IAF0030
Arvutitehnika erikursus I

Enemies of Dependability

Gert Jervan

Arvutitehnika instituut (ATI)
Tallinna Tehnikaülikool

Tallinn University of Technology

Lecture Outline

- **Introduction**
- **Software**
- **Hardware**
- **Humans**

Ajakava

- Viimased loengud (4 tk, k.a. tänane)
  - 16. märts – 6. aprill
- Paus (konsultatsioon Case Study'de kohta)
  - 13 aprill
- Case Study'de presentatsioonid
  (á 3-4 presentatsiooni)
  - 20. aprill – 4. mai
- Paus (konsultatsioonid, järeltööd jms)
  - 11. mai
- Eksam (Case study kirjatöö peab olema minu laual eksamile EELNEVAL päeval!!)

Downtime

- Planned downtime
  - Maintenance, repair, upgrade
- Unplanned downtime
- Dependability:
  - Turn unplanned uptime into planned downtime
  - Reduce downtime (magic nines)

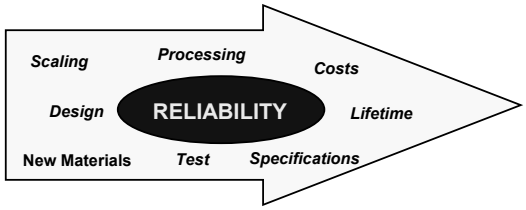
Sources of Problems

Category	Early 80s	Late 80s	90s	2000s
Hardware + environment	32%	29%	20%	?
Software	26%	58%	40%	?
Human Operators	42%	13%	40%	?

Hardware and environment failures

- moving parts, high speed, low tolerance, high complexity: disks, tape drives/libraries
- lowest MTBF found in fans and power supplies
- often fans fail gradually → subtle, sporadic failures in CPU, memory, backplane
- environment: power, cooling, dehumidifying, cables, fire, collapsing racks, ventilation, earthquakes, ...

Hardware Reliability Challenges



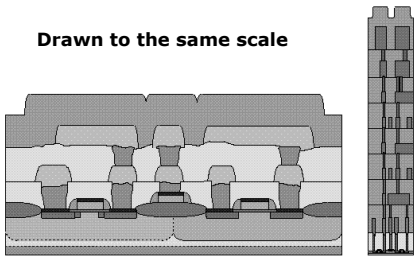
Reliability Dependencies and Impact to Cost

Hardware - Background

- Chip designers, device engineers and the high-reliability community recognize that reliability concerns ultimately limit the scalability of any generation of microelectronics technology
- Statistical methods and reliability physics provide the foundation for better understanding the next generation of scaled microelectronics
  - Microelectronics device physics
  - Reliability analysis and modeling
  - Experimentation
  - Accelerated testing
  - Failure analysis
- The design, fabrication and implementation of highly aggressive advanced microelectronics requires expert controls, modern reliability approaches and novel qualification strategies

