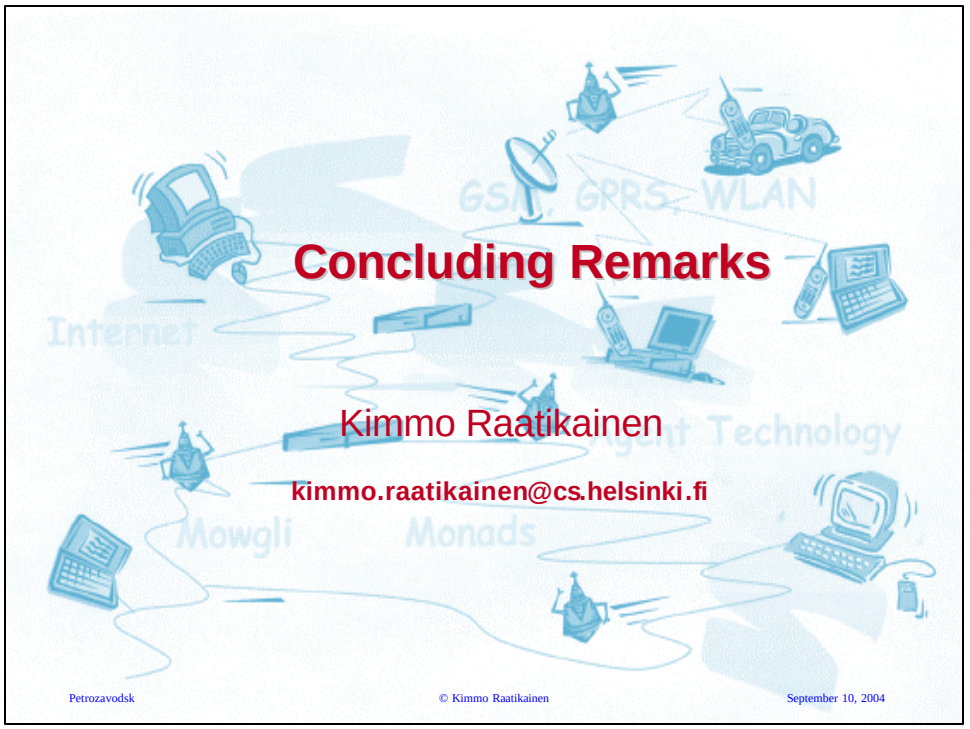


Concluding Remarks

Kimmo Raatikainen

kimmo.raatikainen@cs.helsinki.fi

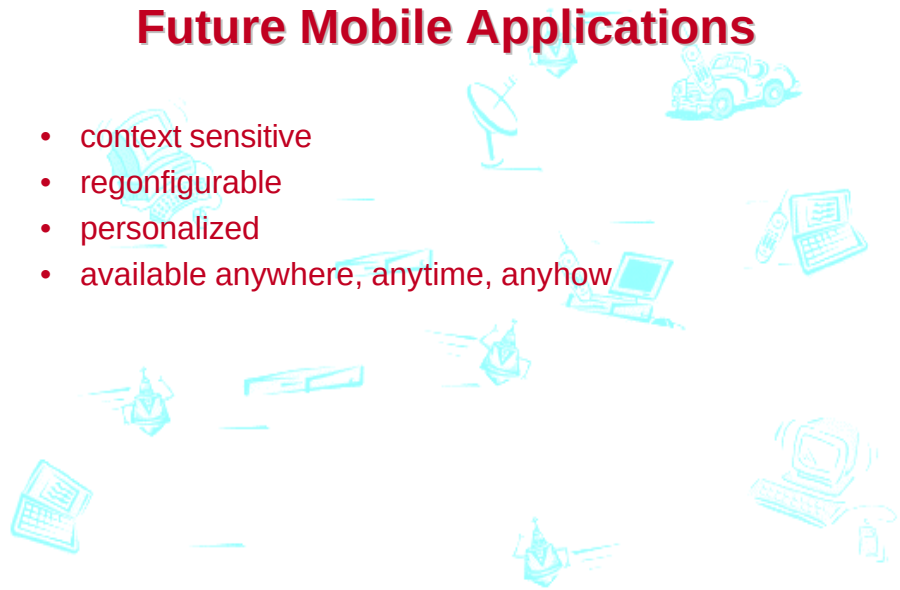
Lession Outline

- New Networking Environments
- Paradigm Shift
  - Three Questions
  - Six Premises
- Six Primary Research Challenges
- Conclusions

New Networking Environments

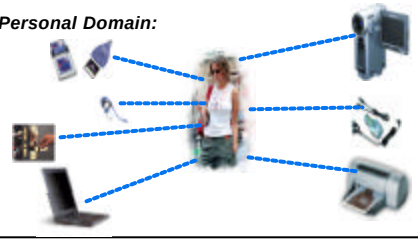
Future Mobile Applications

- context sensitive
- regonfigurable
- personalized
- available anywhere, anytime, anyhow

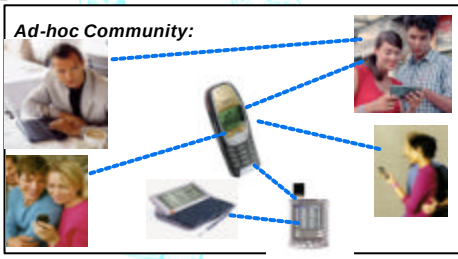


Technology Domains

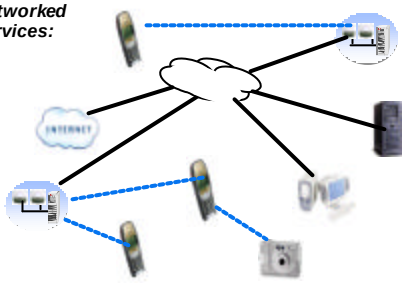
Personal Domain:



