Lero THE IRISH SOFTWARE RESEARCH CENTRE

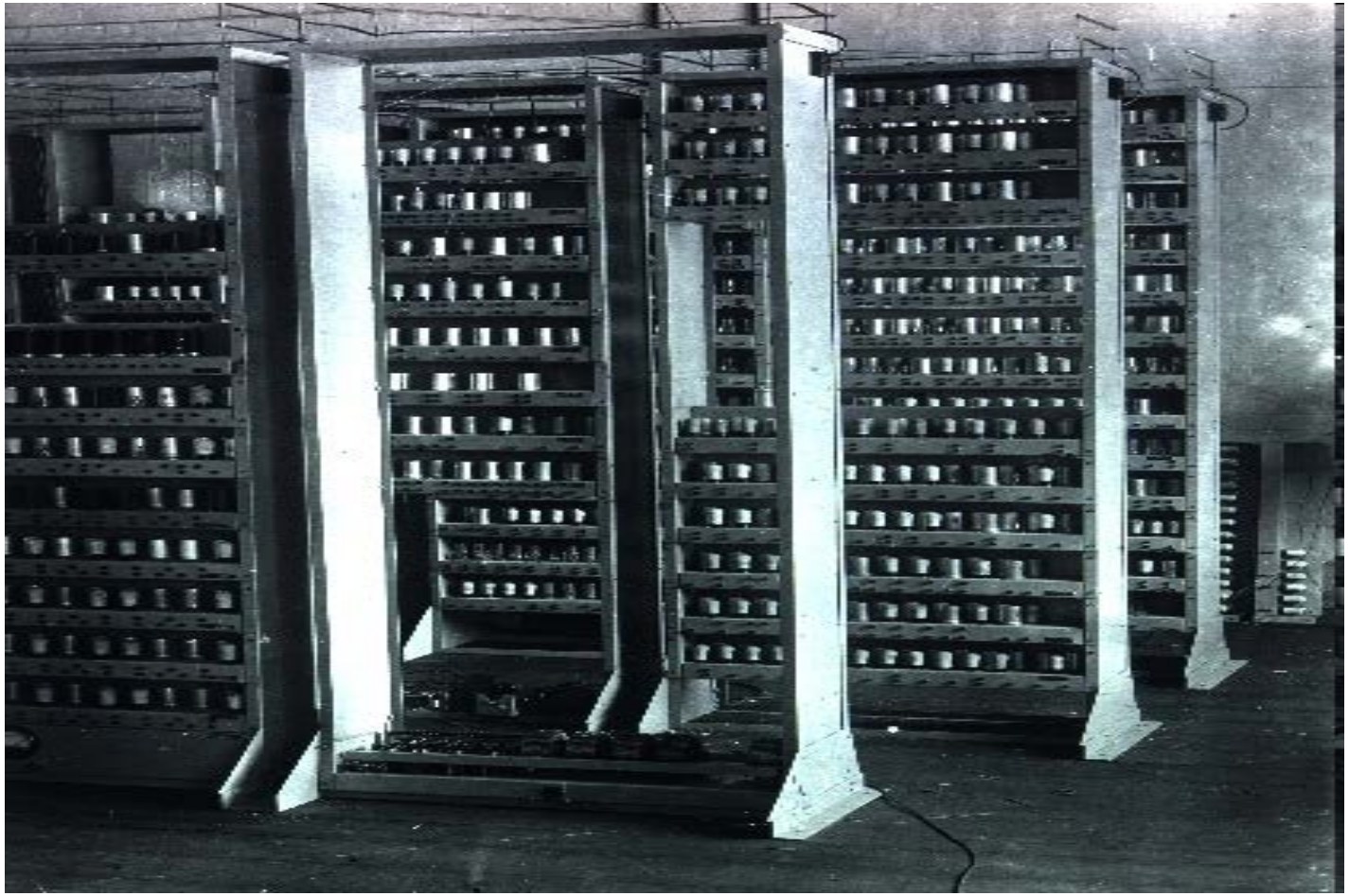
Is there anything that isn't software?