What is Technology Scaling

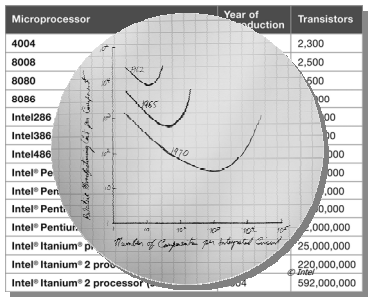
Drawn to the same scale



1.0 μm
Mid 1980s
Speed: 10 MHz

0.1 μm
Early 2000's
Speed: 3 GHz

Moore's Law

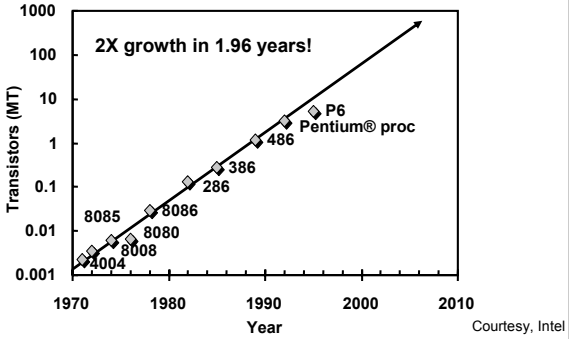


Microprocessor	Year of production	Transistors
4004		2,300
8008		2,500
8080		500
8086		00
Intel286		00
Intel386		00
Intel486		000
Intel® Pe		000
Intel® Pen		,000
Intel® Penti		0,000
Intel® Pentiu		,000,000
Intel® Itanium® p		25,000,000
Intel® Itanium® 2 proc		220,000,000
Intel® Itanium® 2 processor	004	592,000,000

"...(T)he first microprocessor only had 22 hundred transistors. We are looking at something a million times that complex in the next generations—a billion transistors. What that gives us in the way of flexibility to design products is phenomenal."
Gordon E. Moore, April 19, 1965

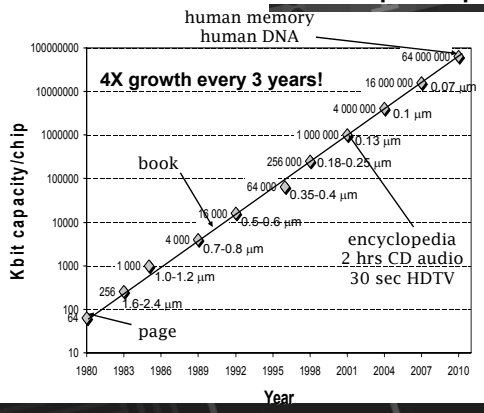
Moore's Law in Microprocessors

Transistors on lead microprocessors double every 2 years



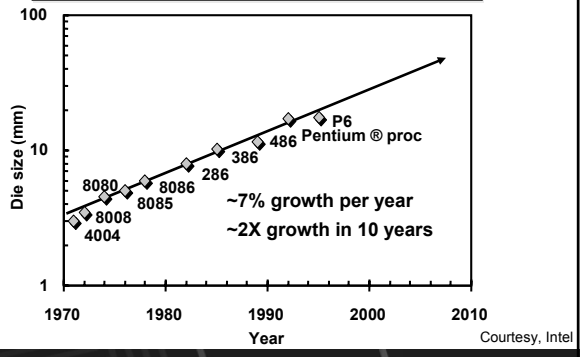
Courtesy, Intel

Evolution in DRAM Chip Capacity



Die Size Growth

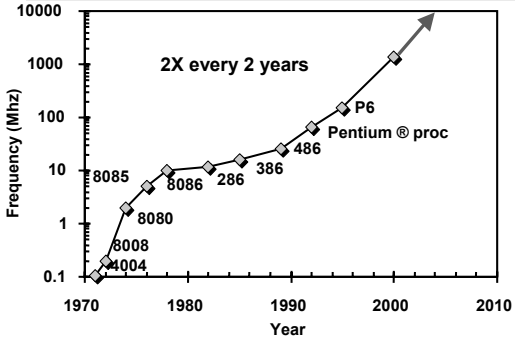
Die size grows by 14% to satisfy Moore's Law



Courtesy, Intel

Clock Frequency

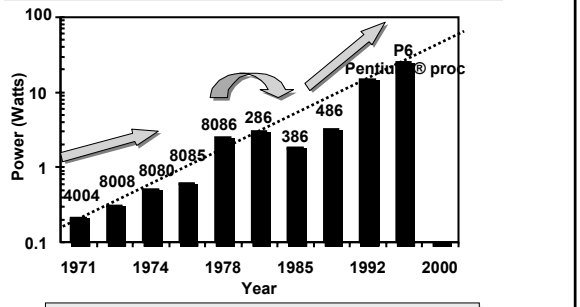
Lead microprocessors frequency doubles every 2 years



