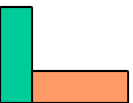


# SATIN: Logical Mobility for Mobile Self-Organisation

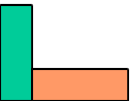
Stefanos Zachariadis

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



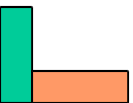
# Outline

- Background
- Component Model
- Middleware System
- Implementation
- Related Work
- Future Work
- Conclusion



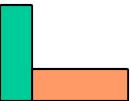
# Trends in (Mobile) Computing (Hardware)

- They are getting faster
- They are getting connected
- They are getting smaller
- They are getting everywhere



# Trends in (Mobile) Computing (Software)

- Not much innovation
- Monolithic apps
- Lack of middleware
- Static apps



# Trends in (Mobile) Computing (Example)

1997:

US Robotics Pilot 1000



128KB 16MHz Serial  
160x160BW

2003:

Palm Tungsten T3



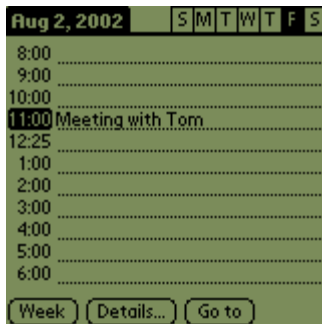
64MB 400MHz  
Serial/USB/Bluetooth/Infrared  
320x480 24bit, Sound, Expansion

# Trends in (Mobile) Computing (Example)

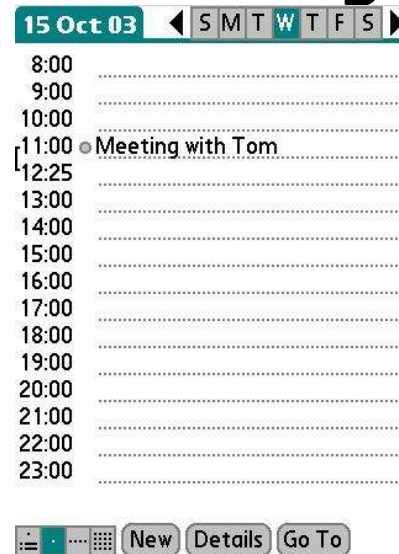
1997:

2003:

US Robotics Pilot 1000 Palm Tungsten T3



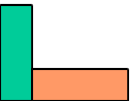
PalmOS 1.0 (DateBook)



PalmOS 5.2 (Calendar)

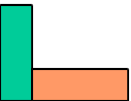
# The Mobile Environment

- Limitations (compared to traditional computing)
  - Memory, battery power, CPU power, erratic (expensive) connectivity
  - Improving but lagging 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



# A Dynamic Environment

- Heterogeneity!
  - Device/Hardware (Physical)
  - Software/Middleware (Logical)
  - Network
- Changes to the environment
  - => Changes to application requirements.





# Self - Organisation

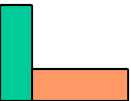
- System adaptation to accommodate changes to its requirements
- Suitability for mobility
- Allows systems to gain new functionality
  - Reacting to changes
- Approaches
  - Expert Systems
  - Genetic Algorithms

# Logical Mobility

- Ability to send 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 exploit this

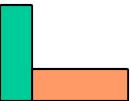
# Components

- Component = functionality
- Coarse-grained guide
- Monolithism vs Componentisation
- Collocation vs Distribution
  - Complexity
  - Size
  - Networking
  - Autonomy



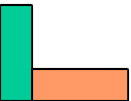
# SATIN

- System Adaptation Targeting Integrated Networks
- Component Model & Middleware
- Minimal Footprint
- Interaction & Autonomy



# Component Model Outline

- Local Component Model
- Distribution Built into the Model
  - But not components
  - Using Logical Mobility
- Applications and the system itself are components

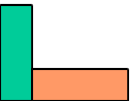


# Components

- Encapsulation of functionality
- Facets
- Properties & Attributes
  - Extensible
  - Heterogeneity (Debian)
  - Identifier, Versioning, Dependencies
    - <ID, “identifier”>
    - <VER, version number>
    - <DEP, dependencies>

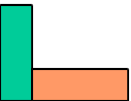
# Container

- Component Specialisation
- Registry/host of components
  - References to all components
- One on each instance
- Dynamic Registration/Removal (delegated)
  - Registrars can have different policies



# Distribution and Logical Mobility

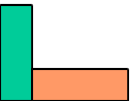
- Provided by the model as a service
- Logical Mobility Entities
- Logical Mobility Units
- Reflective Components
- Deployer





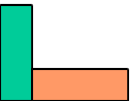
# Logical Mobility Entities & Units

- Logical Mobility Entity (LME)
  - Generalisation of class, object, data
- Logical Mobility Unit (LMU)
  - Composition of LMEs
  - Attributes & Properties
  - Handler
  - Fine grained mobility



