UbiquiTalk

An Infrastructure for Ubiquitous Computing

Noury Bouraqadi and Michaël Piel

http://csl.ensm-douai.fr/research

Ecole des Mines de Douai

Context: Ubiquitous Computing

- Ever smaller computers (nanotechnologies)

624 MHz
128 Mo RAM
7,7 x 13,5 x 1,5 cm
200 g
…

55 MHz
8 Mo RAM
3,5 x 1,9 x 1,9 cm
18 g
…

- "computers" everywhere and often invisible
  - Cars (~20 micro), washing machines, fridges, clothes, …
- Network connections (wireless) everywhere
  - GPRS, UMTS, Wifi, WiMax…

Context: Ubiquitous Computing

- Many devices per person (phone, PDA, …)
  - Access services (software + data) from different devices
  - Use many devices possibly simultaneously
    - Connection, "Synchronization" = Data replication, …
- Mobility of both users and devices
  - Only some devices move with their owners
    - Often Small => with little resources (memory, battery…)
  - Remote access to services (software + data)
    - Exchanging data, collaborative work, …
  - Varying environment
    - Network type and quality, peripherals, …

Consequences: Building Software even more complex

- Multiple varying parameters to take care of
  - Unpredictable Hardware resources/capabilities
  - Unpredictable network characteristics
  - Space/Time environment changes
- Negative impact on software projects
  - Production delays (time to market)
  - Higher production costs
  - Need of experts (scarce and expensive)
  - Decreasing reliability of produced software

Goals of UbiquiTalk

- Help developers build distributed software
  - Framework for development
  - Middleware for automatic deployment
- Minimize the administration tasks
  - Zero networking configuration

Few Assumptions made by UbiquiTalk

- Unanticipated Remote Interactions
  - Open/dynamic set of devices
    - Devices may join and leave the network dynamically
  - Open/dynamic set of software used remotely
    - Softwares may be added and suppressed at run-time
- Any Network Setting
  - Ad hoc, private LAN, Internet, …
  - Wifi, Bluetooth, Ethernet, …
- Heterogeneous hosts
  - Different software/hardware resources (e.g. display, printer)
  - Different amount of resources (e.g. RAM, energy)

Outline

- Motivation
- Overview
- User Interface
- Applications
- Conclusion

An Infrastructure for Ubiquitous Comp.

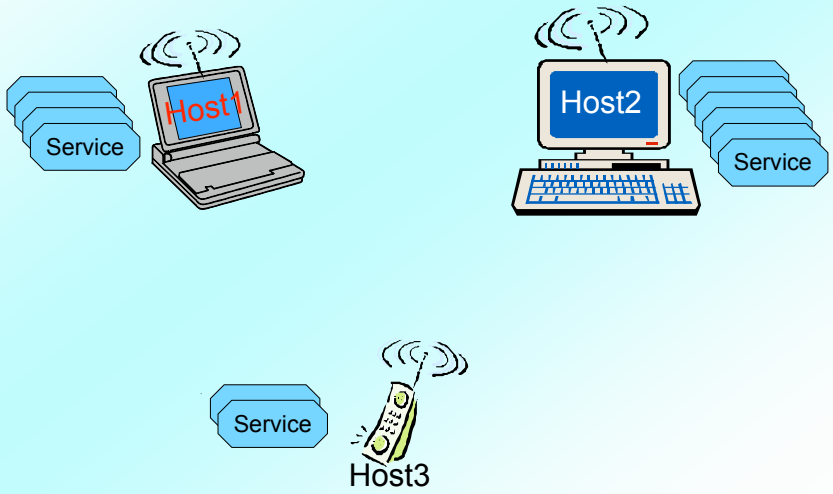
- Infrastructure =
  - Middleware for distribution
  - Framework
    - Domain objects
    - User Interface
- Ubiquitous Computing
  - Unanticipated remote interactions
  - Heterogeneous hosts

Two Basic Concepts

- Host: Any device with computation capabilities and a network interface
- Service: Any object in a host that can be accessed remotely
  - Application service: An application object
  - Middleware service: A middleware object that supports host activities (i.e. other services)
    - Host Discovery, Services registry, …

