Alissa Scherchen
3rd year Comp Sci MSci student
A.Scherchen@cs.ucl.ac.uk

3C05: Mobile Computing

Unit 22: Mobile Computing

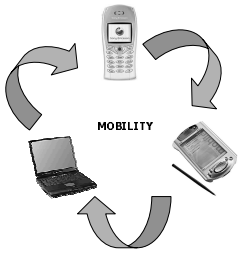
- Objectives:
  - To provide an introduction to mobile computing
  - To distinguish between types of mobility:
    - Ad-Hoc
    - Nomadic
    - Ubiquitous
  - To give an insight in the benefits and problems facing when trying to build a mobile computing system.
  - To show how these principles are applied:
    - Middleware: Tuple Space-Based (LIME)
    - Algorithms
    - Models

What is mobile computing?

- Mobile computing is associated with the **mobility** of hardware, data and software in computer applications.

- Specialized class of distributed systems where some nodes can **disengage** from joint distributed operations, move **freely** in the physical space an **reconnect** to a possibly different segment of a computer network at a later stage in order to resume suspended activities.

- **Not** the same as *wireless computing*



Why do we need Mobility?

- Vast areas of applications (see slide 6)
- Will make a lot of people rich (market is worth billions of pounds and growing each year)!
- Fiction meets Reality:
  - Arm and eye implants (respectively the Cyborg I & II Implants) send a signal to the computer in the department of Reading University, which identifies professor Kevin Warwick. The computer is programmed to open doors, switch on lights etc. depending where the professor is located. His wife has also become a "cyborg" willingly...

PRESENT

FUTURE?

Portable Information Appliances

Mobile applications

- **Vehicles** (traffic monitoring and coordination, GPS).
- **Emergencies** (access to outside world after natural catastrophe, military applications).
- **Traveling salesmen** (mobile office, consistent DB for all agents).
- **Web access** on the move.
- **Location aware services** (new to a place and require printer -> use ad hoc network with own mobile device to access the service).
- **Information services** (push: stock quotes, pull: nearest cash ATM, find nearest printer – JINI services).
- **Disconnected operations** (file-system caching for off-line work, mobile agents).
- **Entertainment** (network game groups).

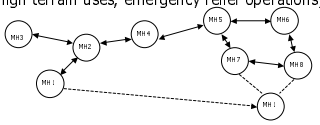
The wireless data challenge

- To enable people to communicate/compute effortlessly **where** they want, **when** they want, **without** "wires".
- **Technical challenges** of mobile computing
  - Portability
  - Mobility
  - Wireless Communications
  - System issues
  - Security issues

Different types...[1]

- **Ad-Hoc**
  - Network that comes together as needed, not necessarily with any assistance from the existing (Internet) infrastructure.
  - Consists of a set of mobile hosts, (dynamically) connected to the network through high-variable quality links, and executing in an extremely dynamic environment.
  - Nodes can communicate with their neighbours only if the ad-hoc network is in the communication range of the nodes.
  - Multiple hops may be required as well to cross the network.
  - Useful in conditions where the infrastructure is absent, impractical to establish or expensive to build (military applications, high terrain uses, emergency relief operations)

Ad Hoc network of mobile nodes – these are able to move relative to each other. MH1 moves away from MH2 and establishes new links with MH7 and MH8. Most algorithms also allow for the appearance of new mobile nodes and the disappearance of previously available nodes.



