

Building safe systems

Mark Burgess
Oslo University College

1 hr

What's it about?

- Introduction
- Threats and strategies
- Analysis
- Interface design
- Risk reducing strategies
- ISO17799

Security – can't install it!

- *"Encryption solves 90% of the world's security problems"*
- Security is a property of complete systems
- We are talking about human–computer systems – not just technology
- Complexity means that if we fix one edge of the carpet, the other pops up...

Misplaced TRUST

- The basis of *everything* in security
- The things we trust are the things that can be used against us...
- Technology can shift the focus of trust, but not eliminate the need for it -- in the end it's our fault

TRUST vs Predictability

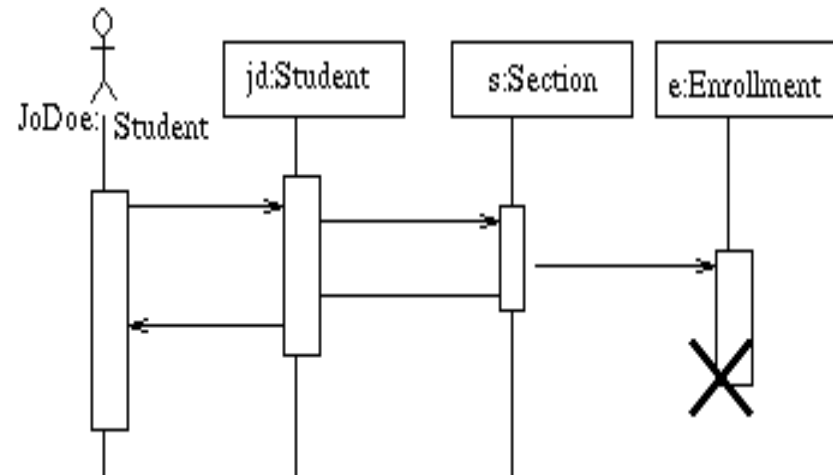
- We trust things that are predictable
- Humans are unpredictable
- Technology can be better...
 - "Simplest is best"
 - "Swap humans for simple mechanisms"
 - "Complexity is our enemy" (or is it?)

Example

- We don't trust neighbours: install locks
- We don't trust locks: install alarm
- Don't trust alarm: police camera
- Don't trust camera or police...stay home

UML

- Is about building trust in our ability to plan and implement



Fail predictably

- A well-designed system fails in a predictable way, and recovers.
- Some companies insist on source *code-review (open source software)*
 - *ISO17799*

The Plan

- Risk analysis
 - There are no objective criteria for safety/security
 - We need to make peace with the probable risks
- Risk aversion
 - Technology management
 - Trust management

Guidelines

- **RFC 2196**
 - "*Site Security Handbook*" (replaces RFC1244)
- **ISO/BS 17799**
 - "*Information Technology -- Code of Practice for Security Management*"
- **Orange Book**
 - "*Trusted Computer Security Evaluation Criteria (TSEC)*"

Threats

- Accidental Threats
 - Human error, spilled coffee...
- Attacks
 - Theft, spying, sabotage
- Pathogenic Threats
 - Virus, Trojan horse, bugs

Scales

- Human-computer interface
- Software functionality
- Algorithms
- Hardware
- Trust relationships (dependency)

Principles

- Maximize predictability
 - Simplicity increases predictability
 - Systems should fail predictably
- Restriction of privilege
 - Access on a "need to know/do" basis
 - Limit the scope of damage

Planning methods

- Formal methods
 - UML,
 - Petri nets,
 - Process algebras
- Checklists and protocols
 - Make behaviour more predictable
 - Standardized = predictable
- Contingency/recovery plan