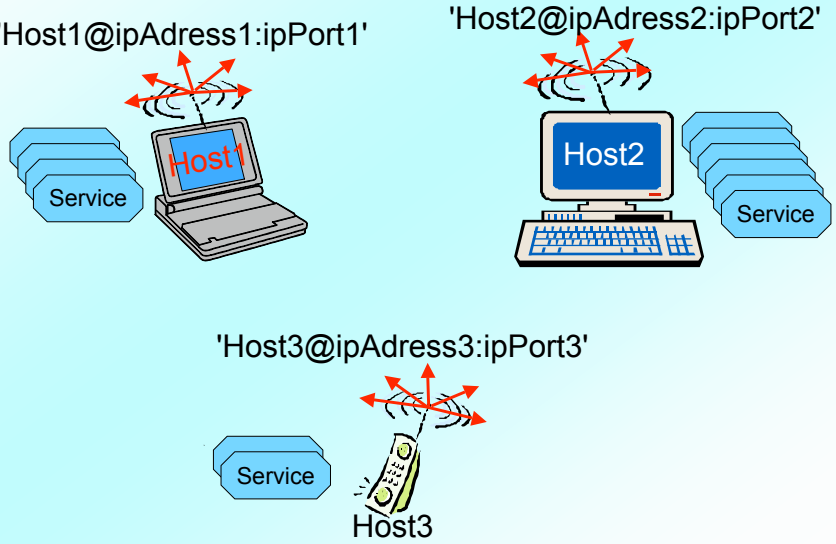
Big picture

Heterogeneous devices in a network



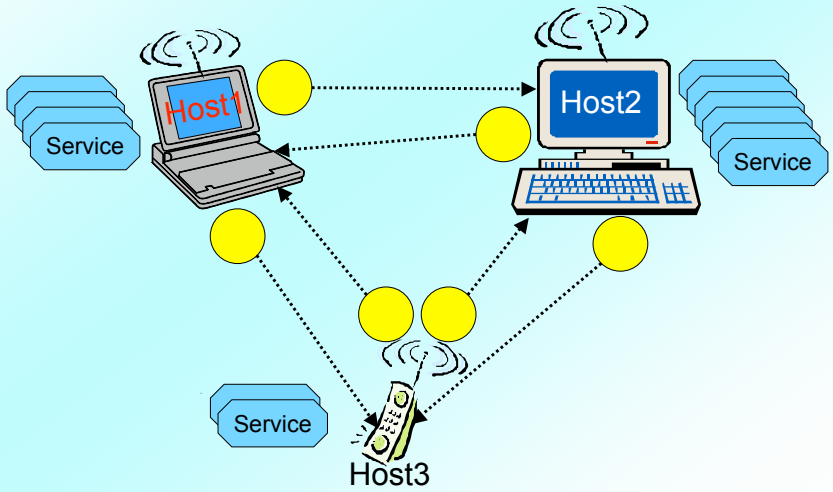
Big picture

Presence Notification Loop



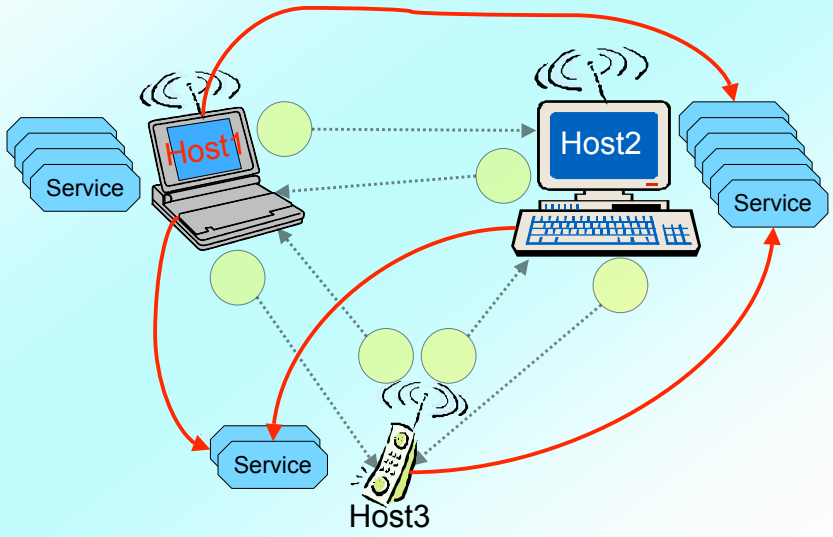
Big picture

Detection => Proxies on peers



Big picture

Interaction => Remote messages



Features

- Middleware
  - Remote communication
  - Automatic host discovery
  - Services registry
  - On-demand deployment at run-time
- Framework
  - Services Functionalities
  - Services GUI
  - Services Administration=Configuration + Usage constraints
    - Each service may have its own specific properties
    - Limited number of simultaneous users of a service
    - Access rights (login/password)