Courtesy, Intel

Power Dissipation

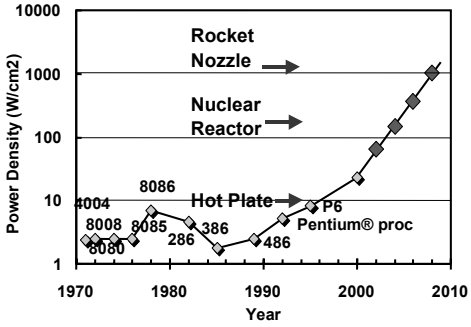
Lead Microprocessors power continues to increase



Courtesy, Intel

Power delivery and dissipation will be prohibitive

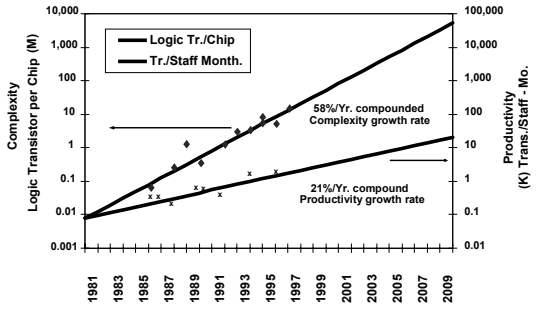
Power Density



Courtesy, Intel

Power density too high to keep junctions at low temp

Design Productivity Trends



Complexity outpaces design productivity

Courtesy, ITRS Roadmap

Technology Directions: SIA Roadmap

Year	1999	2002	2005	2008	2011	2014
Feature size (nm)	180	130	100	70	50	35
Mtrans/cm^2	7	14-26	47	115	284	701
Chip size (mm^2)	170	170-214	235	269	308	354
Signal pins/chip	768	1024	1024	1280	1408	1472
Clock rate (MHz)	600	800	1100	1400	1800	2200
Wiring levels	6-7	7-8	8-9	9	9-10	10
Power supply (V)	1.8	1.5	1.2	0.9	0.6	0.6
High-perf power (W)	90	130	160	170	174	183
Battery power (W)	1.4	2.0	2.4	2.0	2.2	2.4

For Cost-Performance MPU (L1 on-chip SRAM cache; 32KB/1999 doubling every two years)

http://www.itrs.net/ntrs/publntrs.nsf

Industry Scaling Trends & Reliability Considerations

- Reduced gate oxide thicknesses
- Increased thermal/power densities
- Reduced interconnect dimensions
- Higher device operating temperatures
- Increased sensitivity to defects and statistical process variations
- Introduction of new materials with each new generation, replacing proven materials
  - e.g. Cu and low K inter-level dielectrics for Al and SiO2

Industry Scaling Trends & Reliability Considerations

- Dramatic increase in processing steps with each new generation
  - approx. 50 more steps per generation and a new metal level every 2 generations
- Rush to market - Less time to characterize new materials than in the past
  - e.g. reliability issues with new materials not fully understood and potential new failure modes
- Manufacturers' trends to provide 'just enough' lifetime, reliability, and environmental specs for commercial & industrial applications
  - e.g. 3-5 yr product lifetimes, trading off 'excess' reliability margins for performance

Industry Scaling Trends & Reliability Considerations

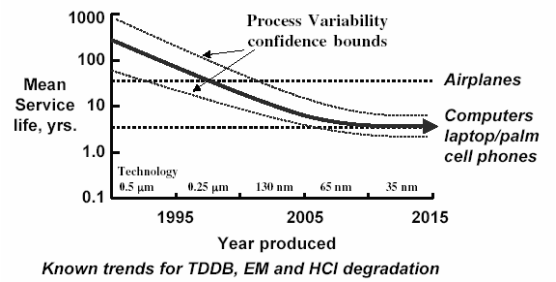
- Significant rise in the amount of proprietary technology and data developed by manufacturers, reluctance to share information with hi-rel customers
  - e.g. process recipes, process controls, process flows, design margins, MTTF
- Next generation microelectronics focus on the performance needs of the commercial customer, with little or no emphasis on the needs of the space customer
  - e.g. extended life, extreme environments, high reliability
- Increasingly difficult testability challenges due to device complexity