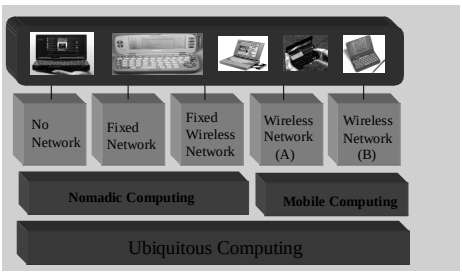
Different types...[2]

- **Nomadic**
  - "transparent virtual networking"
  - Composed of a set of mobile devices and a core infrastructure with fixed and wired nodes.
  - Mobile devices move from location to location , while maintaining a connection to the fixed network.
  - special features:
    - uncertainty of location
    - repeated lack of connections
    - migration into different physical and logical environments while operating

Different types...[3]

- **Ubiquitous**
  - "Third Paradigm" computing (*Alan Kay of Apple*), *"calm technology"* - technology recedes into the background of our lives.
  - Envisions a world of fully connected devices, with cheap wireless networks everywhere - web.
  - Postulates that you need not carry anything with you (unlike PDAs) , since information will be accessible everywhere.

Nomadic, Mobile & Ubiquitous

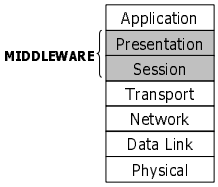


Terminology

- *Physical Mobility* - users and devices can move (*ubiquitous computing*).
- *Logical Mobility* - computations can move on the network (i.e., computation is achieved through *mobile software agents*).
- *Coordination* – investigates models and languages that separate the specification of the behavior of the individual processes from the communication needed to coordinate such behaviors.
- *Location* – identifies the position of a mobile unit in space. Type of location affected by the choice of unit of mobility (e.g.: Cartesian Coordinates for a mobile device).
- *Context* – for a mobile unit (dynamic context) it is determined by its current location which, in turn, defines the environment where the computation associated with the unit is performed. Includes resources, services and other components of the system.
- *Mobile Agents* - migration of execution units autonomously performing tasks without requiring permanent connectivity towards the client. Characterized by an identity, location, a set of labeled tuple spaces and a minimal interface (in, out, rd).

Middleware Systems

- Enable communication between distributed components.
- Higher level of *abstraction* built using primitives of the network operating system.
- Solutions to resource sharing and fault tolerance requirements.



ISO/OSI Reference Model

Middleware for Mobile Distributed Systems

- Currently there is still a lack of high-quality middleware software -> technology is very young still and many assumptions from traditional MW can't be translated into mobile MW.
- Middleware for mobile distributed systems is split into different research areas:
  - Context-aware (*principle of Reflection*);
  - Location-aware (e.g.: *Nexus*);
  - Data sharing-oriented (e.g.: *Bayou, Coda, Odyssey*);
  - Tuple space-based (e.g.: *Lime, TSpaces*).

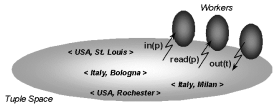
LIME [1]

- **L**inda **I**n a **M**obile **E**nvironment
- *Tuple Space* Middleware
- Java-based middleware which provides a coordination layer that can be exploited for designing applications which exhibit either logical or physical mobility - or both.
- Targeted toward the complexities of the ad hoc mobile environment.
- Lime provides coordination among processes (*Lime agents*) via a **shared memory mechanism**. When processes use Lime to coordinate, no messages are explicitly sent to other processes.
- All communication occurs through access to a shared medium, namely the *tuple space*.
- Tuple access and movement, i.e. when a new tuple is available or created events are used to notify users.

Lime Agents - active entities in Lime presented by Java threads implementing the lime.LimeAgent interface

LIME [2]

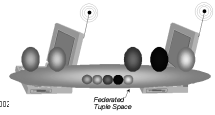
- *Tuple space* - a global and persistent repository of *tuples* (essential data structures constituted by an ordered sequence of typed fields).
- Linda provides a minimal interface to the tuple space, with only three operations:
  - *out* - to write a tuple to the tuple space.
  - *rd* - get a copy of a tuple in the tuple space that matches a given template.
  - *in* - in addition to getting a copy of a matching tuple withdraws it from the tuple space.