Middleware Features

- Remote communication
  - Any IP network: Wifi, Ethernet, …
  - Any Infrastructure/topology: Ad Hoc, LAN, Internet
- Automatic discovery
  - Detect connections/disconnections
  - Without any prior knowledge on remote hosts
- Services registry
  - White pages (by name) - Yellow pages (by description)
- On-demand deployment at run-time (to do)
  - Automatic download and deploy services client parts
    - e.g. client GUI

Framework Features

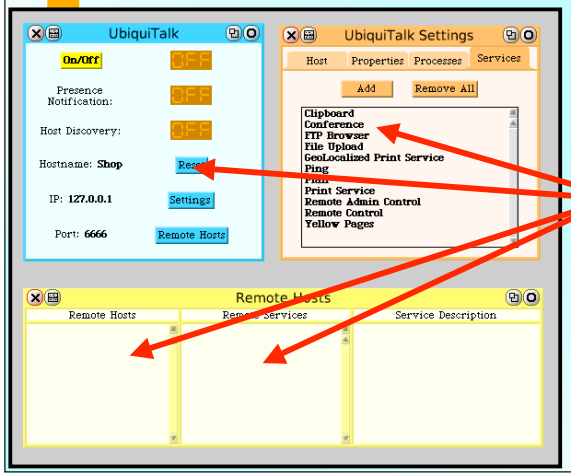
- Service definition
  - Application entry points or middleware extensions
  - Reactive or Proactive
  - 3 parts
    - Provider part
    - Client part (to deploy on-demand) : usually GUI
    - Administration part
- GUI
  - Targeting various display sizes
    - Desktop/Laptop, PDA, Phones (to do)
  - Admin : Service setup, activation, passwords, …

Outline

- Motivation
- Overview
- User Interface
- Applications
- Conclusion

Two GUI depending on the target

For Desktops/Laptops



For PDAs

Variable part

Static Part
Navigation bar

PDA's UI

Main Screen

Activity info

Admin. activities

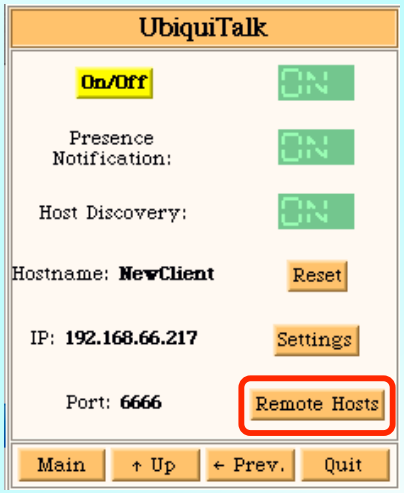
User activities

Host information

Navigation bar

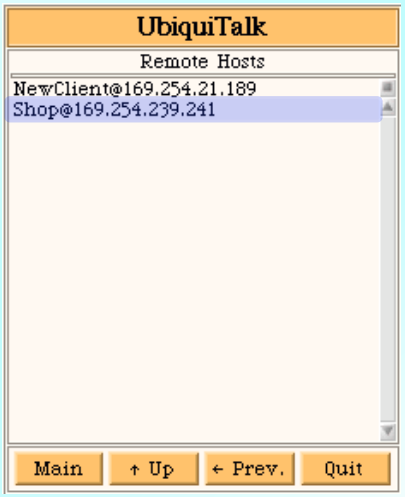
PDA's UI

From Host discovery to service use



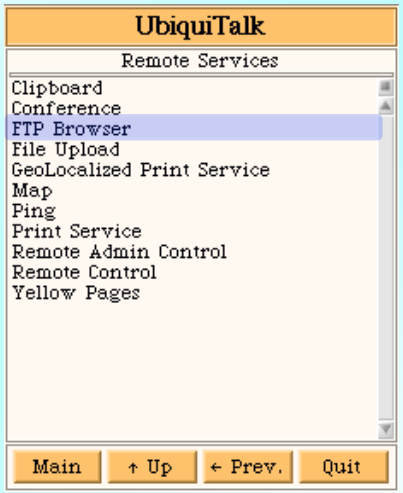
PDA's UI

From Host discovery to service use



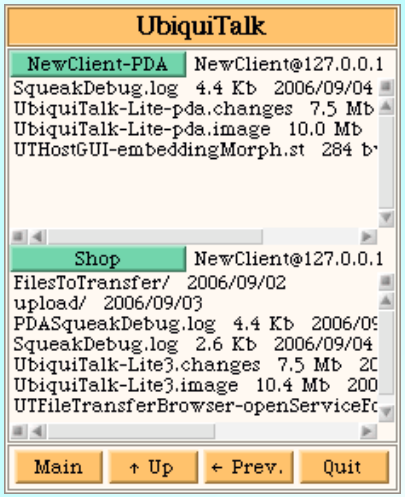
PDA's UI

From Host discovery to service use



PDA's UI

From Host discovery to service use



Outline

- Motivation
- Overview
- User Interface
- Applications
- Conclusion

Some Services implemented so far

- Cross-Platform Copy/Past
- FTP
- Chat Conference
- Printing
- Geo-Localized Printing
- Remote Administration
- Remote Control

