

MADAM

Mobility and Adaptation enAbling Middleware

Svein Hallsteinsen
SINTEF



Background

2003 2004 2005 2006 2007 2008 2009 2010

Qua (NF)-----

Famous (NF)-----

Families(EU)

Madam (EU-IST)---

Osiris(EU)-----

Music(EU-IST)-----

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)



Approach

- Application reference architecture

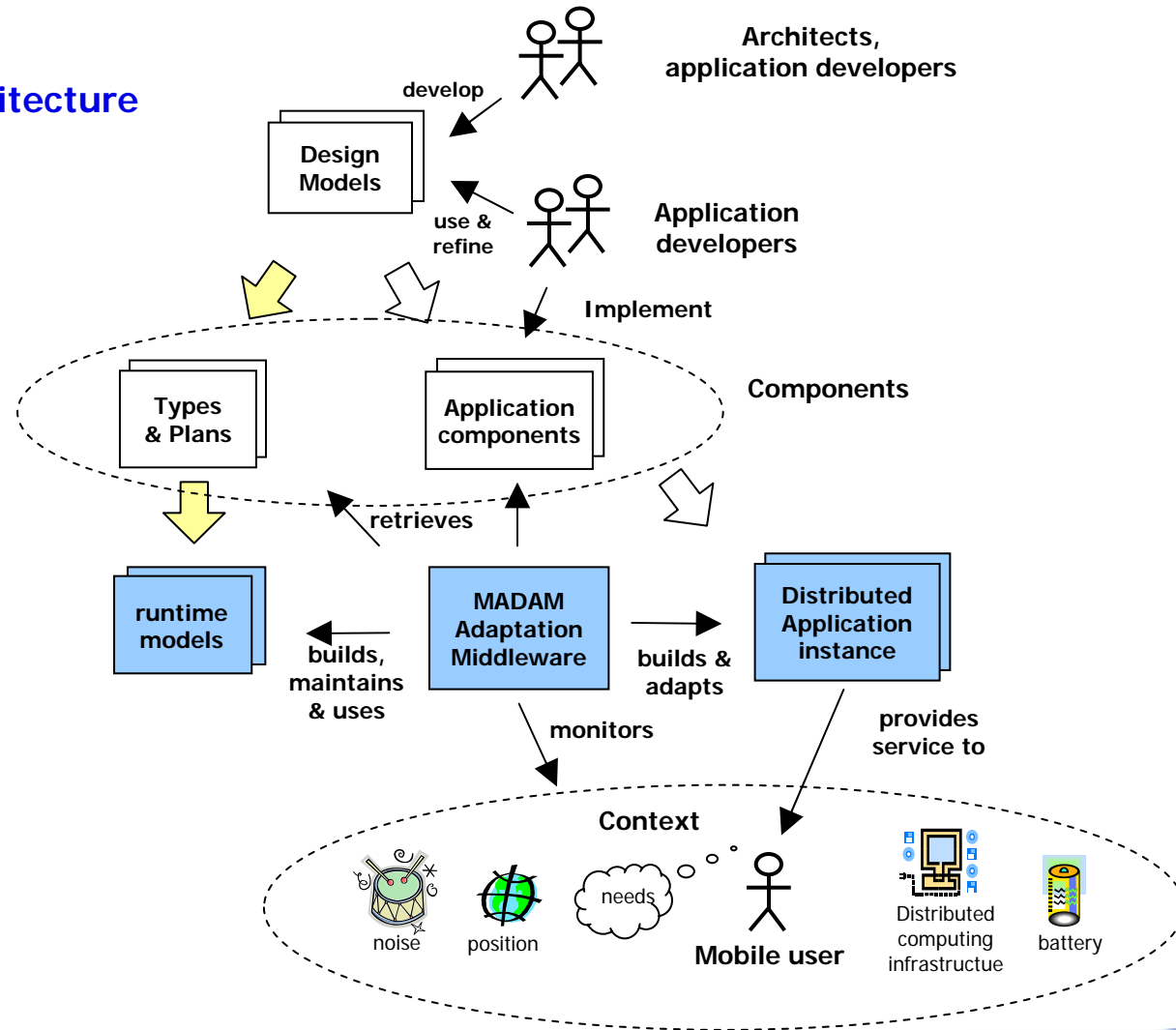
- Notation extensions
(UML profile)