Petri nets – protocols

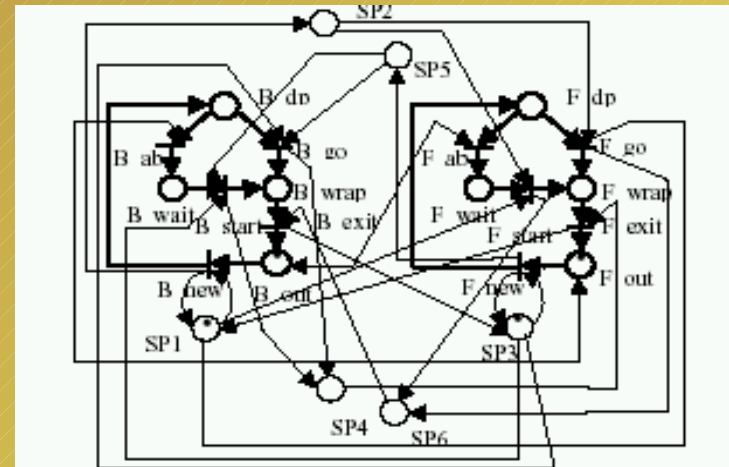
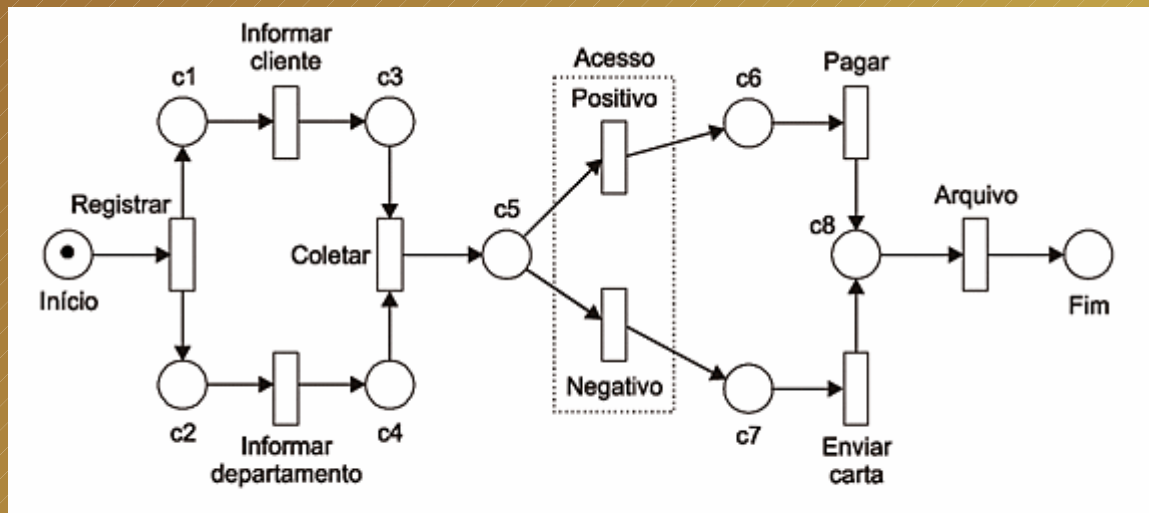
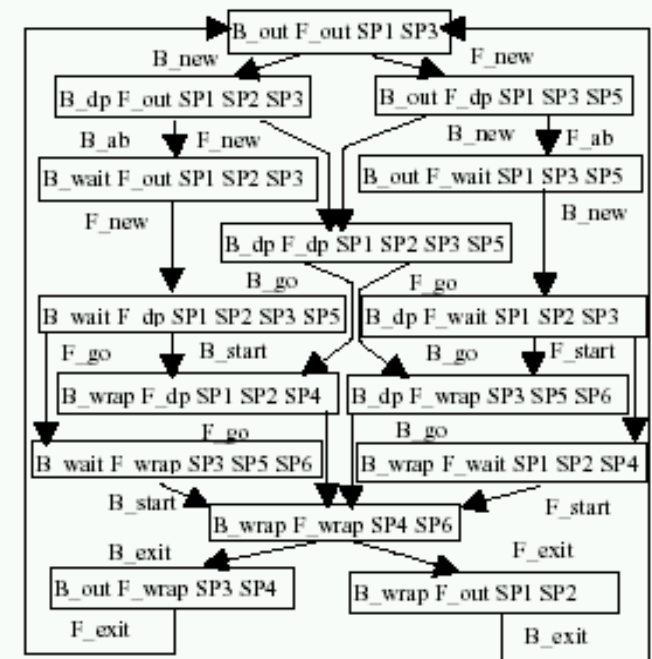


Fig. 6. Petri Net for the discrete part of the production line.