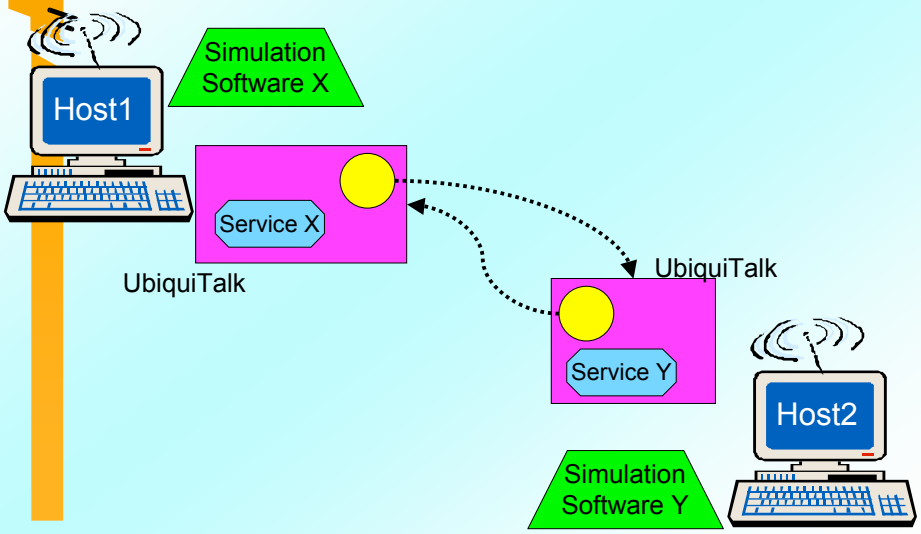
Scientific Computing

The Problem

- Ongoing project
- Large scale application with multiple users
  - Chemistry Consortium (companies and academia)
- Open set of simulation softwares
  - Developed since 15 years
  - Standalone simulation softwares
    - No interaction planned
  - Each partner has a different subset of softwares
- Goals
  - Do cross-simulation
  - Drive simulations remotely

