Middleware for Mobile Services and Applications

Ian Warren
ian-w@cs.auckland.ac.nz

Organisation

- Motivation for mobile service applications
- Challenges
- Odin middleware
- Experience
- OdinTools
- Concluding remarks
Motivation

- Smart phones are:
  - Becoming increasingly ubiquitous in society
  - Well resourced
  - Able to host services

Motivation: patient monitoring

- PDA, augmented with a BAN (body area network), hosts a service to monitor its user’s vital signs
- The mobile service responds to requests from remote healthcare professionals and may summon an ambulance if necessary
- => less intrusive monitoring, data available on demand
**Motivation: surveillance**

- PDAs host a mobile surveillance service
- A remote security system manager discovers mobile services and requests them to deliver images via inbuilt or external cameras
- => mobile monitoring

**Motivation: location tracking**

- Couriers’ smart phones host a tracking service that consumers can use to track the location of their parcels
- => On demand location sensing
Challenges

- Key challenges stem from smart device characteristics:
  - Limited resources
    - Memory, processing resources, power supply, network availability and bandwidth
  - Inherent mobility
  - Dynamic operating environment
  - Heterogeneity
Challenges: mobility

1. Device moves to different location
   - Mobile Service (IPv1)
   - Base Station 1
   - Client
   - Connection OK

2. Device moves to different location
   - Mobile Service (IPv2)
   - Base Station 2
   - Client
   - Connection Solved

Challenges: service reachability

- Service reachability is an issue where a device’s connection is via a mobile (e.g. 3G) network
  - Mobile network operators (MNOs) generally discard traffic that originates outside of the mobile network
  - MNOs typically do not assign well-known addresses to devices
Challenges: scalability

- Smart devices’ processing and networking resources do not scale to support many simultaneous clients.

Challenges: availability

- A mobile service’s availability is affected by both its hosting device’s power and network status.
Challenges: heterogeneity

- Developing applications and services to run on smartphones is difficult because of heterogeneity in:
  - Device manufacturers and capabilities
  - Operating systems
  - Programming languages
  - APIs

Odin middleware

- Middleware, in general, is a layer of software that sits between the application and platform (OS + hardware)
- Middleware promotes reuse, portability, and interoperability
Odin: Jini foundation

- Jini is a service-oriented architecture specification and implementation
- Jini offers a programming model that leverages Java and extends it to address the “eight fallacies of distributed computing”

- Logically centralised, physically distributed lookup service
- Protocol independent
- Mobile code
- Leasing mechanism
- Distributed event mechanism

http://www.jini.org

Odin: Jini Surrogate Architecture (JSA)

- The JSA specification arose to allow devices that cannot run Jini to expose their services to Jini clients

- Key elements of the specification include:
  - Surrogate
  - Surrogate host
  - Surrogate container
  - Interconnect

- Protocol for communication between a service/surrogate pair
Odin middleware

1. Registration
   Host-device contacts and registers its surrogate with a surrogate host

2. Register/Join
   The surrogate host registers the surrogate as a Jini service with the lookup service

3. Discover
   The client discovers the surrogate and obtains its service object.

4. Consume
   Client communicates with the surrogate as if it is a normal Jini service.

HTTP request “I’m alive” → HTTP response

Odin: reachability over 3G networks

HTTP request “I’m alive” [service response]

Service request

MNO Firewall allows HTTP Requests from within the network, and their associated responses.

HTTP response [service request]

Service response

JINI Client
Can reach Surrogate OK.
Odin: vertical handover

- Vertical handover is the process of switching between different network interfaces at run-time.
- Odin guarantees not to lose data when performing vertical handover over any supported interconnect.

**Proactive Vertical Handover (3G → Wi-Fi)**

1. Device sending via 3G
2. Device switches to faster Wi-Fi channel when it becomes available

**Reactive Vertical Handover (Wi-Fi → Bluetooth)**

1. Existing Wi-Fi connection breaks
2. Device switches to available Bluetooth channel

Odin: surrogate migration

(a) Device connected to SH #1 and two clients (C1 and C2) consuming the service.
(b) Device requests to be migrated to SH #2. Odin waits until all pending client calls to complete.
(c) Initiate migration after all pending client calls for the device are handled.
(d) Device is updated to point to the new host (SH #2)

Transfer the surrogate to SH #2 along with its state.
New client requests (e.g., C3) are blocked by SH #1.
Odin automatically updates references of existing clients to point to SH #2.
Odin: context aware

- Context sources (CS) send raw context information (R) to the surrogate-host's context processor.
- Pre-processing where raw context information is transformed and current context is updated.
- Context management where adaptations are recommended/applied and context changes are notified.

Odin: dynamic keep alive

- HTTP request "I'm alive" [service response]
- When a device service is "active", the frequency is increased.
- Dynamic keep alive can have a significant impact on resource conservation.

Motivation
Challenges
Odin
Experience
OdinTools
Concluding remarks
A mobile social networking service: iStalker

Main Menu
Can view friends, locations, etc. Each "friend" has their own service providing info.

Details
Can view details for all friends, even when their phones are off. Their Surrogate will still exist.

Map
Can view friend's locations on a map. Location data is provided as a service.

iPhone Processes
Background process communicates with the Surrogate.

iStalker architecture

- iPhone-specific development issues included:
  - The need to jailbreak the iphone due to iPhone OS 3 limitations
  - (Expensive) porting of device-side Odin to Objective C

Motivation Challenges Odin Experience OdinTools Concluding remarks
OdinTools

- Odin provides fundamental infrastructure for developing mobile services and applications, but many problems remain
- Three key questions include:
  - How can we develop services and applications to run on different mobile platforms?
    - { iPhone, Android, J2ME, Windows7, .Net Compact Framework, Symbian, Blackberry, etc. etc. }
    - => Device platform heterogeneity
  - How might a mobile service or application be developed to satisfy stated QoS requirements?
    - => “ility” requirements, e.g. security, availability, scalability, performance, etc.
  - How can Odin be integrated with other standard tools and technologies?
    - Inter-technology interoperability
- => There’s a basic need for tools to add value to the Odin middleware

OdinTools: a model-driven approach

The status quo

- Application
- Middleware
- Platform

Model-driven

- Model
- Automated model transformation
- Generated app/middleware combo
OdinTools: a possible model

- Tools are required to specify, process and verify models
- Output of a model transformation tool might be software (app/middleware combo) that:
  - Is integrated with technology supporting a particular architectural model (e.g. Cloud, P2P, etc.)
  - Allows mobile services to be accessed using particular technologies (e.g. Web Services, ICE, CORBA, .Net etc.)
  - Consumes minimal device-side resources (e.g. small memory footprint, minimal run-time demands)
  - Includes a generated caching mechanism that is appropriate for the given application
  - Provides for mobile code that can be executed either on the device or surrogate host; the decision as to where to execute the code should be made at run-time based on context information
  - ...
- Verification tools might, through simulation, formal models or “proof by construction”, show that generated app/middleware combos satisfy QoS requirements
Concluding remarks

- Odin is an intermediary-based middleware platform for mobile services and applications
  - Odin’s feature-set addresses fundamental challenges and allows resource use to be optimised
- Use of the Odin middleware alone is insufficient to deal with important issues that include:
  - Device platform heterogeneity
  - Variability in need across different kinds of mobile applications and services
- Use of model-driven techniques appear promising in augmenting Odin with value-added tools that further address heterogeneity and variability

- A set of webpages for Odin will be published very soon, and will be accessible via:
  http://www.cs.auckland.ac.nz/~ian-w/