Prof. Mike Hinchey

Chair, IEEE UK & Ireland

University of Limerick Ollscoil Luimnigh · Trinity College Dublin · DCU · UCD Dublin · Oé Gaillimh NUI Galway · Dundalk Institute of Technology · UCC University College Cork, Ireland Coláiste na hOllscoile Corcaigh · Maynooth University National University of Ireland Maynooth

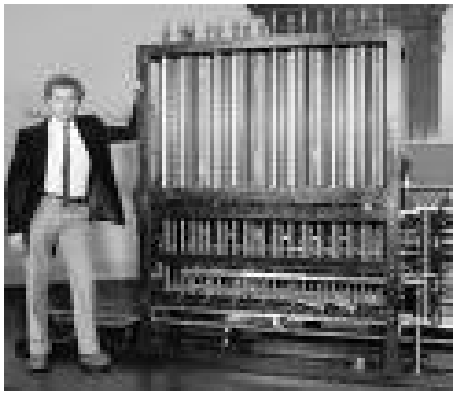
State of the Art (1949)

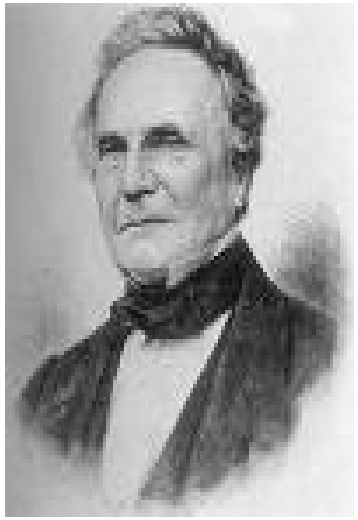


EDSAC

- 650 instructions per second.
- 1024 17-bit words of memory in mercury ultrasonic delay lines.
- Paper tape input and teleprinter output at 6 2/3 characters per second.
- 3000 valves, 12 kW power consumption, occupied a room 5m by 4m.
- "Operating system" occupied 31 words of read-only memory.
- Early use to solve problems in meteorology, genetics and X-ray crystallography.

Difference Engine





Motivation

Errata, detected in Taylor's Logarithms. *London: 4to, 1972 [sic]*

...

Kk Co-sine of 14.18.3 – 3398 – 3298

Nautical Almanac (1832)

...

In the list of ERRATA detected in Taylor's *Logarithms*, for cos. 4 18' 3'' read cos. 14 18'2''.

Nautical Almanac (1833)

ERRATUM of the ERRATUM of the ERRATA of TAYLOR'S *Logarithms.* For cos. 4 18'3'', *read* 14 18' 3''.

Nautical Almanac (1836)