Parts of a system

- Input
 - Tricky, erroneous -- unexpected
- Rules
 - Safe, stable, predictable
- Output
 - Reliable, pass on responsibility

Definition of a secure system

"A secure system is one in which all the risks have been identified and accepted as a matter of policy."

This is the best we can hope for!

Who is the enemy?



Why do "enemies" expose us to risk?

- Greed
- To show off
- Bribery / blackmail
- Revenge
- Vandalism
- Opportunism
- Warfare (policy disagreement)

Why do "friends" expose us to risk?

- Forgetfulness
- Misunderstanding/miscommunication
- Confusion/stress/intoxication
- Ignorance / carelessness
- Slowness of response
- Procedural errors
- Inability to deal with complexity
- Inability to cooperate with others

And why can they do it?

- **Design faults**
 - System does not meet its specification
- **Emergent faults**
 - Behaviour that was not planned for or explicitly designed
- **Environment**
 - Provokes faults – high in entropy

Changing assumptions

- e.g. TCP/IP security
 - ARP authentication
 - Privileged ports
 - Sniffing, sequence numbers
- No one expected ordinary users to have access to these

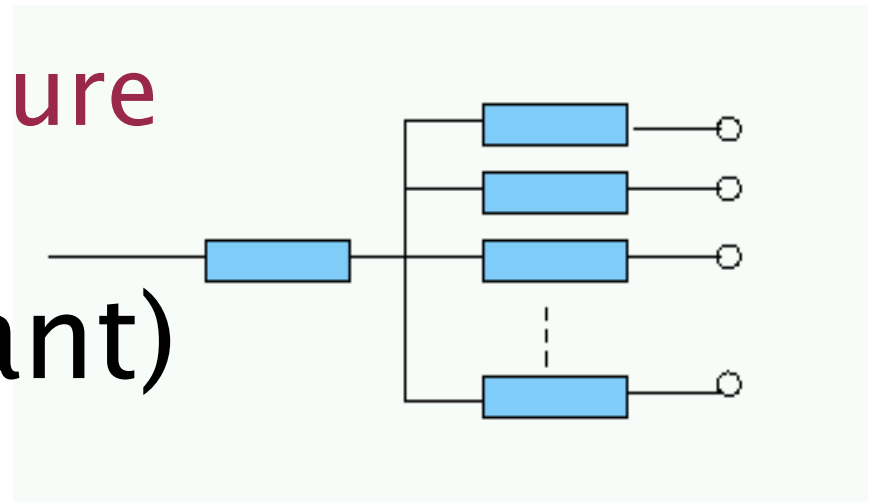
Points of failure

- Serial

- Single point of failure

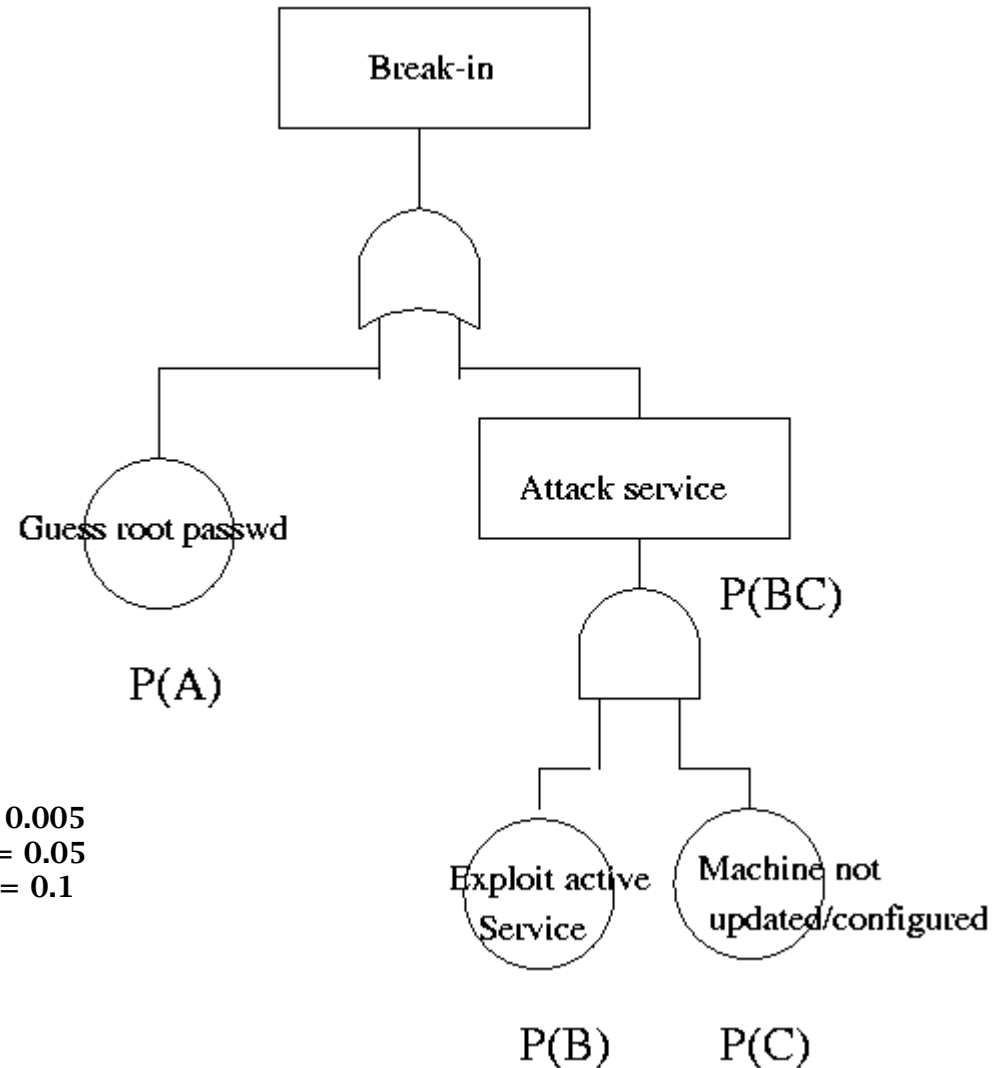
- Parallel (redundant)

