On Uncertainty in Context-Aware Computing: Appealing to High-Level and Same-Level Context for Low-Level Context Verification

Amir Padovitz, Seng Wai Loke, Arkady Zaslavsky
School of Computer Science and Software Engineering, Monash University, Australia

Presented by Dr C.Ling
On Context Uncertainty

Four factors related to context uncertainty:

• Insufficient data to infer context, cost efficiency
• Context ambiguity
• Unknown contextual situation
• Inherent inaccuracy of sensors
Low-level context

Sensorial information is often:

• Inaccurate and unreliable

Hence:

• Contradicting
• Ambiguous when inferred for higher level context

Sensors are black boxes, therefore additional verification needs to be applied!
Low-level context verification

The Goal:
Verify correctness of sensor reading and resolve higher-level ambiguous situations

The Approach:
• Resort to other elements in the system.
• Make use of existing reasoning techniques.
• Reuse generically for various context scenarios.
Logical Context Verification

The Method:

- resort to higher level contextual situations.
- Iterate on possible error values of the filtered attribute, and
- switch assumptions regarding the correctness of the verified context state
- determine the most probable state in regard to higher-level context
A General Approach

• Differentiate between low-level context: e.g. ‘temperature’, ‘location’, ‘light status’ and high-level, more abstract context: e.g. ‘In a meeting’, ‘Sleeping’

• Appeal to Higher-level context and compute relatedness of sensor readings to various situations, considering probable inherent errors.

• Evaluate approach for filtering inaccuracies of in-door positioning mechanisms.
Location filtering evaluation example

**Basic experiment floor plan**
Verification Procedure

1. if sensor readings and context-state correspond to same context-space (current active context) then:
   ➔ Return sensor readings.
2. Adjust location parameters within acceptable error boundaries:
   2.1 if location found in current active context and if passed probability distance test then:
      ➔ Return adjusted location, which has minimal distance to the original sensor readings.
   2.2 if location found in other context spaces but not in current active context-space:
      A ← location with minimal distance to any of the context spaces.
      // reached here if no current active context found in acceptable error distance
3. if original sensor readings are not in a valid location, then force change in location, by
   ➔ Return minimal distance to any valid area.
   // reached here if original location is in a valid area
4 ➔ Return A.
Analysis of filtering procedure

Typical scenario illustration
Typical Experimental Runs

### logical filtering scenarios

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1.png" alt="Office Diagram 1" /></td>
<td><img src="image2.png" alt="Office Diagram 1" /></td>
<td><img src="image3.png" alt="Office Diagram 1" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="image4.png" alt="Office Diagram 2" /></td>
<td><img src="image5.png" alt="Office Diagram 2" /></td>
<td><img src="image6.png" alt="Office Diagram 2" /></td>
</tr>
</tbody>
</table>
Reduction in error after filtering

Filtering errors vs. sensor reading errors

Accumulated 100 runs
Critical Analysis

Increase in error due to logical filtering
Conclusion

• In general, good results for reducing inherent sensor inaccuracies.

• Assists in better reasoning about the correct higher-level context and resolves ambiguity.

• Generic approach that is scalable to more elaborate settings.

• However, highly dependent on initial settings of higher-level contextual situations. (needs related situations that make use of the verified attribute)