Enhancing JSR-179 for Positioning System Integration and Management

Paolo Bellavista
Antonio Corradi
Carlo Giannelli

DEIS, University of Bologna,
V.le Risorgimento n.2, 40136 Bologna Italy
{pbellavista, acorradi, cgiannelli}@deis.unibo.it
Agenda

- Location based services and positioning systems
- JSR-179 Location API for J2ME
- The Translucent approach
  - coupling transparency and visibility
- PoSIM middleware
  - integration and control
- JSR-179 and PoSIM API comparison
Positioning Systems

- Location Based Services (LBSs)
  - virtual museum assistance
  - service discovery

- Positioning systems
  - special purpose modules, e.g., GPS
  - communication purpose wireless technologies, e.g., IEEE 802.11 (Ekahau), Bluetooth (BTProximity), GSM/GPRS/UMTS

- Heterogeneity
  - location information: symbolic vs. physical
  - environment: indoor vs. outdoor
  - accuracy: few centimeters vs. several kilometers
  - power consumption: 1mW – 1W
  - additional features, e.g., location information as probability distribution function
Our Proposal

- Devices may contemporarily access several positioning systems which differ in provided information and capabilities

- A middleware solution to provide
  - an homogeneous access to positioning systems, while preserving their peculiarities
  - integration of available positioning systems
    - to perform location fusion
  - dynamically control integrated positioning systems
    - to switch among available ones depending on their availability and application requirements
JSR-179: Location API for J2ME
- Standardization effort
- Coarse-grained integration and management
- Deployment device: J2ME-based smart phones
- Positioning system leveraging technology: GPS
LocationProvider
- the component actually providing location
- provided at instantiation time

Criteria
- selected provider must satisfy particular criteria
  - required speed and altitude information
  - required minimum horizontal/vertical accuracy
  - required maximum power consumption
Provided information:
- location related: Location class
  - qualified coordinates (physical)
  - address info (symbolic)
- behavior related: state
  - available, out of service, temporarily not available

Location information delivery:
- on demand
  - getLastKnownLocation(), getLocation(timeout)
- periodic
  - setLocationListener(listener, interval, timeout, maxAge)
    - only one listener at a time
- event-driven
  - addProximityListener(listener, coordinates, proximityRadius)
    - the only one available triggering event
No dynamic and flexible management
- one location provider at a time
- LBSs have to monitor location provider performance
  - criteria considered only once, at instantiation time

Completely transparent
- no positioning system low level details at LBSs
- no positioning system control
The Translucent Approach

Differentiated visibility:

- **Transparent** (similar to JSR-179 approach)
  - useful for simple LBSs
  - integrated positioning systems perceived as a unique multi-behavior component

- **Visible**
  - underlying components low level details and management capabilities at application level
  - uniform access to underlying components for smart LBSs
PoSIM middleware

- Positioning System Integration and Management
  - based on translucent approach
    - Transparent: Policy and Data Managers
    - Visible: Positioning System Access Facility
PoSIM components (1)

- **Policy Manager (PM)**
  - pre-defined behaviors as policies, e.g., `POWER_USAGE_LOW`
  - transparent control API
    - declarative policy de/activation

- **Data Manager (DM)**
  - aggregated data delivery as an XML document
  - transparent delivery API
    - on demand: provides already available XML document
    - periodical: provides it at a time interval
    - event-driven: several triggering events
      - `atLocation`, `atChanges` and user defined ones
    - filtering rules: filter XML document and provides only LBS-relevant data
      - `highAccuracy` and user defined ones
PoSIM components (2)

- Positioning System Access Facility (PSAF)
  - positioning system integration in a plug-in fashion
  - underlying layers information access and behavior control
  - visible control and delivery API
    - register/cancel positioning systems
    - request for available information and manage capabilities

- Positioning System Wrapper (PSW)
  - uniform interface to interact homogeneously with positioning systems
PoSIM and JSR-179 APIs (1)

- Integration
  - JSR-179: one location provider at a time (Location class)
  - PoSIM: every available positioning system (XML document), integrated in a plug-in fashion

- Event-driven data delivery:
  - JSR-179: only proximity-based triggering event
  - PoSIM: proximity, movement and user defined events

- Transparent Management
  - JSR-179: criteria exploited only once, at instantiation time
  - PoSIM: criteria exploited to actively and dynamically control positioning system behavior

- Visible Management
  - JSR-179: only available, temporarily not available, out of service
  - PoSIM: uniform access to underlying systems
- equivalent
  - on demand and periodical delivery
- extended
  - event-driven delivery, underlying layers details and control
- additional
  - delivery filtering, dynamic integration

<table>
<thead>
<tr>
<th>API category</th>
<th>JSR-179 API</th>
<th>PoSIM API</th>
<th>PoSIM Component</th>
<th>Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>n.a.</td>
<td>get Instance(criteria)</td>
<td>get Instance()</td>
<td>PoSIM API</td>
<td>extended</td>
</tr>
<tr>
<td>transparent</td>
<td>getLastKnownLocation()</td>
<td>onDemand(listener)</td>
<td>DM</td>
<td>equivalent</td>
</tr>
<tr>
<td></td>
<td>addProximityListener(...)</td>
<td>addEvent(event, listener)</td>
<td>DM</td>
<td>extended</td>
</tr>
<tr>
<td></td>
<td>setLocationListener(...)</td>
<td>periodical(interval_listener)</td>
<td>DM</td>
<td>equivalent</td>
</tr>
<tr>
<td></td>
<td>n.a.</td>
<td>addFilter(filter, listener)</td>
<td>DM</td>
<td>additional</td>
</tr>
<tr>
<td>control</td>
<td>getInstance(criteria)</td>
<td>activateCriteria(criteria)</td>
<td>PM</td>
<td>extended</td>
</tr>
<tr>
<td>visible</td>
<td>n.a.</td>
<td>insertPosSys(newPosSys)</td>
<td>PSAF</td>
<td>additional</td>
</tr>
<tr>
<td></td>
<td>getState()</td>
<td>getFeatures(posSys)</td>
<td>PSAF</td>
<td>extended</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getFeature(posSys, aFeature)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>setLocation(posSys, aFeature)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions & Ongoing work

- **Management** of positioning systems **integrated dynamically** coupling both transparent and visible approaches, i.e., **Translucent** approach

- PoSIM greatly **extends** the JSR-179 capabilities, while mimicking its API to **facilitate adoption**

- wrapper for BTProximity (GPS and Ekahau PSWs already available)

- several pre-defined policies, filter rules, triggering events
Any question?

Acknowledgements:
- Work supported by MIUR FIRB WEBMINDS and CNR Strategic IS-MANET Projects

Web references for software and additional documents:
- http://lia.deis.unibo.it/Research/PoSIM/
- http://lia.deis.unibo.it/Staff/CarloGiannelli/