- Resilience



Identifying points of failure

- Cause trees
- Fault trees



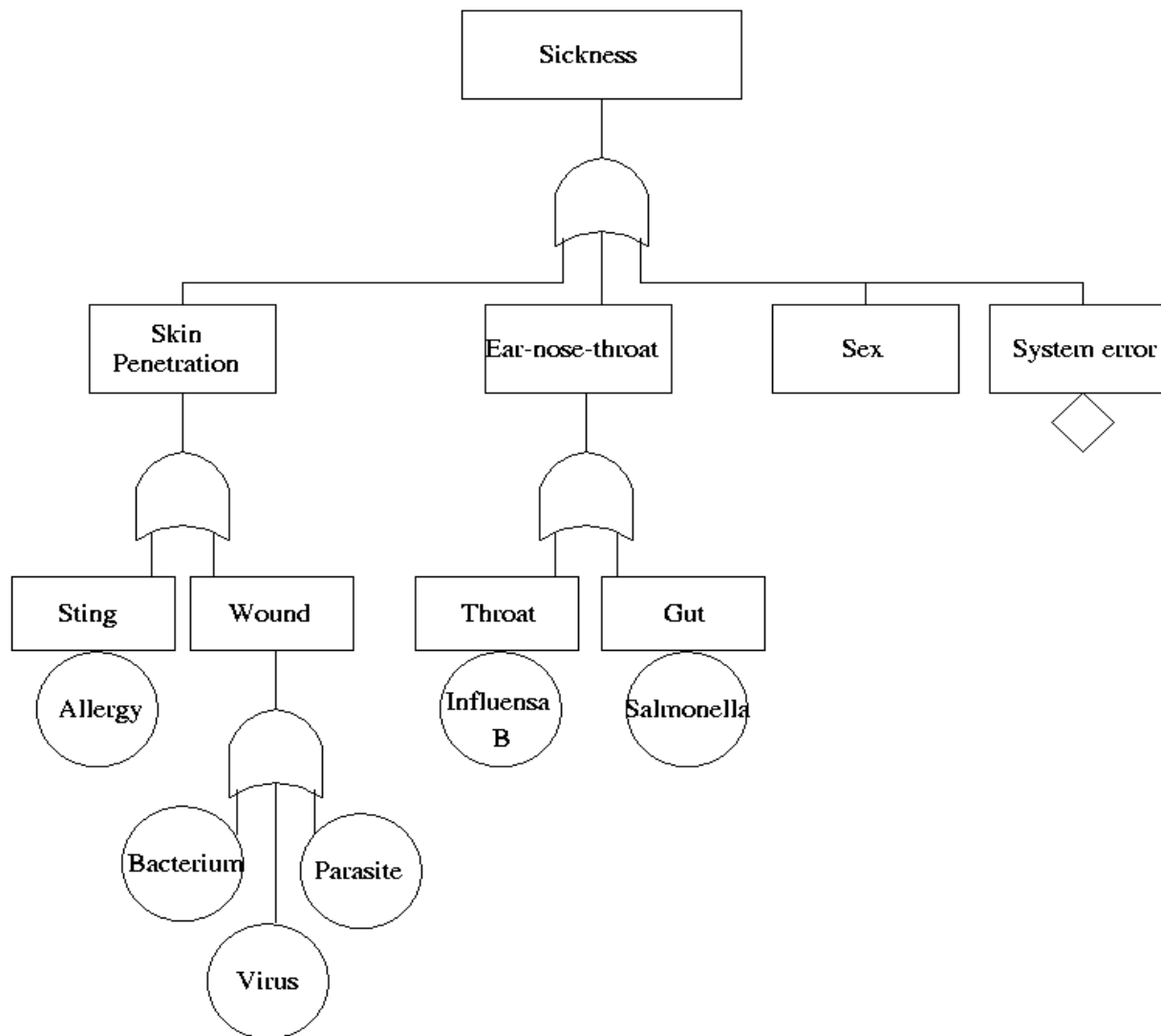
$$\begin{aligned} P(\text{break in}) &= P(A \text{ OR } (\text{NOT } A \text{ AND } (B \text{ AND } C))) \\ &= P(A) + (1 - P(A)) \times P(B)P(C) \end{aligned}$$

Suppose we have, from experience, that

Chance of guessing root pw $P(A) = 5/1000 = 0.005$
Chance of finding service exploit $P(B) = 50/1000 = 0.05$
Chance that hosts are misconfigured $P(C) = 10\% = 0.1$

$$\begin{aligned} P(T) &= 0.005 + 0.995 \times 0.05 \times 0.1 \\ &= 0.005 + 0.0049 \\ &= 0.01 \\ &= 1\% \end{aligned}$$

More FTA



Interface design: HCI

- Heuristics
- Make dangerous things difficult
- Consistency of behaviour
- Keep users awake and alert
 - Air traffic control
 - Power stations
- Fail safely