First Programmer

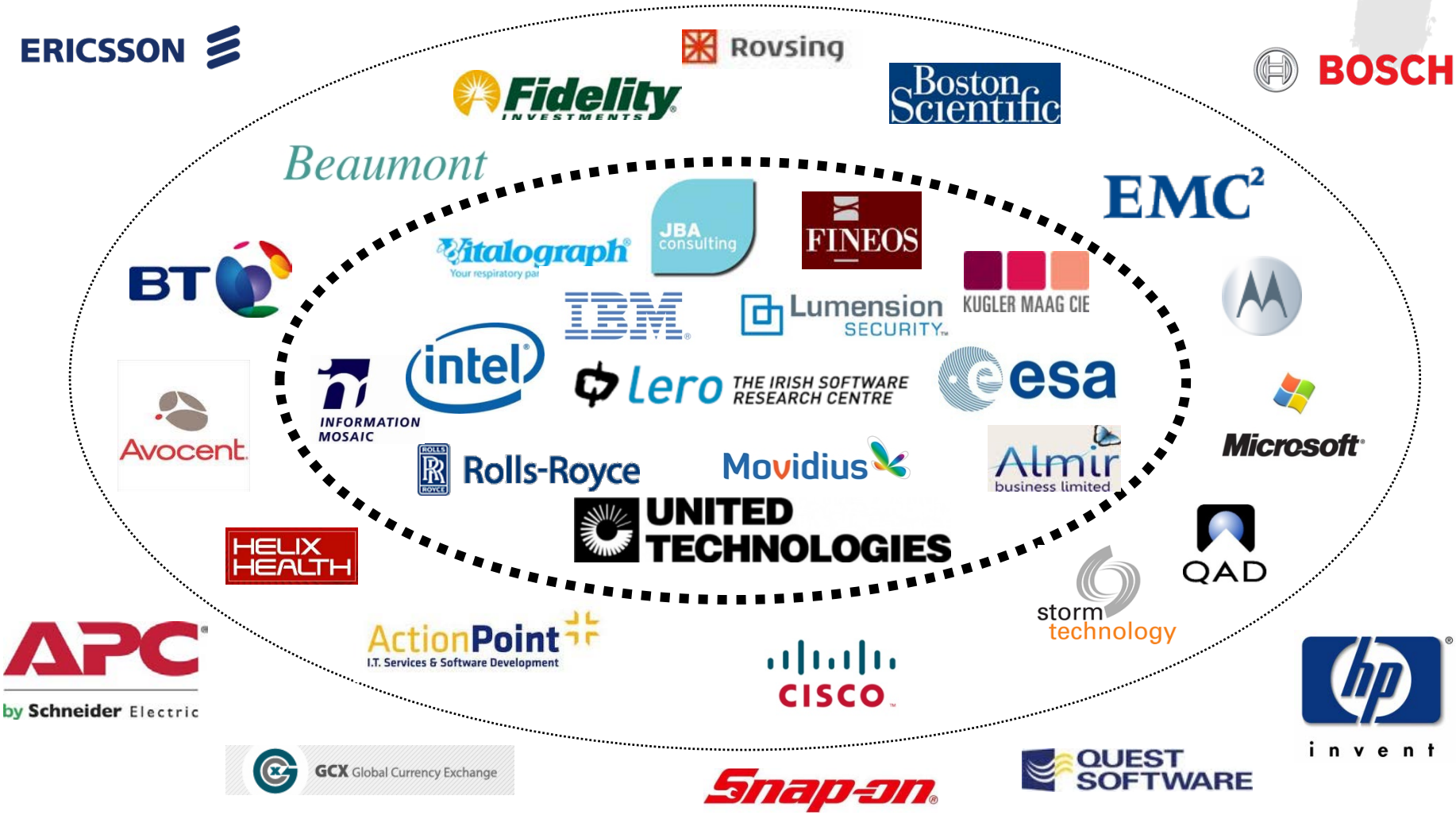
Augusta Ada King, Countess of Lovelace

Courtesy: Siobhan Clarke, Lero@TCD

Who develops Software?

INDUSTRY SECTORS	Critical skill	Necessary Skill
ICT Software	✓	✓
ICT Hardware		✓
Food		✓
Pharmaceuticals		✓
Financial Services	✓	
Engineering	✓	
Medical Devices	✓	
Telecommunications	✓	
Transportation	✓	
Energy	✓	
Business Services	✓	
Consumer/Retail		✓
Construction		✓
Media & Entertainment	✓	

Industry Network

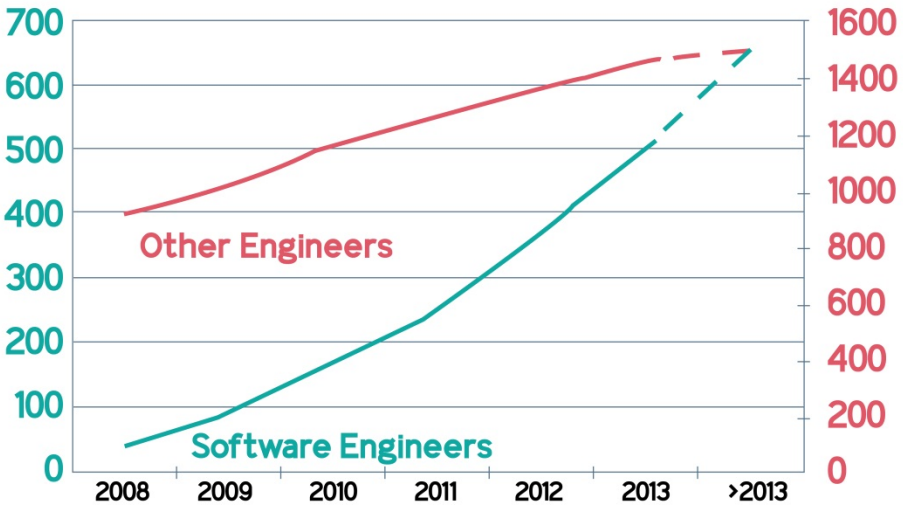


Software is Everywhere

- **Global ICT trends need advanced software capability**
  - Cloud, analytics & big data, cyber physical social systems, multicore, smart infrastructure, etc.
- **Software is pervasive**
  - Primary source of innovation & value creation in ICT, Med Tech, Financial Services, Manufacturing, Automotive
  - All companies becoming software companies - huge learning challenge

> Our company has become a software company. The trouble is our engineers don't realise it yet !
>
> - *VP Research **Analog Devices Inc.***

Growth in Software Engineers Employed at BMW Group



Source: BMW Group

Software Challenges

- **Producing high quality software is complex and difficult**
  - Not just coding (only **7%**)
  - Challenge increasing as systems become larger and more interconnected

The problem of predictable development of software with the intended functionality that is reliable, secure and efficient **remains one of the most important problems** in [ICT]