# Reflective Components

- Components that can be changed
  - LMU Recipients
  - The Container is Reflective
  - Inspect LMUs
    - Acceptance
    - Rejection
    - Partial Acceptance



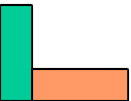
# Deployer

- Component Specialisation
- At least one in each instance
  - Advertised
- Abstracting  
sending/receiving/requesting LMUs
- Uses attributes for matching
- Synchronous and Asynchronous  
primitives



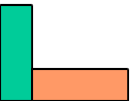
# Middleware

- Advertising & Discovery
  - Advertisable Components
    - Advertising message
  - Advertiser Components
    - Register Advertisable Components
  - Discovery Components
    - Register Listeners



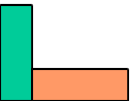
# Logical Mobility

- Finer Grained
- Not only Components, but Classes/Objects
  - Patching
- Logical Mobility as a computational paradigm



# Implementation

- Some Numbers:
  - Prototype
    - 62K dist/satin-20030714.jar
    - 24K lib/kxml2.jar
    - 40K lib/μcode.jar

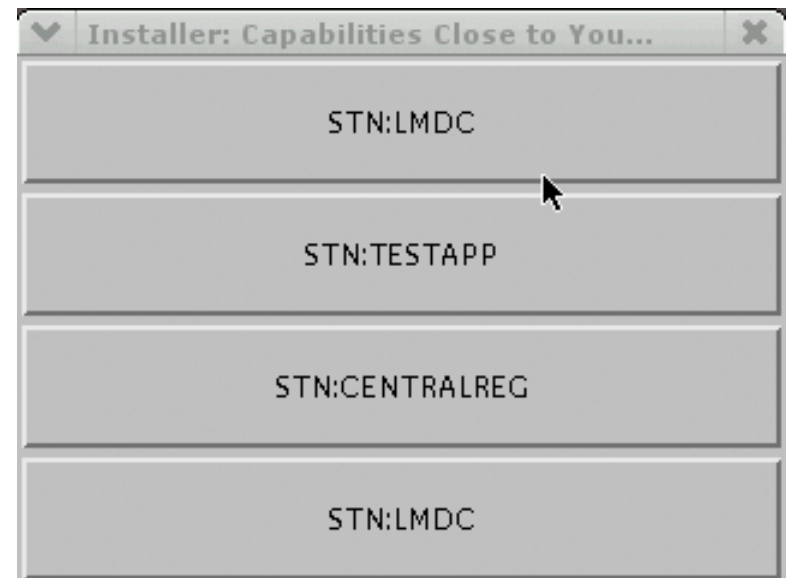
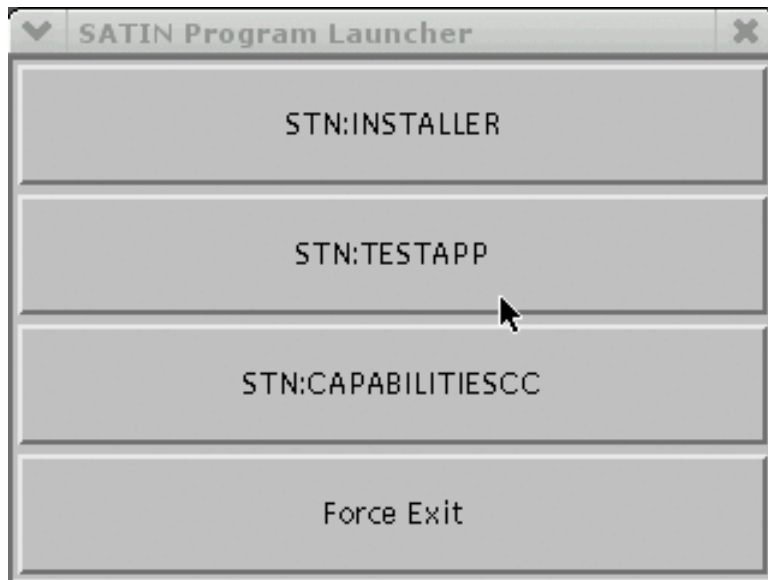