***Could this man be trusted with a
nuclear reactor?***

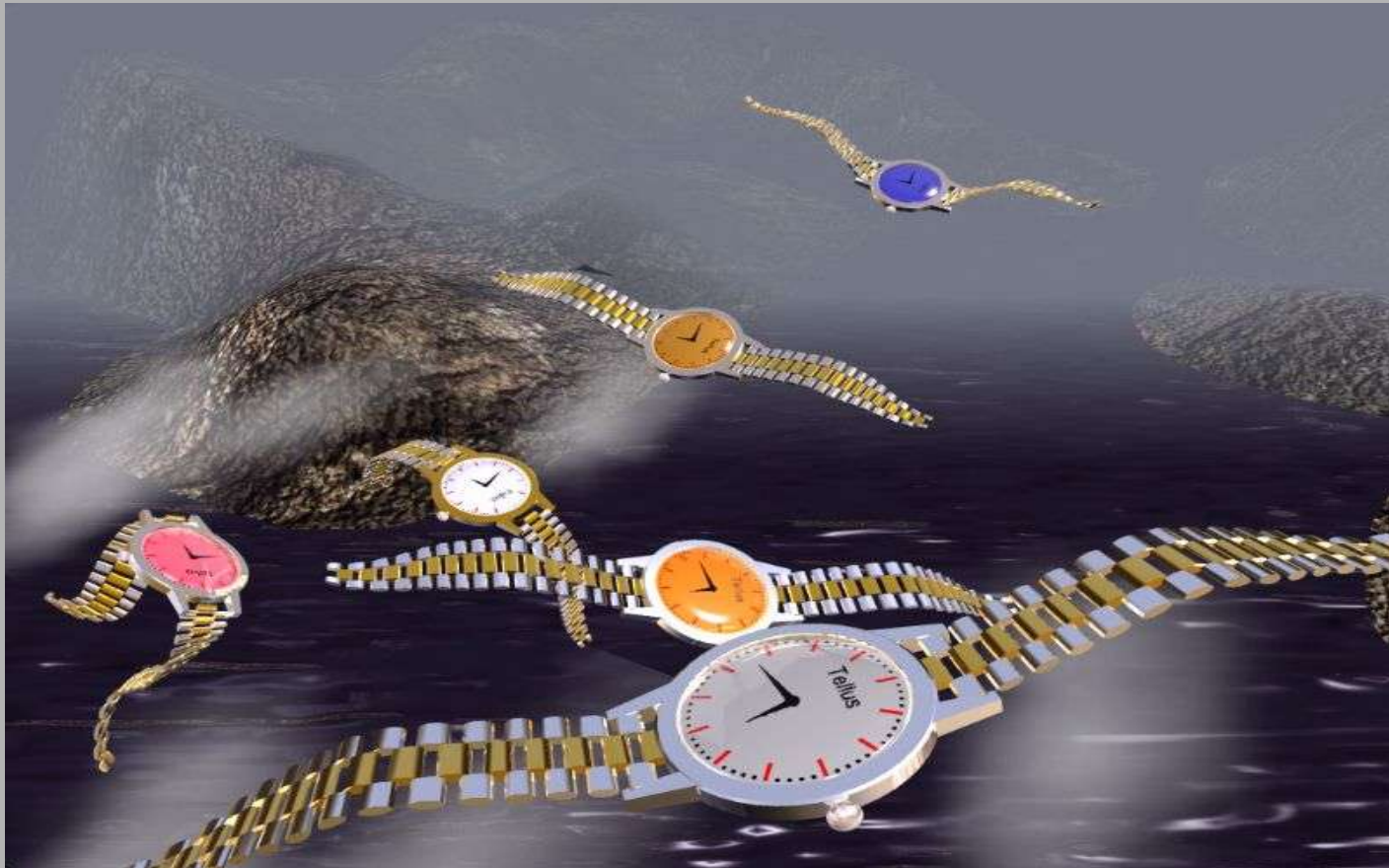


(Gentlemen prefer blondes...?)

Dumming down...

- Convenience society
 - We take things more for granted
- Less respect for technical knowledge
 - Mathematics
 - Formal methods
- Less interested in understanding how things work than before
 - Social sciences and media-studies dominate
 - Who will maintain the machines?