***– US President's Council of Advisors on Science & Technology** 2012*

- **Direct correlation between software process capability and business benefits**
  - Time-to-Market, Quality, Productivity, Regulatory Compliance
- **Software mistakes are expensive**
  - Toyota brake glitch **$3b**; Mars Orbiter **$655m**

Evolving Systems

Software is not static

At r… need to adapt an…

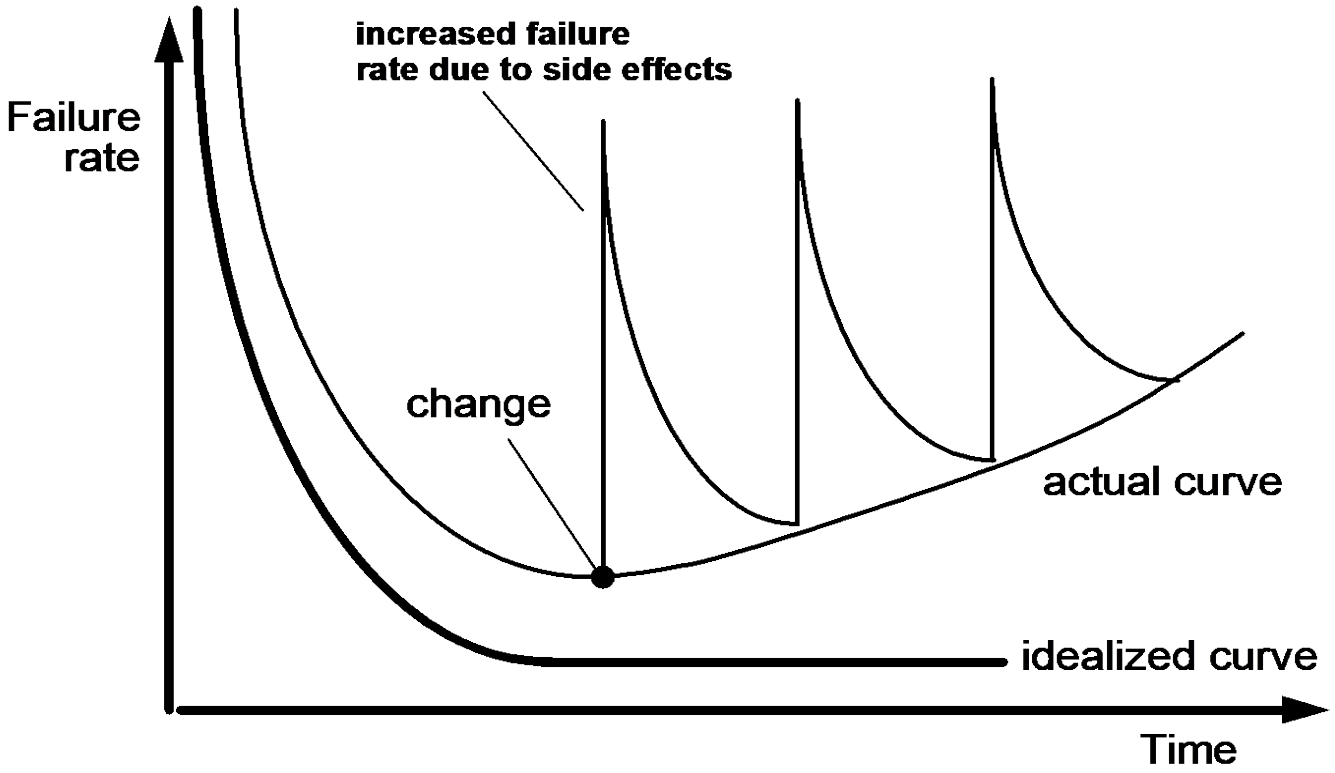
- to react to chang… environment;
- to meet necessary constraints on the system that were not previously satisfied and possibly not previously known;
- to protect the system from external threats.