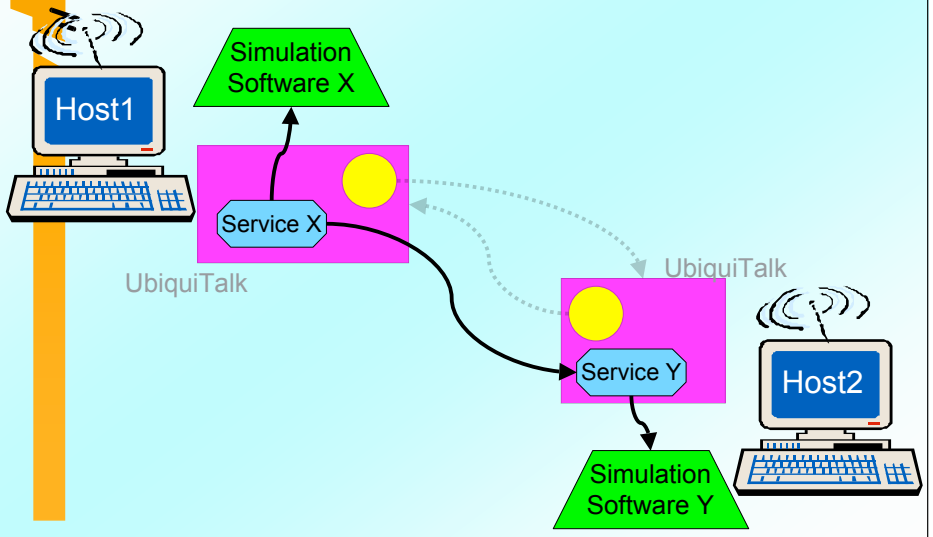
Scientific Computing

The UbiquiTalk-based solution



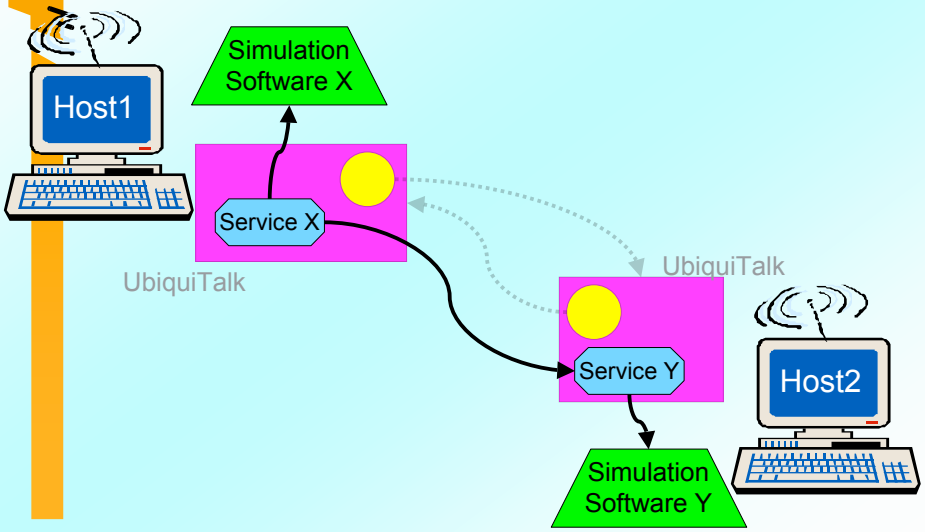
Scientific Computing

The UbiquiTalk-based solution



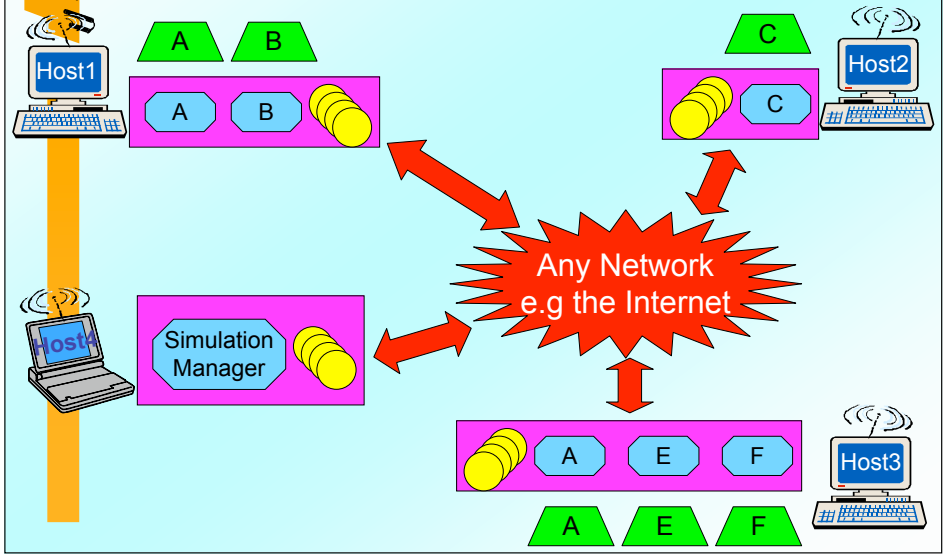
Scientific Computing

The UbiquiTalk-based solution



Scientific Computing

The UbiquiTalk-based solution



Robotic Rescue

- An arbitrary fleet of robots that cooperate in a hostile environment
- Cheap robots
  - High resource constrains

Partners

- o IRD/Géodes, Bondy & Hanoï
- o CNRS/MICA, Hanoï
- o AUF/IFI/MSI, Hanoï
- o INRIA/LORIA/MAIA, Nancy
- o ITC, Phnom Penh
- o CNRS/INRIA/LRI/TAO, Orsay
- o CNRS/GREYC/MAD, Caen
- o ARMINES/CSL, Douai

Seeking for Partners

- Current partners are mainly non-Smalltalkers
- Any Smalltalkers are welcome ☺
  - Companies
  - Academia
- Various partnership possibilities
  - Specific "private" project
  - European funded project
  - …

Outline

- Motivation
- Overview
- User Interface
- Applications
- Conclusion

Summary

- Middleware for P2P unanticipated interaction
- Support for:
  - Automatic Host Discovery
  - Service definition and administration
  - UI
  - Hethorginity
  - Different Uis
  - Managing service hardware requirements
- Goal : Go from research to real world software

Future work

- Support for automatic deployment
  - Deal with heterogeneity
  - I.e. Detect target properties and provide the right implementation of used services
    - What if the implementation is not available?
    - Simply forget it? Provide an incomplete but runnable service?
- Refactoring
  - Fully Uniform architecture
  - => Everything is a service
  - => Fully open architecture : every part will be replaceable
    - E.g. replace discovery protocols

Questions? Comments?

http://csl.ensm-douai.fr/UbiquiTalk