


Johan J. Lukkien, j.lukkien@tue.nl
TU/e Informatica, System Architecture and Networking




Real-Time Architectures 2003/2004

Introduction

Johan Lukkien

29-03-2004 1

Johan J. Lukkien, j.lukkien@tue.nl
TU/e Informatica, System Architecture and Networking




Overview

- Definition of real-time systems
- System classification
- Embedded systems evolution

2

Johan J. Lukkien, j.lukkien@tue.nl
TU/e Informatica, System Architecture and Networking



Real-time systems

- **Definition:** a system in which correct operation depends not only on logical results of computation of values but also on the time these results are produced
 - *Stankovic in Byte, '92*
- Hence, performance plays a role....
- though not just high, but predictable!
- Timeliness: example of an "extra-functional requirement"

3

Johan J. Lekkien, j.lekkien@tue.nl
TU/e Informatica, System Architecture and Networking

Extra-functional requirements

- Requirements that differ from
 - functional relationships (relation between inputs and outputs)
 - behavior (visible interaction sequences)
- Typical domain includes many extra-functional requirements besides timeliness
 - timeliness, energy, speed, memory use
 - look-and-feel, scalability, distribution,
 - dependability
 - robustness, reliability, extensibility, availability, maintainability, fault tolerance, security, consistency, safety

4

Johan J. Lekkien, j.lekkien@tue.nl
TU/e Informatica, System Architecture and Networking

System Architecture

- Architecture
 - ... externally visible, overall structure of a system in terms of components, subsystems and interconnections
 - ... the basic interfaces and interactions
 - between these components
 - and with the system environment
 - ... rules for design and evolution
- The architecture is the prime place to address extra-functional requirements

5

Johan J. Lekkien, j.lekkien@tue.nl
TU/e Informatica, System Architecture and Networking

System classes

- Transformational
 - start/stop at any time
 - response time non critical
 - time independent
 - e.g. administration, information systems
- Reactive
 - interactive
 - response time non-critical, though important
 - in user dialog, no self-initiated actions
 - real-time systems

6

Johan J. Lukkien, j.lukkien@tue.nl
TU/e Informatica, System Architecture and Networking

Reactive systems (cnt'd)

- Interactive systems
- Real-time systems
 - connected to physical world, react to (external) events
 - time dependent, response times vital
 - do not stop during critical operation
 - stopping is unusual, e.g. maintenance, switch-off
 - *monitor and control a physical process*
-hardware/software systems that in themselves are invisible but are there to support an overall function
 - embedded systems

7

Johan J. Lukkien, j.lukkien@tue.nl
TU/e Informatica, System Architecture and Networking

Criticality classes

- Dependable real-time systems
 - High cost of failure
 - Possibly loss of life on failure
 - Guaranteed (especially timeliness)
 - Example: Industrial control
- High performance real-time systems
 - Low probability of failure
 - Constant quality of service
 - High regularity in performance
 - Example: Consumer electronics

8

Johan J. Lukkien, j.lukkien@tue.nl
TU/e Informatica, System Architecture and Networking

Embedded systems characteristics

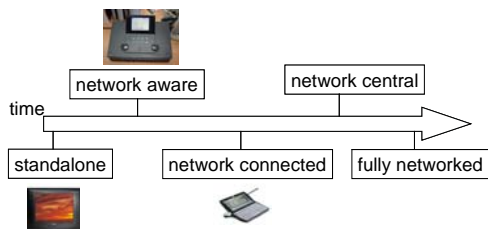
- Heterogeneous
 - both in design as in product, platform
- Resource limited
 - cheap, small, minimal, low power, ...
 - limited user interface
- Cost effective
 - balance hardware and software, optimal hardware use
- In itself not visible
 - platform invisible to user
 - serves device core function

9

Embedded systems evolution

- Software replaces hardware
 - initial and most important application of real-time systems
- Product competition on features
 - features easier to realize with software
 - more and more it is software that determines *what* a product is
 - increasing diversity of functions
- Signals becomes digital
 - telephony, audio, video
- System becomes open
 - programmable, replacement of software [flash firmware, download new applications, remote update]
- Fully networked system
 - cooperation with other embedded systems

Evolution of embedded networking



Conclusion

- Real-time systems initially replace hardware
 - focus on resource control, effective use of resources, control of environment
- There is a trend towards distributed systems and applications
 - devices always function in a distributed context
 - always some 'ad-hoc' nature
 - real-time networking/communication becomes increasingly important
 - controlled environment becomes cooperating environment