Legacy systems are those that have evolved over longer timeframes, due to:

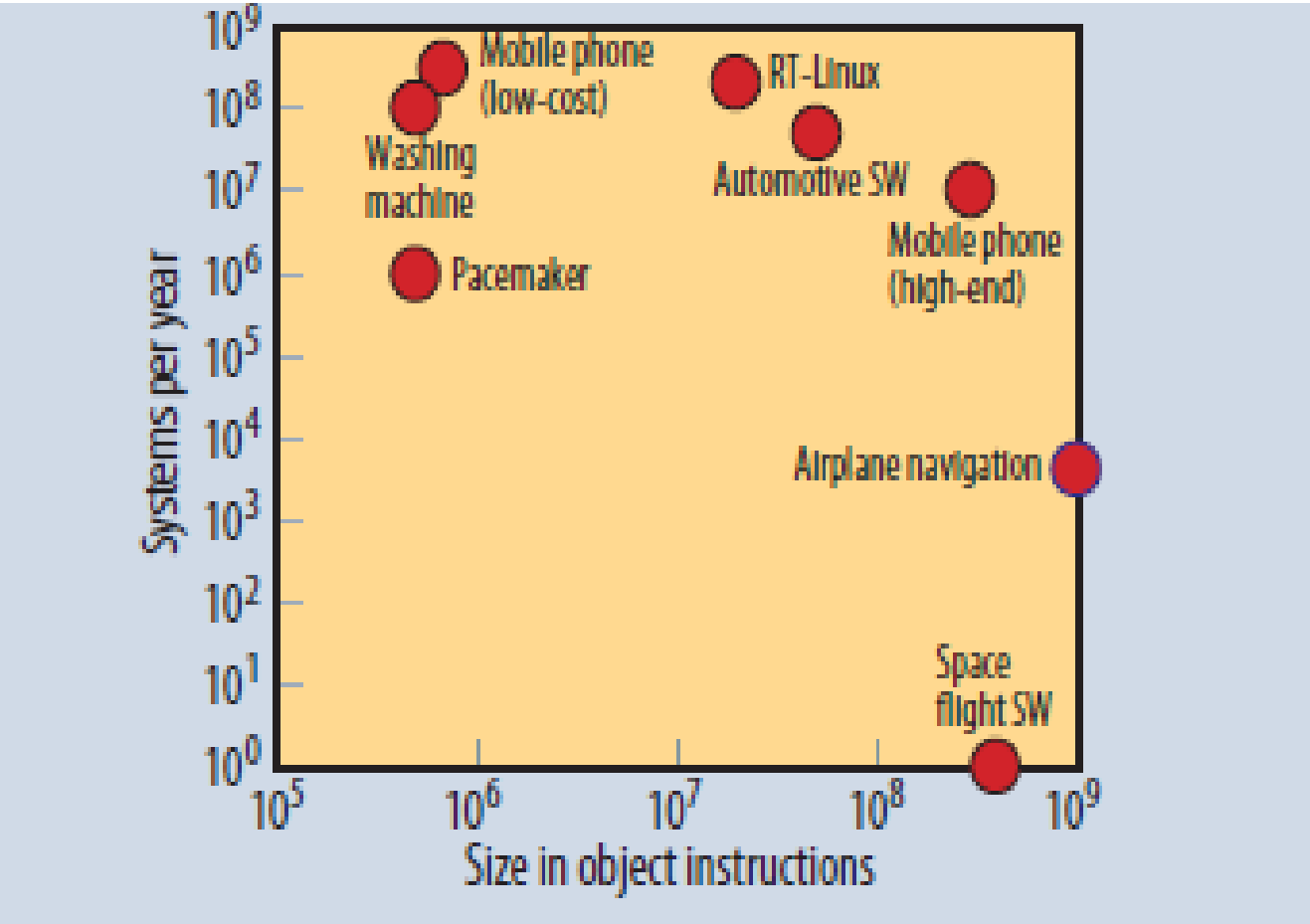
- …te systems being …ther;
- … technolog…
- new user requirem…
- new regulatory compliance requirements.

Software is supposed to change… otherwise it would be in the hardware!