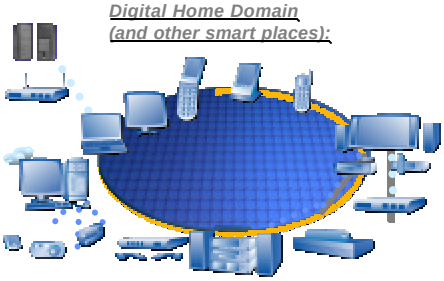
Ad-hoc Community:



Networked Services:



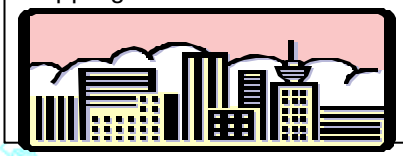
Digital Home Domain (and other smart places):



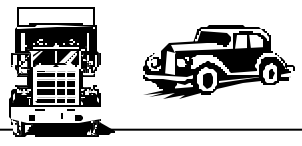
Other Smart Places

Office:

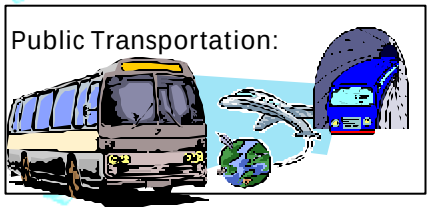
Shopping Center:



Private Transportation:



Public Transportation:



… but do not forget non-smart places

Objective

Solution stack as similar as possible for all technology domains

Paradigm Shift

- From technology centric to user centric
- From what we have now to what we leave behind us

Six Premises

1. Forget end user terminals
   - dynamically configured end user systems
2. Stop thinking users in isolation
   - communities and buddies are important
   - users will have several roles and belong to several communities
3. Users want to be involved
   - user in the driver's seat
4. Devices are for the users
   - understanding users
5. People are different
   - managing mass-scale personalization
6. Trust is the king
   - systems that compromise the trustworthiness abuse our right of privacy and will be considered as an insult and a fraud

Three Questions

1. Do our current operating systems support reconfigurability?
2. Are our programming models and tools adequate for context aware applications?
3. Do our current middleware solutions support development of such applications?

Answers: No, No, and No

Six Primary Research Challenges

1. Reconfigurable systems
2. Context-awareness
3. Security - Trust - Privacy
4. Software Development and Maintenance
5. Programming models
6. Wireless communications

Reconfigurable Systems

- End-user system instead of end-user device
- Research issues:
  - detection of devices
  - environment monitoring
  - event notification
  - event filtering
  - system modeling
  - configuration management
  - management of ad-hoc communities
  - group communication
  - decision rules for reconfiguration

Context awareness

- ***Almost any information available at the time of interaction can be seen as context information***
- Research Issues:
  - extraction of context information
  - interpretion of context information
  - reasoning about the current contextual situation
  - adaptation of application behaviour
  - ways to express which pieces of information belong to the context
  - ***distributed (RDF/XML) data management***
  - ***context modeling***

Security - Trust - Privacy

- Security, trust and privacy must be addressed
  - from the very beginning of system design
  - on all levels: hardware, operating system, protocols, middleware
  - Trust is not of type On/Off
- Research issues:
  - protecting system against unauthorized modifications
  - program validation/verification
    - what an uploaded/downloaded piece of software really does
  - trust modeling
  - how fragments of information can be efficiently shared in a controlled manner
  - key/certificate management
  - implications of ad-hoc communities
    - what can be done without trusted servers

Software Development and Maintenance

- Software Architectures
  - to get the software architecture right:
    - Not too detailed – Not too summary
  - modularity allowing exchanges
    - any block of software/hardware can be replaced
    - new “hardware” technologies can be incorporated
  - clarify thinking: architecture is a design tool
    - not to mix apples and oranges
- Software Processes
- Software Life Cycle Management
- Service Configuration and Deployment

- Targets:
  - Increase abstraction level without scarifying performance too much
  - Increase automation (through modeling)
  - From craftswork to industry

Programming Models

- Currently Java and client/server dominates
- Many useful applications are naturally expressed according to the client-server paradigm
  - How to implement servers in proximity networks
- Context expressed as rules
  - Condition-action programming model: to specify conditions under which each action is to be launched
  - Guarded Commands: Old ideas of Dijkstra and Hoare
- Partially available systems – a new approach to fault tolerance
  - Programming models to compensate/overcome missing functionality

Wireless Communication

- air link is and will remain narrow and error-prone
- optimization on all levels is necessary
  - link, network and transport layers
  - messaging layer
  - communication languages (presentation layer)
  - interaction protocols and patterns
  - ***One can destroy the performance on each level!***
- Quality-of-Service in wireless and mobile systems
- Group communication
- Mobility management:
  - terminal mobility
  - (sub)network mobility
  - personal mobility
  - session mobility

Conclusion

- It is time to reconsider fundamentals!
- Evolution is fine but you need to be ready for a revolution.
  - In 1972 Dijkstra defined computer science as managing complexity
  - Now we really need to manage the complexity

Final Remark

Linus took my OS course in Spring 1991

I did not spoil him