td>
<td>Supplying parameters to mobilets</td>
<td></td>
</tr>
<tr>
<td>MobiPad System</td>
<td>Cannot reconfigure application</td>
<td>System profiles can be used to adjust the chain depending on context</td>
<td>System profiles can be used to specify behaviours of mobilets</td>
<td></td>
</tr>
<tr>
<td>Mobilet</td>
<td>Cannot affect application</td>
<td>Mobilets cannot affect the whole chain</td>
<td>Reaction of events</td>
<td></td>
</tr>
</tbody>
</table>

Reflection

- Three Metaobjects
  - Context: examination and modification of context info (permits subscription to events)
  - Configuration: shows configuration of service chain (allows adding and removing of mobilets)
  - Adaptation: shows adaptation policies
Examples

- System profile:
  - When bandwidth is low but battery is OK use a service chain with image transcoding and compression services
  - If an event of network disconnection is experienced the application will insert asynchronous data service in the service chain (to keep executing).

Evaluation

- Measured the overhead of reconfiguration
  - Reconfiguration time under different bandwidth for different mobilet sizes with and without active deployment of new mobilets
  - Overhead when active deployment is on is quite high!
Related Work

- Open ORB/COM (Lancaster)
- CARISMA/SATIN/Runes (UCL)
- Coda/Odyssey/AURA (CMU)
- GAIA (Illinois)