Orchestrator: Active Resource Orchestration Framework for PAN-Scale Sensor-rich Mobile Environment
**PAN-scale Sensor-rich Mobile Computing Platform**

- Upcoming pervasive computing platform
  - Dynamic mobile platform with diverse sensors (wearable/phone-embedded/space-embedded, ...)
- A number of personal context-aware applications will run on it.
  - Providing proactive, personalized, situation-aware services

**Wearable sensors**

**Space/object-embedded sensors**

- U-Secretary
- Location-based Services
- Health Monitoring
- U-Learning
- U-Trainer
- Diet diary
- U-Reminder
- Behavior correction

Location-based Services

Health Monitoring

U-Trainer

U-Secretary

U-Reminder

Diet diary

U-Learning

Behavior correction
Continuous Personal Context Monitoring

- A key building block for personal context-aware applications
- Involves complex, multi-step, continuous processing spanning multiple sensors/ mobile devices
  - E.g. “Running” → sensing in five 3D accelerometers, FFT processing, decision tree [Pervasive ‘04]
Challenges

- A number of context-aware applications share highly scarce resources of the PAN-scale mobile computing platform
  - Context monitoring requires complex and continuous processing over multiple devices
  - Many concurrent applications and scarce resources incur serious resource shortage/contention

- Significantly scarce resources
  - Limited battery power due to mobility
  - E.g. MicaZ Motes: 8MHz CPU, 4KB RAM, ~50Kbps Bandwidth
    - A light FFT library, kiss_fft, requires 40KB RAM, 10 MHz CPU

- Dynamic join/leave of heterogeneous sensors
  - E.g. take off a watch sensor, enter a space with sharable environmental sensors

- Dynamic changes in resource demands and status
  - Continuous changes in running applications and their requests
  - Sudden drops in bandwidth availability due to mobility, obstacles, ...
Applications themselves cannot solve the challenges

- **Context monitoring is complex.**
  - Burdensome programming and debugging
    - to implement complex feature extraction and recognition modules
  - Repetitive and time-consuming training process

- **Resource use/scheduling of the platform is complex.**
  - Need to aware resource demands to fit in the scarce resources
  - Need to handle dynamics in sensor availability and status

- **Hard to coordinate resource use with many other applications**
  - No adequate tools to communicate and negotiate with each other
  - Although possible, hard to expect and deal with every case
Objective

• An effective “resource coordination system”
  ▫ for PAN-scale dynamic distributed computing platform

Enable the platform to support a number of concurrent and long-running apps with highly scarce and dynamic resources
**Approach: Active Resource Use Orchestration**

**Active resource use orchestration**

- High-level context monitoring request
  - E.g. Context == Running

- System-wide holistic view of applications and resources

- Flexible system-driven resource binding

**Vs. Passive resource use management**

- Low-level resource allocation request
  - E.g. 5MHz CPU, 5KB RAM, 10kbps Bandwidth for a watch sensor

- Limited view on the resource requests

- Static app.-driven resource binding
## Proposed approach: Active resource use orchestration

<table>
<thead>
<tr>
<th>Active resource use orchestration</th>
<th>vs. Passive resource use management</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-level context specification/request</td>
<td>Low-level resource specification/request</td>
</tr>
<tr>
<td>activity==running, location==library</td>
<td>Acceleration sensor on wrist Memory: 3KB, BW: 20pkts/sec, Energy: 3J/min</td>
</tr>
<tr>
<td><strong>Flexible system-driven</strong> resource binding and scheduling</td>
<td><strong>Static application-driven</strong> resource binding</td>
</tr>
<tr>
<td>(Option 1) wrist-worn acceleration sensor (1KB, 3pkts/sec, 1J/min) (Option 2) waist-worn acceleration sensor (50B, 20pkts/sec, 4J/min)</td>
<td></td>
</tr>
<tr>
<td><strong>System-wide holistic view</strong> on applications and resources</td>
<td>Limited understanding(or view) on the resource requests</td>
</tr>
</tbody>
</table>
Advantages

• From the platform’s perspective
  ▫ Efficient resource utilization
    • Exploit overall resource capacity with balanced utilization (avoiding skewed/unbalanced resource use)
    • Host more applications with scarce resources

• From the applications’ perspective
  ▫ Provide longer lasting services to its mobile users even with highly scarce resources
  ▫ Provide seamless and stable services even under highly dynamic resource situation
Related work (Environment)

• No previous work provides a active orchestration system
  ▫ For multiple context-aware applications
  ▫ Over the whole PAN-scale sensor-rich mobile platform