- Modeling tool

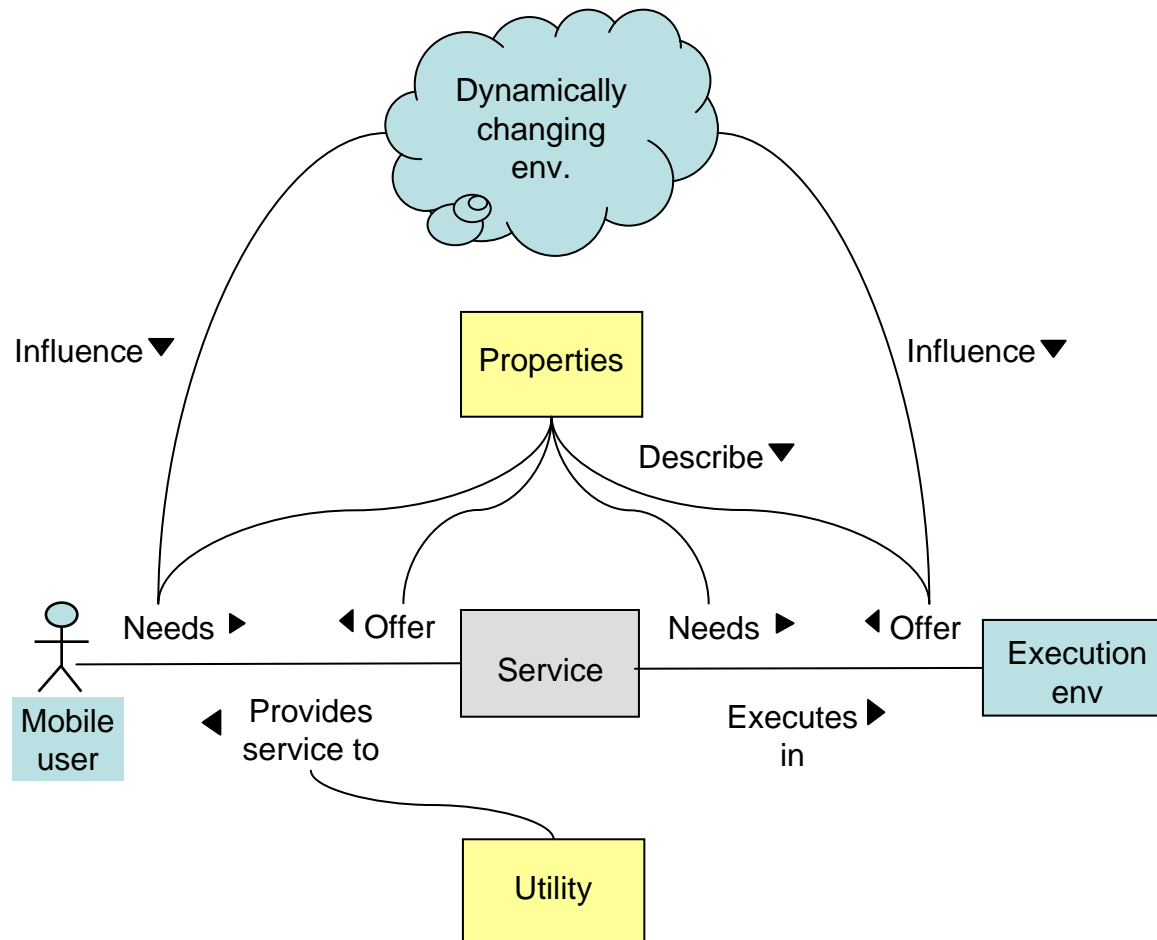
- Transformation tool

-Middleware

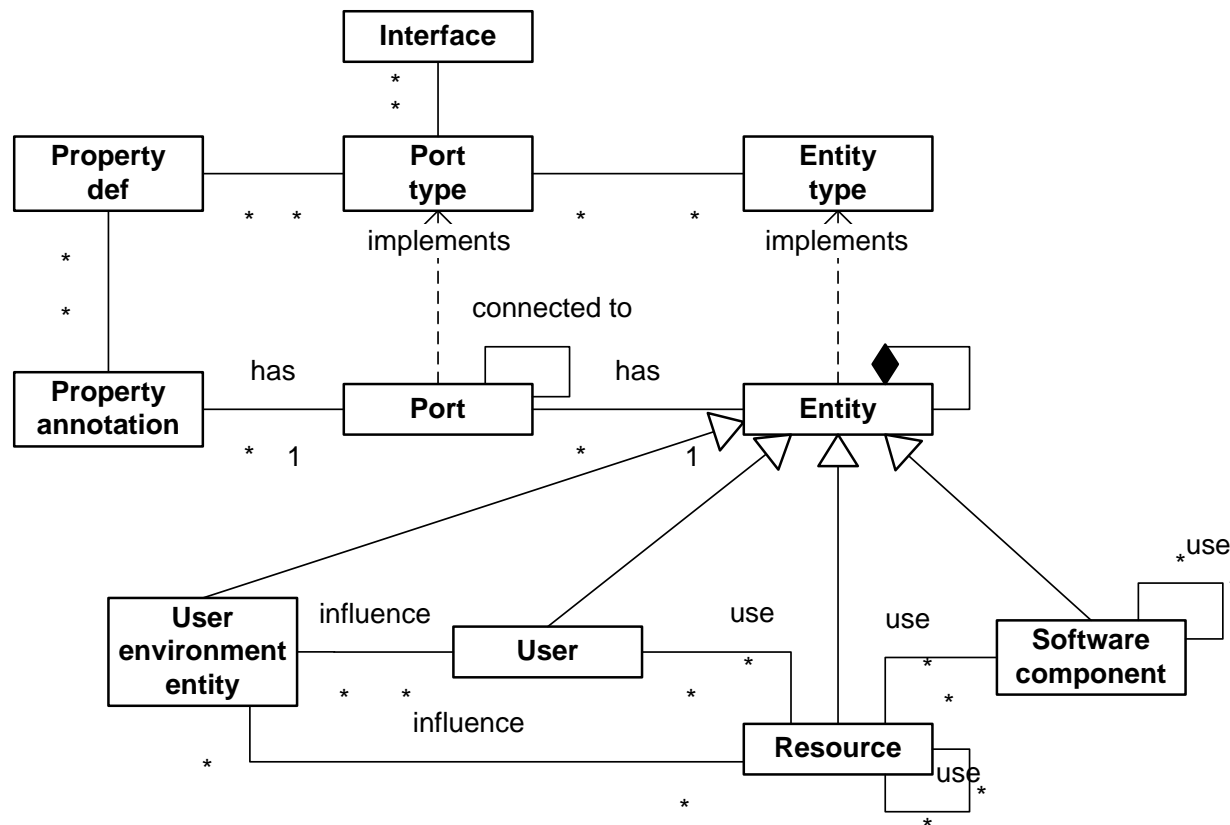
- context monitoring
- context reasoning
- adaptation reasoning & decision making
- (re)configuration
- application launch and initial adaptation



