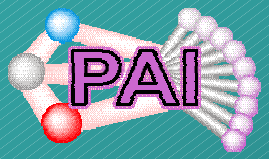


Providing Service in a Changing Ubiquitous Computing Environment

Sergio Maffioletti, Simon Schubiger, Béat Hirsbrunner

DIUF-PAI

Workshop on Infrastructure for Smart Devices-
How to Make Ubiquity an Actuality

HUC 2000, Bristol

University of Fribourg, Switzerland

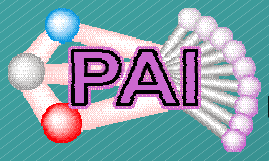
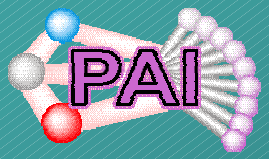


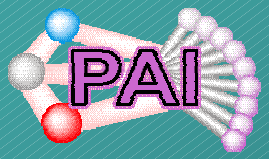
Table of Content

- Introduction and Goals
- Ubiquitous interactive device (UbiDev)
- Service description (ontology & classifiers)
- Conclusion



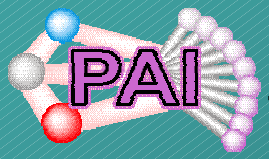
Observations

- Current UbiComp projects often focus on hardwired applications
- Today's middleware rely on standardized service descriptions for interoperation
- Today's UbiComp applications are not conceived to interoperate with each other



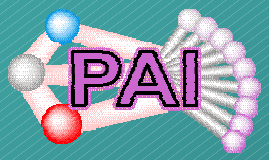
Goal

- Define a common design methodology for UbiComp Applications
- Find abstractions for UbiComp applications (a middleware)
- Find a standard independent and machine adapted way to identify services



Main Points

- UbiDev
  - Ubiquitous computing device abstraction
- Service description
  - Ontologies
    - Application’s conceptual view of the world
  - Classifiers
    - Bridge between application concepts and real world

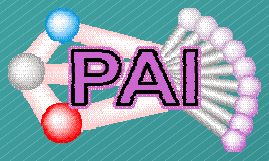


Ubiquitous Interactive Device

- UbiDev is a UbiComp device abstraction
  - Perception devices
    - GPS, Temperature sensor, Light sensor
  - Interactive devices
    - PDA, PC, SmartBoard
- Application is a composition of cooperating services

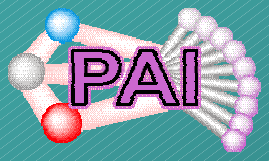
UbiDev

- Layered middleware
- Define an interaction scheme in an interactive environment
- Applications development rely on first class abstraction



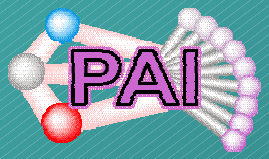
UbiDev

- Application layer
  - Composition of services, user interaction
- Service layer
  - Service loading / unloading, service linking (cooperation), resource management
- Federation management layer
  - Authentication, Session management



UbiDev

- Data management layer
  - Communication protocols, protocol constraints, addressing
- Medium layer
  - Low level communication capabilities (physical)