• Resource management systems in a piecemeal, per-device fashion
  ▫ For a mobile device (mostly laptop)
    • Odyssey [SOSP’ 99], Chameleon [TMC ’08], ECOSystem [ASPLOS ’02]
  ▫ For a sensor
    • Pixie [Sensys ’08], EON [Sensys ’07], Level [Sensys ’07]
  ▫ These systems are difficult to resolve
    • resource contention among multiple applications
    • skewed resource utilization

• Systems for sensor network ?
  ▫ It is mainly considered as a data collection platform
    • Focus on the MAC/routing algorithms/methods to reduce communication cost
  ▫ Recently peloton is proposed for resource management
    • But, simple passive resource use management
## Related work

- No work provides a resource use orchestration framework for PAN-scale dynamic distributed computing platform

- Work for resource management in mobile computing, sensor network

<table>
<thead>
<tr>
<th>Area</th>
<th>Apps</th>
<th>Target device</th>
<th>Resource coordination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile system</td>
<td>Odyssey [SOSP’99]: Energy-aware adaptation of apps</td>
<td>Single mobile device</td>
<td>Application-driven, Priority-based</td>
</tr>
<tr>
<td></td>
<td>Chameleon [TMC’08]: Application-level DVFS for energy management</td>
<td>Single mobile device</td>
<td>Application-driven, Isolation</td>
</tr>
<tr>
<td></td>
<td>ECOSystem [ASPLOS’02]: System for energy management to meet lifetime goal</td>
<td>Single mobile device</td>
<td>System-driven energy allocation, Static proportional share</td>
</tr>
<tr>
<td>Sensor network/system</td>
<td>Pixie [SenSys’08]: Sensor OS for resource-aware application</td>
<td>Single sensor node</td>
<td>Application-driven, Priority-based (for app sub-tasks)</td>
</tr>
<tr>
<td></td>
<td>Peloton [HotOS’09]: Distributed sensor OS</td>
<td>Multiple sensor nodes</td>
<td>Based on Pixie Conceptual architecture proposed, yet to build a system</td>
</tr>
<tr>
<td></td>
<td>[SenSys’04], [Infocom’02], [HICSS’00]: Energy-efficient MAC/routing protocols</td>
<td>Multiple sensor nodes</td>
<td>Not a system to coordinate sensor network resources as a whole</td>
</tr>
</tbody>
</table>
## Related work (Approach)

<table>
<thead>
<tr>
<th>Multiple Application</th>
<th>Passive Management</th>
<th>System-aided Static Management</th>
<th>Active Orchestration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Odyssey [SOSP ’99]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chameleon [TMC’08]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECOSystem [ASPLOS ’02]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EON [Sensys ‘07]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pixie [Sensys ‘08]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peloton [HotOS ‘09]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peloton [HotOS ‘09]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Orchestrator**

**Mobile Systems**
- Sensor Systems
- Pan-scale system
Architecture design
Key functional components (operation flow)

Applications

API

Request

Context

Translation

Plan candidates

Selection

Plan for execution

Resources

Runtime Adaptation

Monitoring
High-level specification of context

- **API**
  - Int RegisterQuery (string queryString);
    - Returns query id
  - Bool DeregisterQuery (int queryID);

- **E.g.**
  RegisterQuery(“CONTEXT Activity == running ACCURACY 90% DURATION 7 days”)

- Free application developers from taking care of processing method and corresponding resource allocation (scheduling)
Dynamic query translation

Context monitoring query

Dynamic translation

Monitoring dynamically joining/leaving sensors, resource status

Running

Context specified in a query

Multiple processing plans (used processing modules and sensors, task distribution)

Available sensor pool

Sensor-side

Mobile-side

Accelerometer attached to sleeve

Frequency domain feature extractor

Decision Tree

Accel_wrist
Window 128
Sensing 50Hz

Space-embedded light sensor

U-Trainer

U-Secretary

U-Reminder

Diet diary

U-Learning

Behavior correction

Location-based Services

Health Monitoring
Two-phase translation

- Translate a context into processing plans
  - Each plan is mapped with different hardware resources
- Generate a variety of plans by combining different processing modules and sensors

**Logical Translation**
- Processing module decision
  - Sensing
  - Feature extraction
  - Context recognition

**Physical Translation**
- Hardware mappings
  - Combination of sensors
  - Distribution of processing modules

---

**Context**

**Logical Processing Plan (LPlan)**

**Running**

**LPlan**

**Physical Processing Plan (PPlan)**

**PPlan**

\( \ldots \)