Wear versus Deterioration

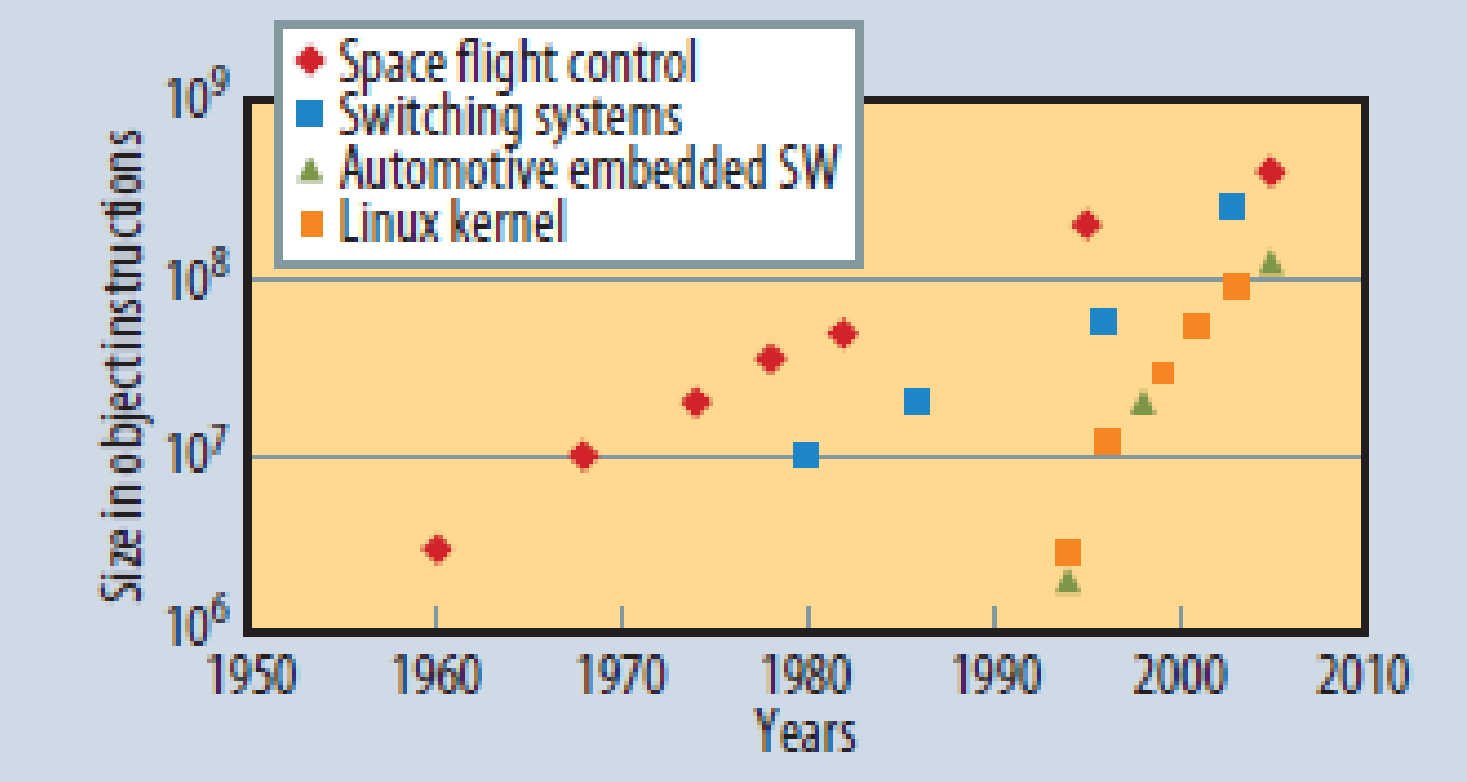


Size of Modern Applications



Source: Ebert & Jones, *Computer*, April 2009

Increasing Size



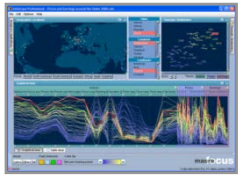
Source: Ebert & Jones, *Computer*, April 2009

Critical Systems

- Systems where failure or malfunction will lead to significant negative consequences.
- Strict requirements for security and safety to protect the user or others.
- Critical to the organization's mission, product base, profitability or competitive advantage.

Automotive Systems

Medical Devices



Financial / Enterprise Information Systems

Current Situation

- Software is pervasive, widely used, and often invisible.
- Much legacy code, badly structured, poorly maintained.
- Many software failures, declining quality:
  - E.g., Therac 25, ARIANE 5, Mars Polar Lander, … and many more!
- Complex physical environments and diverse hardware platforms.
- Insufficient number of qualified developers and testers.
- Current techniques do not scale sufficiently and have failed to overcome 50 years of declining quality.

Evolving Critical Systems

- have evolved from legacy code and legacy systems, or
- result from a combination of existing component-based systems, possibly over significant periods of time, or
- evolve as a result of a focused and intentional change in organization and architecture to exploit newer techniques believed to be beneficial;
- they require that the system adapt and evolve at run-time in order to react to changes in the environment or to meet necessary constraints on the system that were not previously satisfied and possibly not previously known.