Product Technical Trends

	1990	2000	2010
Operating temperature, °C	-55 to 125	-40 to +85	0 to 70
Supply voltage	5v	1.5v	0.6v
Max. power (high perf.)	5	100	170
No. of package types	<10	<60	??
Design support life	>10 yrs.	1-5 yrs.	<1yr.
Production life	>10 yrs.	3-5 yrs.	<3yrs.
Service life	*>20 yrs.*	*5-10 yrs.*	*<5yrs.*

*MRQW-2002, Bernstein

Commercial Chip Reliability Estimation



Known trends for TDDB, EM and HCI degradation

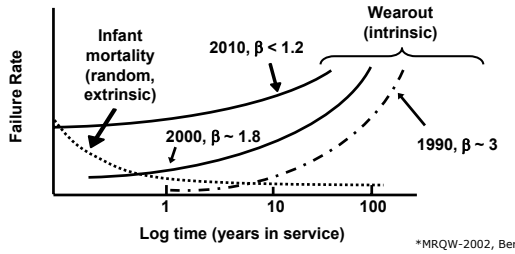
*Extrapolated from ITRS roadmap, MRQW-2002, Bernstein

Impact of scaling on wear-out failure mechanisms

- Dominant Failure Mechanisms
  - Electromigration (EM)
    - Migration of atoms in a conductor
  - Hot Carrier Injection (HCI)
    - High energy carriers degrade oxide
  - Negative Bias Temperature Instability (NBTI)
  - Time-Dependent-Dielectric-Breakdown (TDDB)
    - Oxide breakdown: Formation of a conduction path through gate oxide

Device Reliability Trends

As technology progresses, wearout failures become statistically indistinguishable from infant mortality failures with the same wearout drivers.



*MRQW-2002, Bernstein

Some information on the following slides: © George Candea

Software

- Is software getting worse?
  - Tandem OS (1985): 4 MLOC
  - Linux (2001): 30 MLOC (kernel 2.4 MLOC)
  - Windows XP (2001): 40-50 MLOC
  - Jim Gray's estimate: 1 bug/KLOC
  - Reducing bugs/KLOC vs. increasing KLOCs/product

Failures

- Hard to pinpoint a single root cause:
  - Coca-cola → disk crash → database failure
- Software bugs are faults!

Types of Bugs

- **Heisenbug**: disappears (or manifests differently) when you try to debug it ("Uncertainty Principle")
- **Bohrbug**: constant, reproducible, easy to deal with ("Bohr's atomic model")
- **Schrödingbug**: only starts manifesting when someone realizes it should be there ("Cat thought experiment")
- **Mandelbug**: underlying cause is so complex and obscure, it makes the bug seems nondeterministic ("Mandelbrot set")

Duration of Failures

- Permanent failure: once it manifests, won't go away unless you repair the system
  E.g., cut a network cable
- Intermittent failure: only occurs on occasion, for unknown reasons (until debugged… often workload)
  E.g., Patriot missile defense
- Transient failure: if you wait or retry, goes away
  E.g., various media corruption

Software Failures

- crash
- hang
- respond correctly but too late
- provide wrong data
- how to classify ? (fail-stop, fail-fast, Byzantine)
- how does recovery affect classification ?