Properties and utility



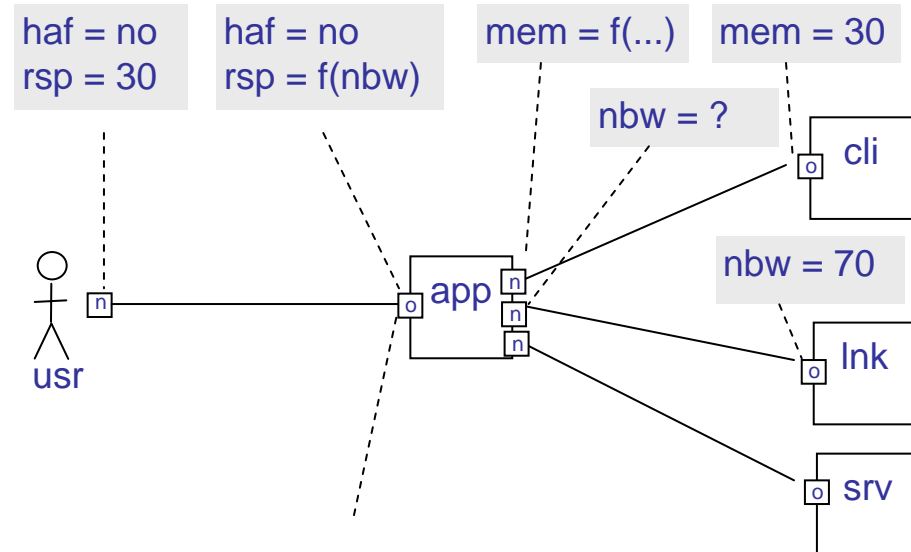
Conceptual model



Properties example

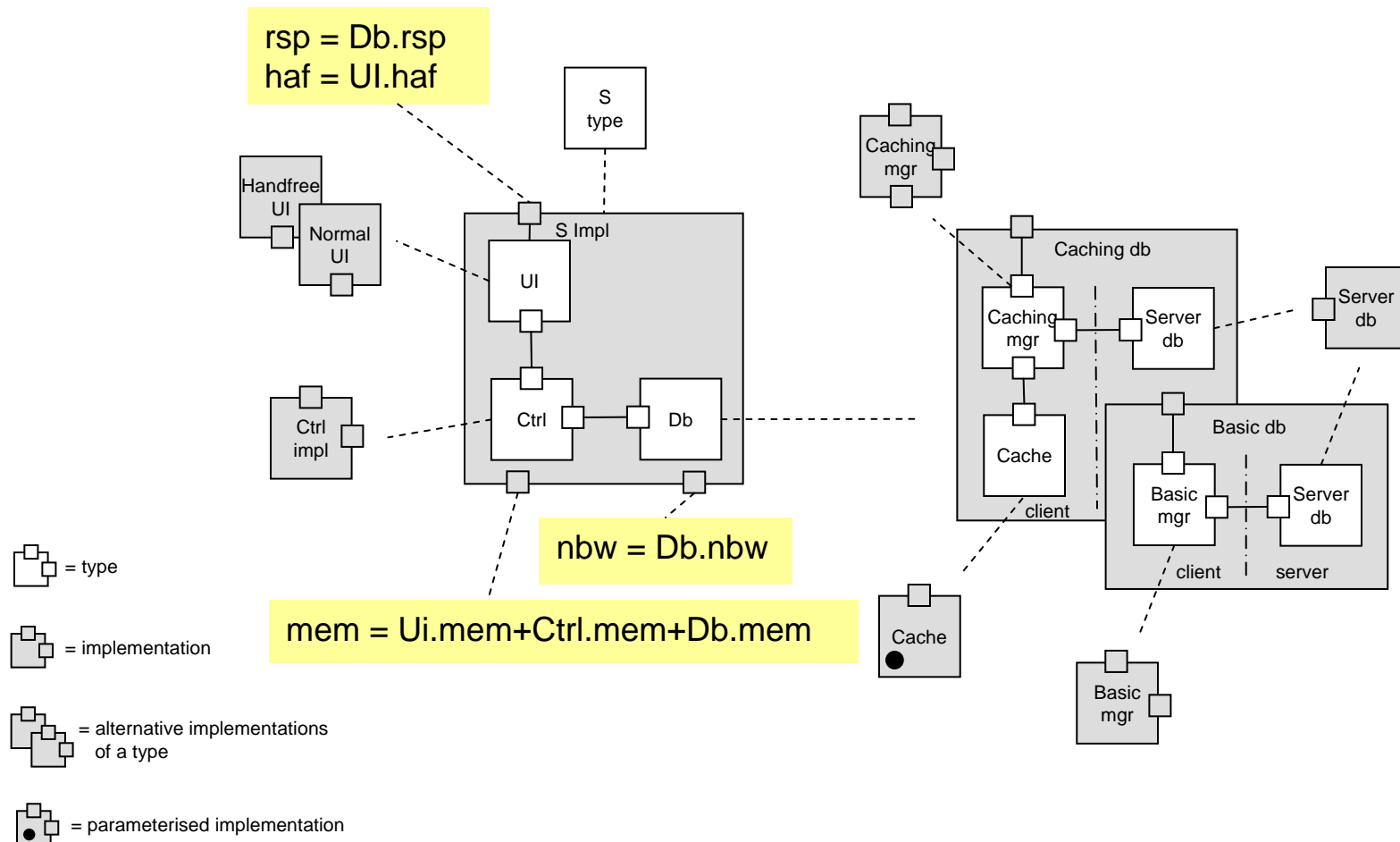
Property vocabulary

Name	Value range	Explanation
rsp	1:100	Response time
mem	1:100	Amount of memory
nbw	1:100	Bandwidth
haf	yes, no	Handsfree operation



$$\begin{aligned}
 \text{utility} = & ((\text{if } \text{usr.rsp} \geq \text{app.rsp} \text{ then } 1 \\
 & \text{else } 1 - (\text{app.rsp} - \text{usr.rsp}) / \text{app.rsp}) \\
 & + (\text{if } ((\text{usr.haf} \text{ and } \text{app.haf}) \\
 & \text{or } (!\text{usr.haf} \text{ and } !\text{app.haf})) \text{ then } 1 \\
 & \text{else } 0) \\
 &) / 2
 \end{aligned}$$

