MADAM
Mobility and Adaptation enAbling Middleware

Svein Hallsteinsen
SINTEF
Background

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qua (NF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Famous (NF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Families (EU)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Madam (EU-IST)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Osiris (EU)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Music (EU-IST)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partners:
- 2 Univ. +1
- 2 Indep res inst +1
- 3 SMEs -1 + 3 SMEs
- 1 Big company +2 (big Telcom) +1 (Public transport)
Motivation

• Computing is going mobile, ubiquitous, service oriented

• Mobile use means dynamic variation in user needs and available computing and communication resources

• Applications must adapt to such changes in order to sustain availability, usability and usefulness
Madam Objectives

• Provide support for the development of applications that adapt dynamically to changes in context (at launch time and during use)
- Application reference architecture

- Notation extensions
  (UML profile)
- Modeling tool
- Transformation tool

-Middleware
- context monitoring
- context reasoning
- adaptation reasoning & decision making
- (re)configuration
- application launch and initial adaptation

Brussels September 27, 2005
Properties and utility

- Properties
  - Dynamically changing env.
  - Influence
  - Describe
  - Needs
  - Offer

- Service
  - Mobile user
  - Provides service to
  - Offers

- Execution env
  - Executes in

- Utility
  - Brussels September 27, 2005
Conceptual model

![Diagram of conceptual model with relationships between entities such as Interface, Port type, Entity type, Property def, Port, Entity, Property annotation, User environment entity, User, Software component, Resource, Influence, Use, Connected to, Has, Implements, and Various arrows indicating relationships.]
Properties example

Property vocabulary

<table>
<thead>
<tr>
<th>Name</th>
<th>Value range</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsp</td>
<td>1:100</td>
<td>Response time</td>
</tr>
<tr>
<td>mem</td>
<td>1:100</td>
<td>Amount of memory</td>
</tr>
<tr>
<td>nbw</td>
<td>1:100</td>
<td>Bandwidth</td>
</tr>
<tr>
<td>haf</td>
<td>yes, no</td>
<td>Handsfree operation</td>
</tr>
</tbody>
</table>

utility = ( ( if usr.rsp >= app.rsp then 1 else 1 - (app.rsp-usr.rsp)/app.rsp) + ( if ((usr.haf and app.haf) or (!usr.haf and !app.haf)) then 1 else 0) ) / 2
Adaptable services through component frameworks

\[
\text{mem} = \text{Ui.mem} + \text{Ctrl.mem} + \text{Db.mem}
\]

\[
\text{nbw} = \text{Db.nbw}
\]

\[
\text{rsp} = \text{Db.rsp}
\]

\[
\text{haf} = \text{Ui.haf}
\]
Multiple applications

- Common vocabulary of properties
  - use OMG std for QoS and FT
- Combined utility function
  - weighted sum or product of differences
  - Weights represent user priorities
Mw architecture - overview

1. Context manager
2. Adaptation manager
3. Configurator

- Context change
- Reconfigure to selected variant
- Find implementations, derive and evaluate variants

Core
Mw architecture - plugins
Experimentation

• Prototype middleware implementation (Java CDC/Personal)
• 2 Pilot services developed
• Tested on iPAQ pocket pc with Windows mobile and Creme JVM
• Useful adaptation demonstrated
• Runtime overhead acceptable (Mw footprint < 10% of program memory, adaptation time < 1sec)
Challenges

- Modeling variability is not trivial
  - Leave to experts?
  - Learn from product line community
  - Tool support for testing and tuning
  - Need not be perfect

- Runtime overhead / Scalability
  - Adaptation time grows exponentially with the number of variation points
  - Pre-runtime processing
  - Background processing
Challenges

• Context sensing & reasoning
  – How to detect significant context changes?
  – Region maps?
• User in the loop
  – How much can we bother the user asking for input?
  – To which extent does the user want to be involved?
• Distribution
  – Centralized vs. decentralized control
  – Focusing on client side
Thank you!

Questions?

More information:
www.ist-madam.org