Bug Triggers

- Timing
  - interleaving of events → many execution traces
  - hard to test all
- Recovery code
  - deals with exceptions → hard to simulate prior to shipping (ex. check NULL on return from malloc())
  - fault injection often used
- Third-party code
  - customer software, drivers, extensions, library users
  - Microsoft's "driver certification" → a way to combat this
- Boundary conditions
  - simple ones found through static analysis, complex ones are hard
- Bug-fix patches
  - customer system diverges over time
  - OS patches particularly evil

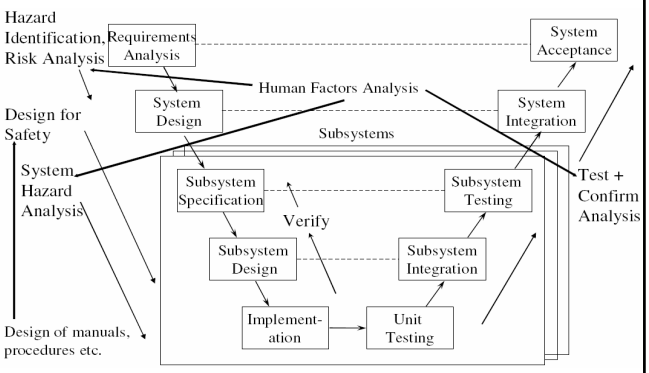
A Solution

- In the Web community: high availability is achieved via three-tiered model:
  - Reliable back end (databases)
  - Stateless middle tier (application servers)
  - Front end (web servers)
- Other communities?

Human Factors

- The role of humans in safety-critical systems
- Human Reliability Analysis
  - task analysis
  - human error identification
    - human error model: Reason
  - human reliability quantification
  - mitigating human error
- Safe user interface design

Human Factors



Have we learnt since Therac-25

Software for Certain Medtronic Implanted Infusion Pumps Recalled

FDA Patient Safety News: Show #32, October 2004

- Medtronic is recalling certain software application cards.They're used in the company's Model 8840 N'Vision Clinician Programmers. These hand-held devices are used to program a number of implantable devices, including the SynchroMed and SychroMed EL implantable infusion pumps.
- The recall is prompted by reports of data entry errors that have led to serious drug overdoses, including two patient deaths. The overdoses occurred when clinicians who were programming the pump entered the wrong time duration or the wrong interval --- for example, mistakenly putting the time interval between periodic drug boluses in the "minutes" field, instead of the "hours" field.

Have we learnt since Therac-25

- The recalled software may have contributed to these errors because one part of the screen did not have labels on the fields for hours, minutes, and seconds. Medtronic is now distributing replacement software that adds time labels to the screen to help reduce the risk of these kinds of programming errors.
- If you use the Model 8840 N'Vision Programmer with SynchroMed or SynchroMed EL infusion pumps, the company says you should pay particular attention to selecting the appropriate time field when entering time duration or time intervals. You should also be sure to check your software application card for your N'Vision Programmer. If you have the older software version (AAA 02), Medtronic says you should order the new version (AAD 02).

Automation

- A driving force of automation is to compensate for human disadvantages
  - humans are unreliable components of systems requiring replacement by reliable computers
  - humans have limited capabilities in response time and capacity
- However, humans play an essential role in safety-critical decision making
  - computers are not flexible or adaptable, e.g., response in emergency situations
  - computers cannot make creative judgements or strategic decisions

Human Error and Risk

- Automation yields
  - Increased capacity and productivity
  - Reduction in manual workload and fatigue
  - Increased safety
- But
  - Need specialised training
  - Cost of maintenance
- Impact on human operators
  - Unclear if overall workload reduced
  - Increased complacency due to overconfidence?

Role of Humans

- **Monitor**: detecting errors
  - it may not be possible to determine if an error has occurred
  - the system may provide inadequate feedback
  - operators may become complacent
- **Backup**: in an emergency
  - operators may become de-skilled
  - information provided may be inadequate for intervention
  - automated systems are usually too complicated

Role of Humans

- **Partner**: responsible for part of a task
  - humans may be assigned "hard to automate" part
  - humans may be responsible for monitoring and maintaining
  - division of responsibility may make building a mental model harder

Do Humans Cause Most Accidents?