Some Examples of Lero Research

1. Smarter Cities
   - In conjunction with Intel Labs Europe, Dublin City Council and IBM
2. Software Product Lines
   - Use of models to gain efficiencies
3. Adaptive Security and Privacy (Cloud, smart buildings)
   - In conjunction with United Technologies and IBM
4. Parallelisation of code to optimise use of multicore hardware
   - In conjunction with Movidius and IBM

Some Examples of Lero Research

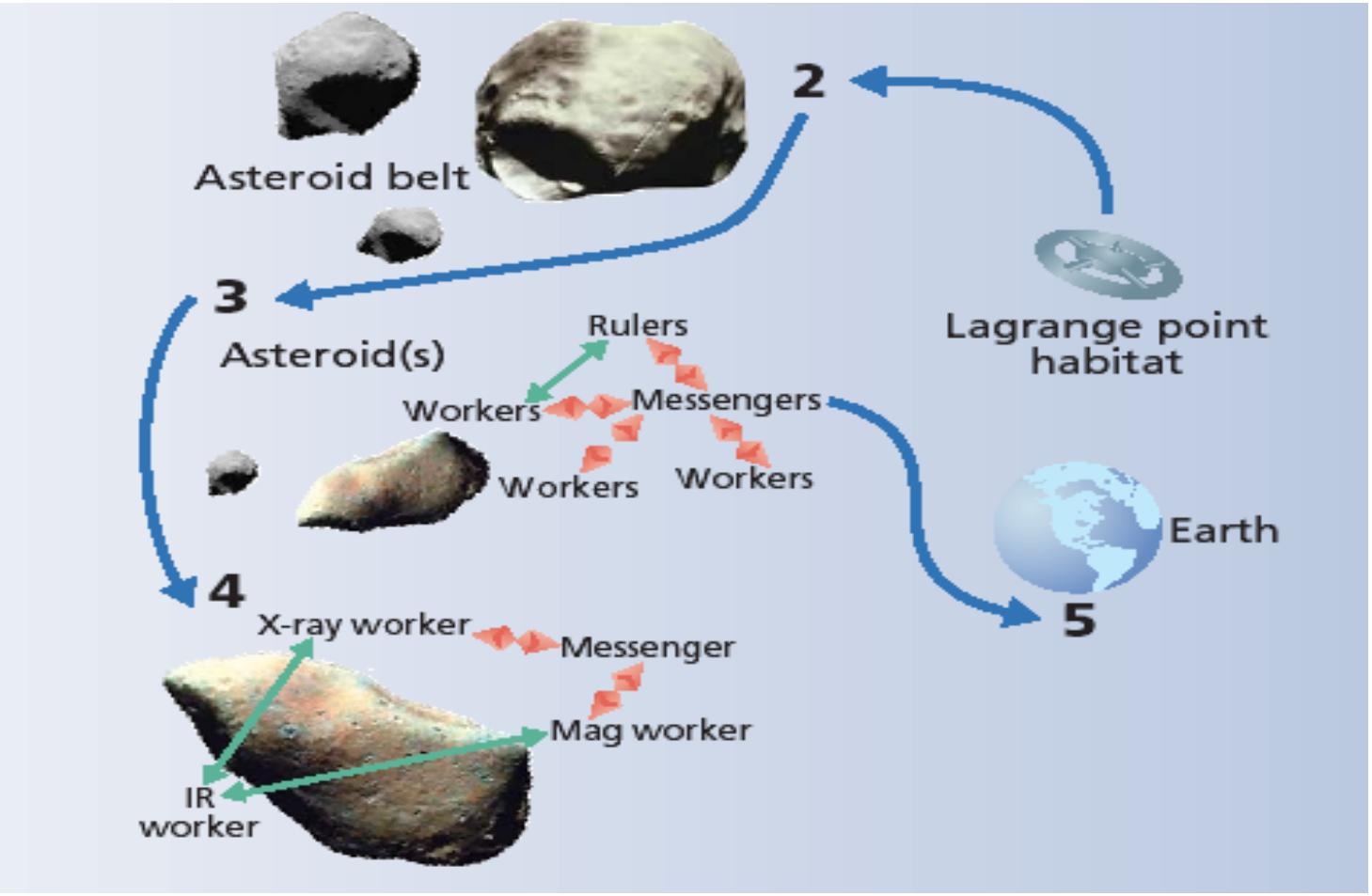
5. Architectural Recovery and Preservation
   - In conjunction with several financial services companies
6. Performance Evaluation in Large Systems
   - In conjunction with IBM
7. Autonomous Space Systems
   - In conjunction with NASA and ESA and EU FP7 Project ASCENS

Autonomous NanoTechnology Swarm

Using swarms of “intelligent”, autonomous spacecraft to explore

1. Lunar and Martian surface (Lander Amorphous Rover Antenna, LARA)
2. Saturn’s rings (Saturn Autonomous Ring Array, SARA)
3. Asteroid belt (Prospecting Asteroid Mission, PAM)