Three Linda processes interacting with a single tuple space with the standard Linda operations

LIME [3]

- Agents (the active components in the system) can roam across mobile hosts (which act as mere containers for the agents) which can roam across the physical space.
- The presence of mobility prevents the existence of a global and persistent tuple space.
- In Lime, each mobile agent owns a *Lime tuple space* (LTS) which follows the agent during migration.
- The notion of a global and persistent tuple space is dynamically recreated on a host by merging the LTSs of all the mobile agents there collocated -> *host-level transiently shared tuple space*.



Two agents on one host and three on another. The small spheres represent the tuples of the same-colored agent. Because there is connectivity between the hosts, the federated tuple space has the illusion of containing the tuples of all agents on the hosts. In other words, an in operation issued by any one of the agents can return a matching tuple from any agent

LIME [4]

- Promotes the reduction of details of mobility (transparency) and the distribution to changes to what is perceived as the local tuple space.
- Allows designers of mobile systems to be alleviated from the burden of constantly maintaining a view of the context consistent with changes in the configuration of the system.
- BUT...
- Simplification may be too restrictive where there is a need for a more detailed amount of context awareness (e.g.: control portion of context that has to be accessed).
- May be solved by:
  - Extending Linda operations with tuple location parameters.
  - LimeSystem
  - Reactions set to tuple space

Current Research [1]

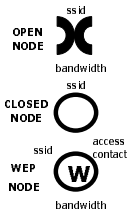
- Models:
  - Permit the precise description of existing languages and system semantics.
  - Used to emphasize parallels and distinctions among various forms of mobility (logical and physical) and are concerned with the formulation of appropriate abstractions useful in specification and evaluation of such mobile systems.
  - Mainly concerned with the characteristics of mobile units such as the unit of mobility (who is allowed to move), its location (where a mobile unit is positioned in space) and its context (determined by the current location of mobile units).
  - There are many existing models and many more are still in research:
    - Random mobility model(s)
    - Markovian model
    - Exponential Correlated Random Model
    - Nomadic Community Model

Current Research [2]

- Algorithms:
  - Reflect the assumptions that are made about the underlying system.
  - Mobile algorithms are obliged to treat in much detail space and coordination of mobile systems.
  - Have to carefully take into consideration location changes, the frequency of disconnection, power limitations and the dynamic changes in the connectivity pattern of mobile systems.
  - This field of theory is in fact spread among a vast spectrum of research due to the large diversity of mobile systems.
  - Example (Dept. of Computer Science Washington University):
    - Movement of a mobile unit within a region of base stations is tracked down -> broadcast of a message can be limited in range, involving less network traffic while still allowing for rapid mobile unit movement -> based on the model of diffusing computations by Dijkstra and Scholten.

Warchalking

- Founder Matt Jones but originally dates back to 1930s where during the Depression in the U.S, hobos drew signs to indicate to each other where they could get a meal.
- Chalk symbol drawn on a wall or pavement to indicate the presence of a wireless networking node.
- Ad hoc process of people discovering Wi-Fi nodes.
- System administrators made aware of flaws in their system.
- Concern for "hacks" and "cracks" within a company – has increased awareness in the lack & complacency of security that currently resides in wireless networks.



Conclusion:

- **Mobility**:
  - Provides intellectual and social excitement
  - Challenges old assumptions
  - Offers great opportunities for research
  - Demands new kinds of solutions
  - Requires an integrative research strategy
    - Application centered
    - Reliance on coordination models with precise semantics
    - Delivery through middleware

References for the hungry mind…

- **WWW**
  - www.softwaresystems.org/future.html
  - www.it.iitb.ac.in/it644/lectures/
  - http://lime.sourceforge.net/
  - www.free2air.org
  - www.warchalking.org
- **Books**
  - *Mobility,* Milojicic, et al. (Addison-Wesley)
  - *Concurrent Systems*, Jean Bacon (Addison-Wesley)
  - *Ad-Hoc Networking*, Charles E. Perkins (Addison-Wesley)
- **Papers**
  - *Software Engineering for Mobility: A Roadmap*, G.P. Picco et al.
  - *Mobile Computing Middleware*, C. Mascolo et al.