Adaptable services through component frameworks

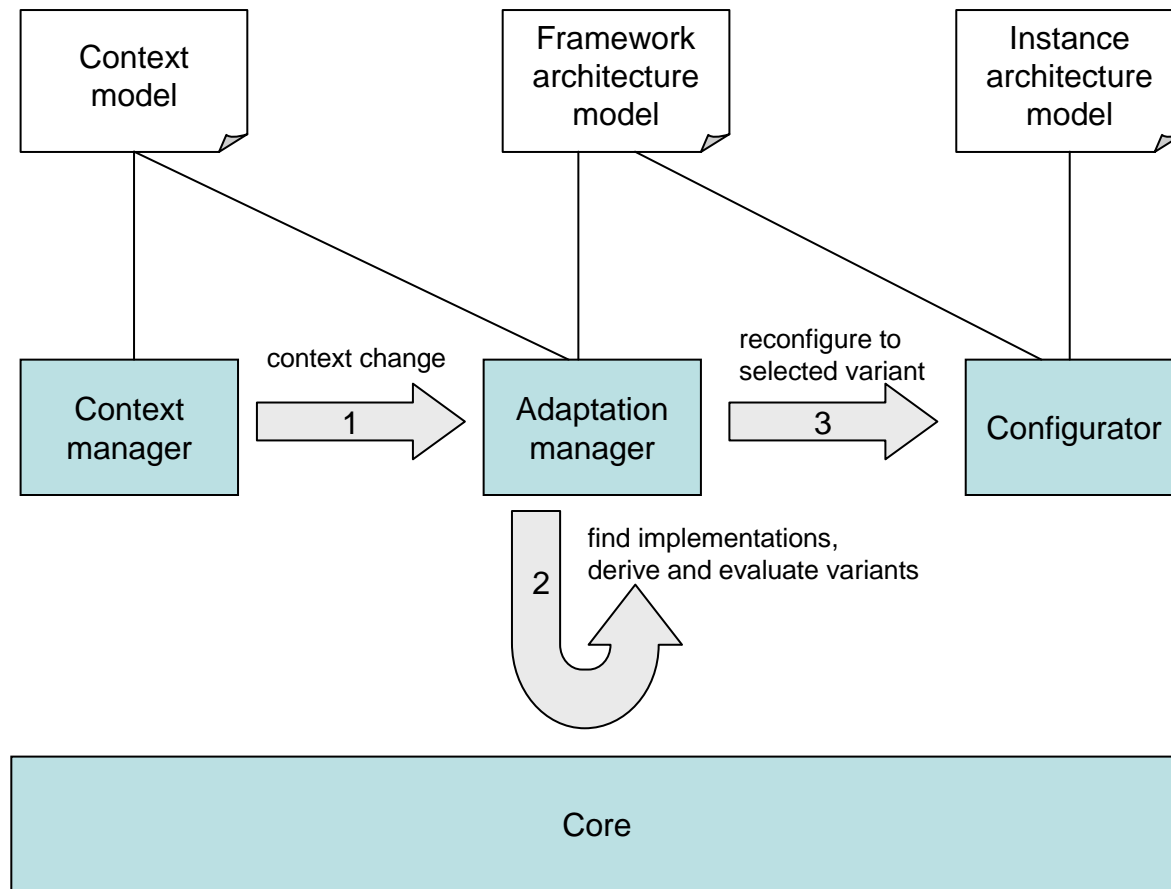


Multiple applications

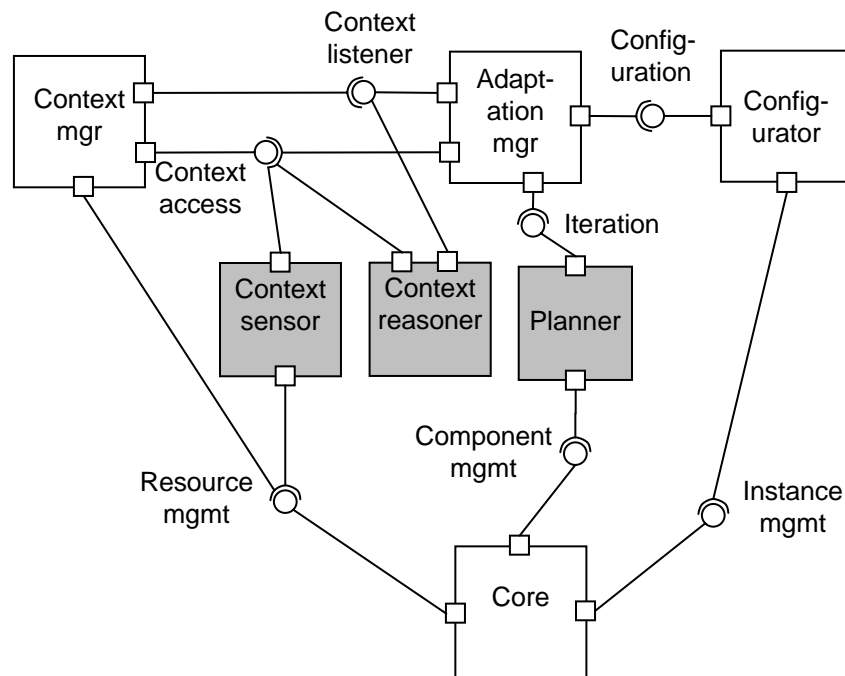
- Common vocabulary of properties
 - use OMG std for QoS and FT
- Combined utility function
 - weighted sum or product of differences
 - Weights represent user priorities



Mw architecture - overview



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 no 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