- 85% of work accidents are due to **unsafe acts by humans** rather than unsafe conditions
- Should we believe the statistics?
  - Data may be biased and incomplete: in 60-80% of accidents caused by operator's loss of control, 75% of those had system/safety malfunction that preceded the operator action
    - e.g. DC-10 crash deemed pilot error, involved autopilot headings alteration without telling the crew
  - Positive actions are not usually recorded
    - only 10% of recovery from emergency are pilot errors
  - Operators are expected to always recover from emergency
    - Error can be due to poor design

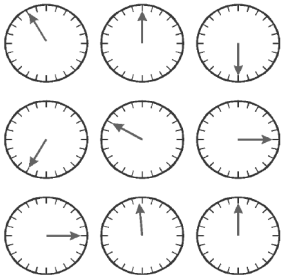
Do Humans Cause Most Accidents?

- Should we believe the statistics?
  - Operators have to intervene at limits, diagnose/respond quickly
    - E.g. consequences can be serious
  - Hindsight allows to identify a better decision
    - Operator's knowledge may be partial, or understanding erroneous
  - Separating operator error from design error is difficult
    - Examples from nuclear power plants:
      - Dials measuring the same quantities calibrated in different scales
      - Location of critical decimal points unclear
      - Critical displays located at back panels
      - Labels/colours inconsistent and misleading

What are humans good at?

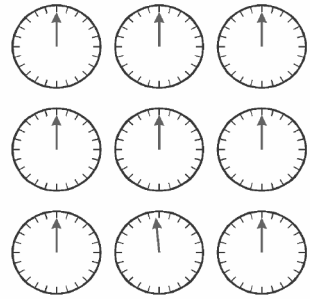
- Detecting correlations and exceptions
  - Patterns/clusters in graphical data
  - Breaks in lines
  - Visual/sound disturbances
- Detecting isolated movement
  - Waving
  - Flashing lights
- Detecting differences
  - Sounds, alarms, etc
  - Lights on/off
  - etc.

Example of Dial Controls



- **Bad interface**, cannot tell normal from abnormal.
- Advice is to fix normal at 12 o'clock position.

Example of Dial Controls



- **Good interface**: can spot abnormal position even for 5 deg change

Humans vs Machines

- Where machines have advantage…
  - Sensing/Actuating: broader range of sensors, able to perform in harsh environments
  - Cognition: no boredom, precision of calculations, repeatability, predictability
- Where humans have advantage…
  - Sensing/Actuating: image processing, edge & anomaly detection, flexibility
  - Cognition: ability to respond in unknown situations
- Should you trust humans or machines?
  - Boeing trusts people (pilot has ultimate authority).
  - Airbus trusts machines (flight control software has authority over pilot).

Human Machine Interaction (HMI)

- Hybrid discipline: psychology, engineering, ergonomics, medicine, sociology, mathematics
- Concerned with the impact of human operators and maintainers on system performance, safety and productivity
- Concerned with enhancing the efficiency, flexibility, comprehensibility and robustness of user interaction
- In the safety-critical context, the primary concern is to enhance robustness, possibly at the expense of efficiency and flexibility

Human Reliability Analysis (HRA)

- Identify potential operator errors that may lead to hazards and reduce error where risk is sufficiently high
- Four steps:
  - **task analysis**: characterise the actions performed to achieve particular goals
  - **human error identification**: identify possible erroneous actions in performing a task
  - **human reliability quantification**: estimate likelihood of error
  - **mitigation of human error**: identify control options

Task Analysis

- Tasks are activities to transform some given initial state into a goal state, i.e., goal-directed
- Structured from sub-tasks and elementary actions
- Each elementary action is concerned with a manipulation to be performed upon an object in the task domain
- Procedures for
  - normal operation of the system
  - maintenance of the system
  - emergency situations
- Logical sequence of actions that the operator engages in and the detailed physical executions that the operator

Human-Task Mismatch

- Human error is not a useful term
  - Implies possible to improve humans
- Human-Task Mismatch better term
  - Erroneous behaviour inextricably connected to the behaviour needed to complete a task
- Tasks
  - Involve problem solving, decision making
  - Need adaptation, experimentation, optimisation
- Levels of cognitive control [Rasmussen's]
  - Skills-based behaviour (smooth sensory based)
  - Rule-based behaviour (conscious problem solving)
  - Knowledge-based behaviour (goal known, planning by selection, trial and error, etc)

