A Framework for Real-Time Context Provision in Ubiquitous Sensing Environments

Adel Shaeib, Paolo Cappellari, Mark Roantree

Interoperable Systems Group
Dublin City University
Dublin, Ireland
http://www.computing.dcu.ie/~isg/

June 22th, 2010
Introduction

Contribution

Data Processing

Experiments

Conclusion
Context Provisioning in Sensor Networks

Sensor Networks Today

- Sensors and sensor networks are widespread
- Information from sensors can be combined with the context in which it is generated

Challenges

- Enrich sensor data
- Uniform access to sensor, context, mixed information
- Ease of context configuration
- Support Real-time applications
Scenario & Contribution

**Smart Building: monitor the environment within a building**
- Buildings are generally organized in *spaces*, each with *contextual information*
- Spaces and Contextual information reorganized frequently
- Queries on streaming, contextual, mixed data
- The volume of data and queries can be high

**UbiQuSE: Ubiquitous Queries for Sensing Environments**
- An hybrid query interface to manage streaming, contextual and data mining queries
- Simplified context configuration
- Small footprint for real-time applications
Query Classification

- **Live (L):** expressed on live data streams
- **Context (C):** expressed on the static context data or on a mix of static and live data
- **Mining (M):** expressed on stores of historical data

<table>
<thead>
<tr>
<th>Type</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>What is X’s Current Location?</td>
</tr>
<tr>
<td>L</td>
<td>How many “people” are in this location?</td>
</tr>
<tr>
<td>C</td>
<td>What seervices are available in Space Z and adjacents?</td>
</tr>
<tr>
<td>C</td>
<td>Given current direction, what is next Space?</td>
</tr>
<tr>
<td>M</td>
<td>How long has X been in this Space?</td>
</tr>
<tr>
<td>M</td>
<td>What is the most popular service for this Space?</td>
</tr>
</tbody>
</table>
Architecture
Stream Processing

Data formatting

- Raw data is translated into structured, XML, data format
- A *template* describe the association between raw data and its semantic
  - Each value is associated with an XML attribute
- Raw data are “wrapped” with XML tags

### Raw data format

<table>
<thead>
<tr>
<th>Description</th>
<th>Sensed Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>16/02/2010</td>
</tr>
<tr>
<td>Time of day</td>
<td>15:27:46</td>
</tr>
<tr>
<td>Dimension-X</td>
<td>19.0431594848633</td>
</tr>
<tr>
<td>Dimension-Y</td>
<td>0.91026896238327</td>
</tr>
<tr>
<td>Dimension-Z</td>
<td>1.21332836151123</td>
</tr>
</tbody>
</table>

### Structured data format

```xml
<event_detect tagId="20000007106">
  <session id="1">
    <date>16/02/2010</date>
    <time>15:27:46</time>
    <x>19.0431</x>
    <y>0.9102</y>
    <z>1.21332</z>
  </session>
</event_detect>
```
Querying live data

Live data

- Live data can be queried by XPath/XQuery expressions
  - Consolidated languages
  - Declarative expressions
  - Device independent
- The overhead of XML-izing live data is rather small because each event produces a small volume of data
- The overhead of executing XPath/XQuery queries remains low as the XML structure of live data is simple (generally one or two levels of nesting)
Querying context data

Static contextual data

- Static context information can be retrieved by static queries
  - Configuration of the spaces, services available, …
- Static information is generally well structured and can be stored in a relational database

Dynamic contextual data

- Dynamic context queries retrieve mixed live and contextual information
- Live information from a device feed arguments to (static context) queries to retrieve contextual information associated with current device information
Query Repository

Predefining queries

- Queries are declarative: can be stored, added or edited without altering the application source code
- Both static and dynamic queries can be predefined and stored in a query repository
- In effect, users can interact with the system by choosing from available queries
Stream Processing: historic series

Data enriching

- Simplicistic sensor data is enriched with contextual information
- Enriched data is stored in a data warehouse

Structured Data

```
<event_detect tagId="2000007106">
  <session id="1">
    <date>16/02/2010</date>
    <time>15:27:46</time>
    <x>19.0431</x>
    <y>0.9102</y>
    <z>1.21332</z>
  </session>
</event_detect>
```

Enriched Data

```
<event_detect tagId="20000007106">
  <session id="1">
    <date>30/11/2009</date>
    <time>15:27:46</time>
    <x>19.0431</x>
    <y>0.9102</y>
    <z>1.21332</z>
    <zoneID>Zone 3-2</zoneID>
    <productlist>
      <productID>P015</productID>
    </productlist>
  </session>
</event_detect>
```
Experiments

Setting

- In-lab reproduction of business partner scenario
- 4 sensors per room, 4 mobile tags devices moving around
- System in a “push style” mode, where information is automatically and continuously passed to the user

Accuracy

- For (relatively) slow moving objects we measured

<table>
<thead>
<tr>
<th>Area (cm)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 20</td>
<td>60</td>
</tr>
<tr>
<td>20 - 100</td>
<td>100</td>
</tr>
</tbody>
</table>
Experiments

Query performance: live queries

- Queries retrieving basic information (location, time, device-id) have execution time of \( \sim 205 \text{ms} \), which is in the same order of the sensors’ specs
- This demonstrate the little overhead the XML conversion and query adds to live data processing

Query performance: context queries

- The execution requires to retrieve live information to feed arguments to a query on the context database
- On relatively complex queries like “describe the information on the current and adjacent spaces” have an execution time of \( \sim 26 \text{ms} \), on top of the time to retrieve live data
Conclusions

- We have proposed a framework for real-time context provisioning in ubiquitous sensing environments.
- Data can be accessed by standard query languages.
  - Device independence.
- Data is decoupled from the application logic.

Future Work

- Scaling up to larger scenarios
  - Number of users, spaces.
- Exploit the historical series in the data warehouse to analyze and, possibly, obtain predictions.
  - Users behaviour, context evolution, sensors activity, ...
Thank you!

Questions

- “Real-time...”
- Or to paolo.cappellari@computing.dcu.ie

More details

- ISG web site: http://www.computing.dcu.ie/~isg/