Adaptable Mobile Applications through SATIN:

Exploiting Logical Mobility in Mobile Computing Middleware

Stefanos Zachariadis, Cecilia Mascolo.

Department of Computer Science

University College London

http://www.cs.ucl.ac.uk/staff/s.zachariadis

Physical Mobility

- Ubiquity of mobile computing devices
  - Laptops, PDAs, cellular phones
- Variable connectivity
  - Bluetooth, 802.11x, GSM/GPRS/CDMA/…/3G, infrared, docking
    - Nomadic, ad-hoc …
    - Variable in cost and type
- Numbers increasing
  - 2002: 15.5 million PDAs, 2005: 700 million Bluetooth chips (Gartner)

Characteristics

- Limitations (compared to traditional computing)
  - Memory, battery power, CPU power, erratic (expensive) connectivity
  - Improving but lagging behind still
- Different usage paradigms
  - Input/output
  - Speed, ease of use, frequent but brief usage
    - E.g. Check schedule
  - Reports show that users rarely install applications on mobile devices
    - Applications need to cater to users' needs throughout the device's lifetime

Characteristics (2)

- Heterogeneity!
  - Device/Hardware (Physical)
  - Software/Middleware (Logical)
  - Network
- Very dynamic environment

Logical Mobility

- Ability to sent parts of an application (or migrate/clone a process) to another host
- Popularised by Java
- Classification into paradigms
  - Client/Server (CS)
  - Remote Evaluation (REV)
  - Code on Demand (COD)
  - Mobile Agents (MA)
- Various middleware (mobile & stationary) systems use it

Advantages of Logical Mobility

- Flexibility
  - Dynamic applications
    - For a Dynamic Environment?
    - For a Heterogeneous Environment?
- Automatic software update
  - Maintenance
- New abilities
- Use of remote resources

Motivation

- Investigate the use of Logical Mobility by mobile applications
  - Middleware
- Prove that logical mobility can bring tangible benefits to mobile application developers and users
  - Benefits include faster operation, less user-interaction, services offered based on context and location, reduced cost, better user experience

Deficiencies of Related Work

- Limited use of LM
  - Usage of LM to provide reconfigurability to middleware
    - ReMMoC (Lancs), UIC (Ubicore.com)
    - Allows interaction with services provided by heterogeneous platforms/middleware systems
  - Usage of particular LM paradigms to provide particular services to applications
    - LIME (Wustl) uses MA , PeerWare (Politecnico di Milano) uses REV , Jini (Sun) uses COD
  - Others are not really geared for mobile networks
    - In Fargo-DA disconnections are announced

Current Mobile Application Engineering (PalmOS)

- Event driven, single threaded applications
- Files (Applications & Data) stored in main memory (usually 8MB).
  - Files stored as databases (collection of records)
- Developers compile application into a single file (Palm Resource, PRC)
- Application data can be stored in multiple Palm database files (PDBs).

Current Mobile Application Engineering (2)

- Very limited use of libraries
- Applications have a unique identifier, Creator ID (4 bytes)
  - Registered on a central database
  - Identifies PRCs & PDBs to the OS

What's Wrong with this Model?

- Very limited code sharing
  - On the device itself, between different devices
- Monolithic applications
- Difficult to update application
- No versioning scheme for libraries
- No standard way to know which PRCs a device in reach has.
- Difficulty to install applications
  - Statistics suggest that majority of users never install any 3rd party application

Proposed Solution: SATIN

- Component based middleware
- Allows for static & dynamic configuration
- Small footprint
- Encourages decoupling of applications into modules
- Relies on developers following guidelines

Principles: Architecture

- Modular
- Stresses componentisation
  - Including the middleware itself
- Component identification
  - Dependency scheme
  - Versioning scheme
  - Easy to transmit
- Dynamic addition and removal of modules

Capabilities

- A SATIN component is a capability
  - Ranges from applications to libraries
    - SATIN applications are collections of capabilities with an "executable" one.
  - A capability provides some functionality to either the user or other capabilities.
- Uniquely identified
- Provide a versioning scheme
  - Revisions of a capability
- Provide a Dependency Scheme
- Middleware is a Collection of Capabilities
  - Advertising and Discovery

Logical Mobility in SATIN

- Ability to encapsulate all LM paradigms to a Logical Mobility Unit (LMU)
  - Hosting environment
  - Requesting / sending
  - Deployment
    - Containers, acceptance/rejection
- Language abstractions
  - Objects, Classes, RPCs…
  - Code which does not map directly to the underlying platform is data
- Group various LM entities together
- Signature
- Identification

Some Numbers

- Prototype
  - J2SE
  - Personal Java & J2ME considered
- 40K dist/satin-20030714.jar
- 24K lib/kxml2.jar
- 40K lib/µcode.jar

Future Work

- Looking for the killer app
  - Self-organisation
    - Adaptable mobile computing is an instance
- Evaluation of approach
  - New applications possible
  - Comparison to applications that don't use LM
    - Definition of "best"?
  - Scalability

Conclusion

- Physical Mobility
  - Increased popularity
  - Increased abilities
- Logical Mobility
  - Principles
  - Harness potential of mobile devices
- SATIN
  - Superset of previous approaches
  - Flexible use of LM to applications

Thank You!

http://www.cs.ucl.ac.uk/staff/s.zachariadis

{s.zachariadis,c.mascolo}@cs.ucl.ac.uk