

ICSE 2000



Software Engineering for Mobility

Gruia-Catalin Roman

Gian Pietro Picco (Politecnico di Milano)

Amy L. Murphy



Outline



- Working definitions
- Forms of mobility
 - Physical mobility
 - Logical mobility
 - Mobility paradigms
 - Points of distinction
- Roadmap
 - Coordination
 - Middleware
 - A research framework
- Conclusions



Working Definitions



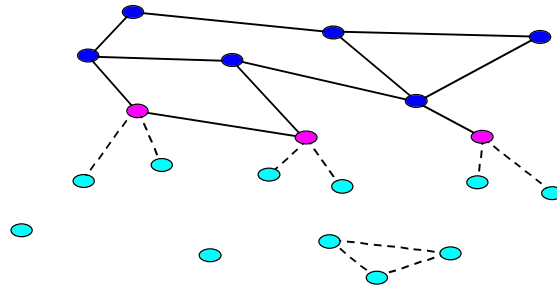
- Software Engineering is a discipline that studies
 - Software artifacts
 - Processes as (approximate) guarantors of software quality
 - People as (imperfect) producers of software
- Mobile Computing entails the study of systems in which computational components may change location



Physical Mobility



- Fixed network
- Nomadic computing
- Ad hoc computing



Design Issues



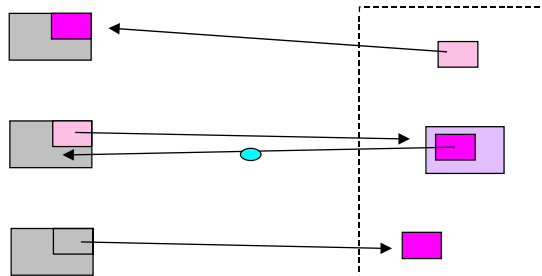
- Technological limitations
 - Frequent disconnection
 - Unpredictable movement
 - Asymmetric bandwidth
 - Limited resources
 - Short battery life
- Exploiting spatial knowledge
- Exploiting movement profile
- Reasoning about behavior
- Connectivity maintenance
- Adaptation
- Standardization



Logical Mobility



- Code on demand
- Remote evaluation
- Mobile agents



Logical Mobility Trends



- Code on demand is popular
 - Java is the dominant technology
- Mobile code languages (e.g., Oblique, Facile, Telescript) are losing ground
- Mobile agent systems (e.g., Agent TCL) tend to rediscover old problems
- Middleware is growing in practical importance
 - Java/RMI and Jini



Mobility Paradigms



- Continued access to remote resources
- Delegation to mobile agents
- Disconnected operation
- Ad hoc grouping of people and devices
- Mass mobility
- Wide open systems



Points of Distinction



- Unit of mobility
 - Blurring of physical and logical
 - Relation to the unit of execution
 - Fine-grained mobility
- Treatment of location and space
 - Cartesian coordinates
 - Host or process id
 - Complex evolving spaces
- Context definition
 - Location affects context
 - Coordination manages context



Context Definition



- Location management
- Discovery
- Clustering
- Security
- Local versus distributed
- Quality assessment
- Transparency
- Transitivity



Coordination Advantage



- Coordination is a design strategy that promotes a high degree of separation between
 - Computation
 - Communication
- In the mobile context a coordination perspective fosters
 - Decoupling among mobile units
 - Open system programming

11

Coordination Challenges



- Fundamental assumptions
 - Window of vulnerability
 - Eventuality of encounter
 - Pattern of movement and behavior
- Guarantees
 - Strong atomicity
 - Transaction scope
 - Weak consistency
- Response to change
 - Event processing
 - Reactive programming

12

Middleware



- Middleware rests above the operating system and provides developers with specialized mechanisms and services in a highly integrated fashion
- Recognized advantages
 - Higher level of abstraction
 - Domain specialization
 - Syntax of existing language
 - Ready to use and easy to adopt

13

Middleware Desiderata



- Model-driven
 - clean concept presentation
- Resource-sensitive
 - flexible encapsulation
 - controlled transparency

14

A Research Framework



- Models
 - Mobile UNITY and CodeWeave
- Algorithms
 - Message delivery to hosts and agents
- Middleware
 - LIME (Linda in a Mobile Environment)
- Applications
 - RedRover (video games en plain air)
 - RoamingJigsaw (collaborative work)

15

Conclusions



- Mobility
 - Provides intellectual 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

16