"Any sufficiently advanced technology is indistinguishable from magic"



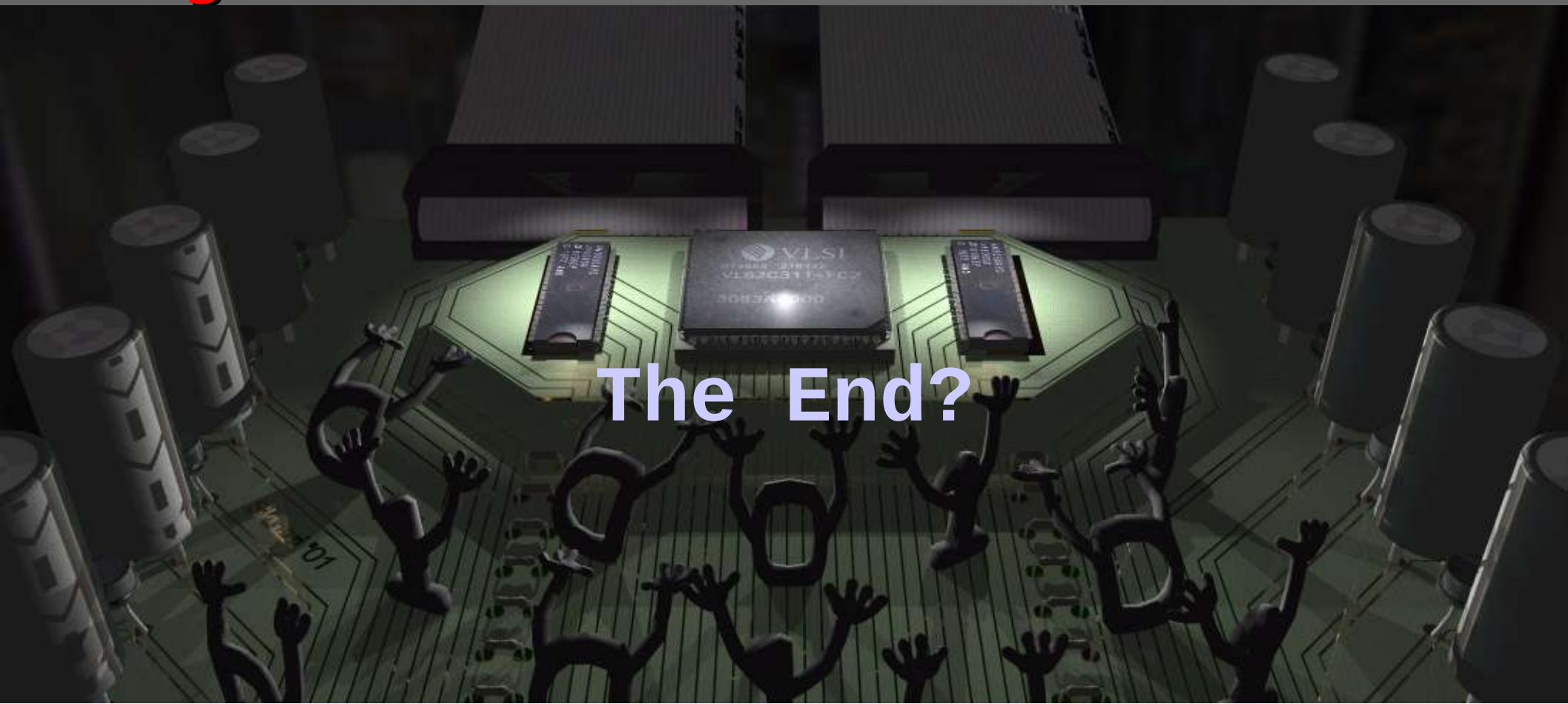
Summary

- Demanding predictability in systems.
- Formal methods – “calculate” possible faults, prevent and mitigate them

“The future should not be taken for granted. Our generations have the responsibility to convince future generations of the need for technical skill and education...”

Science is culture Technology is art

- Be, not just good technologists, but good citizens!



The End?

(end)

mark@iu.hio.no