# Example Application: Dynamic Launcher

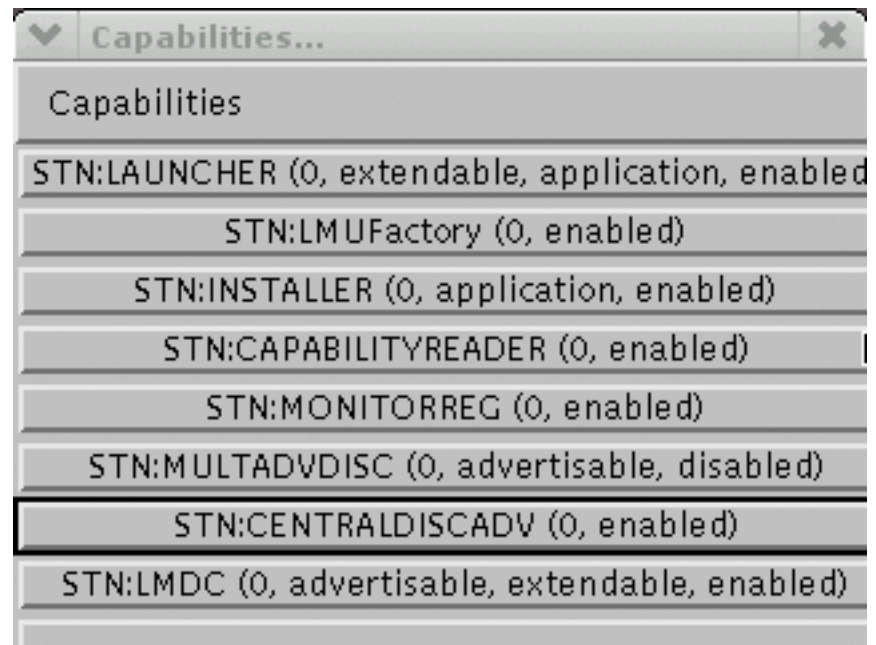
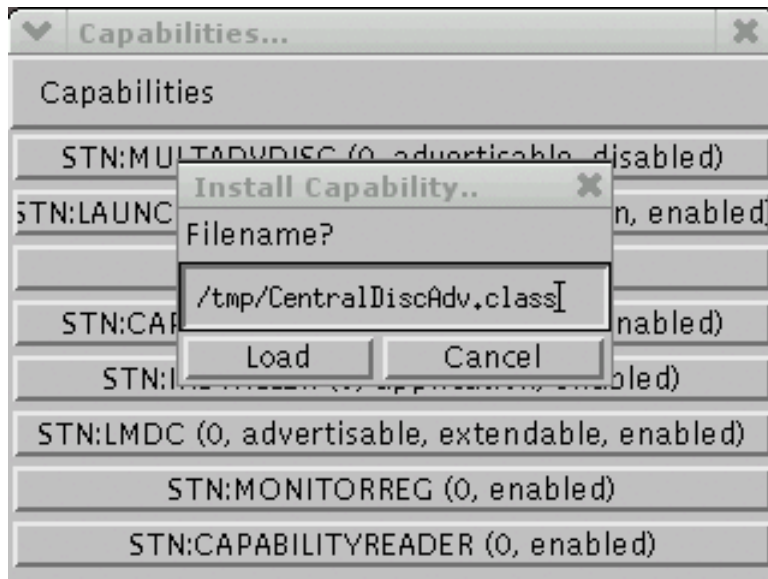
- Similar in Functionality to PDA Launchers
- Installs Components from multiple sources
  - Centralised Source, p2p...
  - Uses any discovery components installed to find components available
  - Uses Deployer to request and receive components
- Transparent update
  - Using any Discovery components installed and Deployer to find and install updates



# Dynamic Launcher [2]

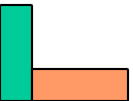


# Dynamic Launcher [3]



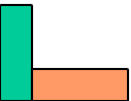
# More Numbers

- Times
  - Startup Time on PDA: 21 seconds
  - Memory Usage on PDA: 1155KB
  - Update to PDA from peer: 2063 ms



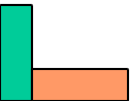
# Related Work

- Logical Mobility Middleware
  - Limited Use of LM
    - System Reconfiguration (UIC, ReMMoC)
    - Too Specific (Lime, PeerWare, Jini, XMIDDLE)
  - Not geared for mobility
    - Disconnections pre-announced (Fargo-DA)
    - Fixed advertising and discovery (one.world)



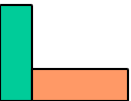
# Related Work (2)

- Component Model Systems
  - Distributed ones unsuitable
    - Large
    - No autonomy (P2PComp, PCOM)
  - Local Component Models
    - Distribution as a service
    - Heterogeneity
    - Some make a distinction between Component providers and consumers (Beanome/OSGi)



# Future Work

- Re-implementation
- More Testing
- Applications



# Conclusion

- The SATIN Component model
  - Distribution as a service
  - Attributes for heterogeneity
  - Applications & System: interconnected local components
  - Reconfiguration of Local Components
- The SATIN Middleware System
  - Componentised Advertising and Discovery
  - Logical Mobility as a Computational Primitive