Experimentaton versus Error

- Designer relies mostly on knowledge-based behaviour
- Operator employs all three
  - In training, from knowledge- or rule-based to skills based
  - In unfamiliar situation, use knowledge-based to develop rules-based
  - Needs to maintain knowledge-based throughout
- Experimentation
  - Test a set of hypothesis through mental reasoning
  - May be unreliable
- Human error
  - unsuccessful experiments, in unkind environment
- Design for error tolerance

Human as Monitor

- Monitoring, rather than active control
  - Responsible for detecting/repairing problems
- Humans perform badly…
  - Task may be impossible
    - Cannot check in real-time if computer performs correctly
  - Operator dependent on information provided
    - Too much or too little is bad
  - Information is indirect
    - System handles most functionality
  - Failures may be silent or masked
    - E.g. autopilot disengages
  - Tasks are such that lower alertness results
    - Mechanical, lack of stimulation, can act without noticing

Human as Back-up

- Emergency only, rather than active control
  - Expected to take appropriate action
- Good design is essential
  - Can lower proficiency and increase reluctance to intervene
    - Infrequent usage
    - Cognitive and physical skills decline in absence of practice
    - High skills often needed!
      - E.g. emergency shutdown of nuclear plant
  - Fault-intolerant systems may lead to larger errors
    - May fail in ways difficult to anticipate
  - Harder to manage in crisis
    - Not fully aware of the internal state
    - Computer support for decision making

Human as Partner

- Both humans and automated system assigned control tasks
  - Number of human tasks reduced
  - Must be planned appropriately
- Modes
  - Partial automation
  - Shared control (primary responsibility with humans, but computer continuously performs checks)
- Potential problems
  - Good mental models are important
    - Must know the system state
  - Good communication is essential
    - Clarity, correctness

Accident Models

- Reduce description of accident to a set of events and conditions
  - Used in investigations, for prediction, etc
- Domino models
  - Social environment
  - Fault of a person
  - Unsafe act or mechanical/physical hazard
  - Accident
  - Injury
- Chain-of-events
  - Event trees, fault trees
- System theory
  - Accidents result from complex interactions

Human Tasks

- Simple tasks
  - Uncomplicated sequences
- Vigilance tasks
  - Detection of signals
- Emergency response tasks
  - May involve complex reactions
  - Performed under stress
- Complex tasks
  - Defined tasks, involve decision-making

Human Error Models

- Cognitive, e.g. Reason's model eight primary error groups
  - False sensation (lack of correspondence between subjective experience and reality)
  - Attentional failures (distraction, dividing attention)
  - Memory lapses (forgetting items)
  - Unintended words/actions
  - Recognition failures (wrongly observed signals)
  - Inaccurate and blocked recall (misremembering sequences)
  - Errors in judgement (misconceptions)
  - Reasoning errors (false deduction)
- Also Norman model of slips, mistakes in planning

Human-Task Mismatch again…

- Errors are an integral part of learning!
- Mechanisms of human malfunction
  - Skills-based level
    - Disorientation, motor skills failure
    - Stereotype take-over
  - Rule-based level
    - Incorrect recall of rules
    - Stereotype function
  - Knowledge-based level
    - Mental overload
    - Premature hypothesis (way of least resistence, point of no return)
- Also performance affecting factors (separately)
  - Work conditions, stress, social aspects

Human Factors Summary

- Understanding cognitive aspects essential
- Probability of failure difficult to predict
  - Human response affected by stress, fatigue, etc
- Must assume human error will happen sooner or later
  - Hardware support, failsafe operations
- Design for safety
  - Fault-tolerance
  - HCI (layout, communication, correctness etc)

Questions?

Gert Jervan

Department Of Computer Engineering
Tallinn University of Technology
Estonia