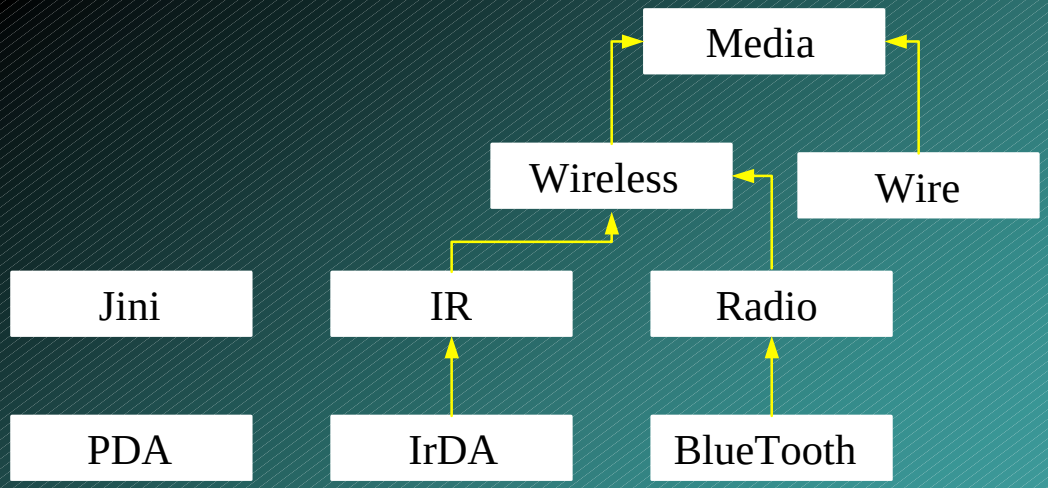


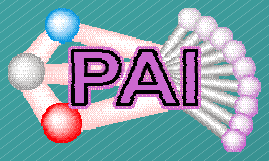
Service Description

- An application should only have to care about concepts it relies on
- An application should be able to provide concept semantics (e.g. “Nearest printer”)
- Middleware should provide or search concept instances (services)

Ontology

- Application's view of the world in terms of interrelated concepts
  - Organizational unit
  - Captures a problem domain



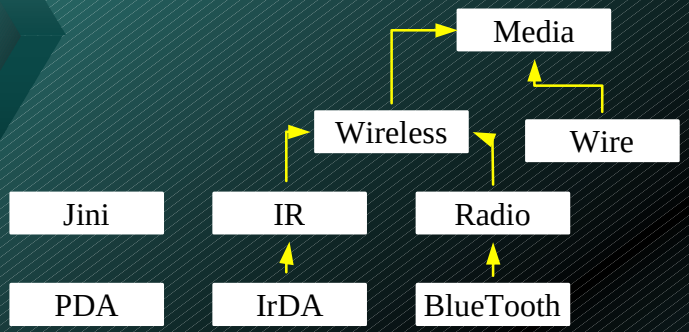


Classifier

- Associate concepts with real-world entities (services)
- Implementation of concept semantics
- Allows “Addressing by Concept”

Classifiers

Input Ontology



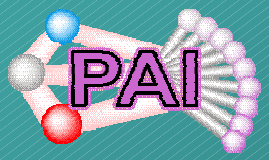
Classifier

IrDA

Output Concepts

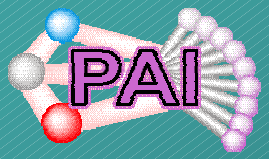
- **Media**
- **Wireless**
- **IR**
- **IrDA**

UbiDev



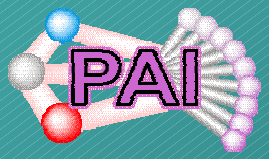
Addressing by Concept

- Application address resources by concept instead of addressing them directly by an URL, remote reference or an ID
- Concept implementation are selected on the fly and may even change during runtime (e.g. “Nearest Printer”)



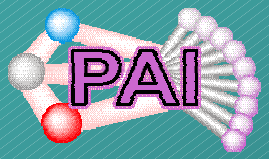
Conclusion

- Goal: Supply a design methodology for UbiComp Applications
  - Abstractions
    - UbiDev layers
  - Service description & composition
    - Ontologies & Classifiers
  - Self-management
    - Federations



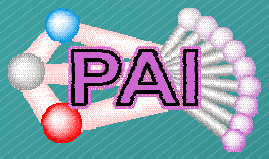
Conclusion

- Goal: Define an abstractions for UbiComp applications
  - UbiDev layers provide service structuring
  - UbiDev layers define a minimal set of services and interactions



Conclusion

- Goal: Find a machine adapted way to identify services and resources
  - Application can express their requirement and view of the world by ontologies
  - Classifiers encapsulate as much semantics as needed
  - Classifiers can be tailored to the applications needs



Outlook

- Currently
  - Experimenting with ontologies and classifiers
  - Working on UbiDev model
- Future
  - Implementation of middleware
  - Simulation environment
  - Evaluation of existing infrastructures (e.g. E-speak)