ANTS Concept Mission - PAM



Contributions

1. Formal Methods
2. Autonomic Computing
3. Software Product Lines
4. Automatic Code Generation

Model of Formal Method

V

Swarm Formal Method Model and Outline

$$LEADER\_COM_{i,com} = leader.in?\,msg \rightarrow$$
$$\text{case } LEADER\_MESSAGE_{i,com,msg}$$
$$\quad if\ sender(msg) = LEADER$$
$$MESSENGER\_MESSAGE_{i,com,msg}$$
$$\quad if\ sender(msg) = MESSENGER$$
$$WORKER\_MESSAGE_{i,com,msg}$$
$$\quad if\ sender(msg) = WORKER$$
$$EARTH\_MESSAGE_{i,com,msg}$$
$$\quad if\ sender(msg) = EARTH$$
$$ERROR\_MESSAGE_{i,com,msg}$$
$$\quad otherwise$$

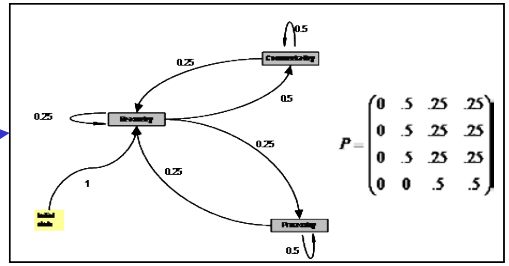
$$\Phi = \{SendMessage, ReceiveMessage, Reason, Process\}$$

is a set of (partial) transition functions where each transition function maps

$$Memory \times Input \rightarrow Output \times Memory$$

$$memory' = (Goals', Model', CommsTracking)$$

[Communicating]ReasoningDeliberatve(Leader)[Reasoning]
[Reasoning] SendMessage (Leader,Worker)[Communicating]
[Processing] SendMessage (Leader, Worker) [Communicating]



Agent State	Actions leading to the agent state	f	p
	Identity		
Communicating	SendMessageWorker	50	2
	SendMessageLeader	50	2
	SendMessageError	1	1
	ReceiveMessageWorker	50	2
	ReceiveMessageLeader	50	2
	ReceiveMessageError	1	1
Reasoning	ReasoningDeliberatve	50	2
	ReasoningReactive	50	2
Processing	ProcessingSortingAndStorage	17	2
	ProcessingGeneration	17	2
	ProcessingPrediction	17	2
	ProcessingDiagnosis	16	2
	ProcessingRecovery	16	2
	ProcessingRemediation	17	2

$$Communicating = 50\omega^2 : SendMessageWorker.Communicating$$
$$+ 50\omega^2 : SendMessageLeader.Communicating$$
$$+ 1\omega^1 : SendMessageError.Communicating$$
$$+ 50\omega^2 : ReceiveMessageWorker.Communicating$$
$$+ 50\omega^2 : ReceiveMessageLeader.Communicating$$
$$+ 1\omega^1 : ReceiveMessageError.Communicating$$
$$+ 50\omega^2 : Reasoning.DeliberativeReasoning$$
$$+ 50\omega^2 : Reasoning.ReactiveReasoning$$
$$+ 17\omega^2 : Processing.SortingAndStorage.Processing$$
$$+ 17\omega^2 : Processing.Generation.Processing$$
$$+ 17\omega^2 : Processing.Prediction.Processing$$
$$+ 16\omega^2 : Processing.Diagnosis.Processing$$
$$+ 16\omega^2 : Processing.Recovery.Processing$$
$$+ 17\omega^2 : Processing.Remediation.Processing$$

$$n\omega^{k+l} + m\omega^k = n\omega^{k+l} = m\omega^k + n\omega^{k+l}$$

$$n\omega^k + m\omega^k = (n+m)\omega^k = m\omega^k + n\omega^k$$

Specification

```
AEIP {
   MESSAGES { ... }
   CHANNELS { ... }
   FUNCTIONS { ... }
   MANAGED_ELEMENTS {
      MANAGED_ELEMENT worker {
         INTERFACE_FUNCTION getDistanceToNearestObject { RETURNS { DECIMAL } }
      }
   }
} // AEIP

METRICS {
   METRIC distanceToNearestObject {
      METRIC_TYPE { RESOURCE }
      METRIC_SOURCE { AEIP.MANAGED_ELEMENTS.worker.getDistanceToNearestObject }
      DESCRIPTION { "measures the distance to the nearest space object" }
      MEASURE_UNIT { "KM" }
      VALUE { 100 }
      THRESHOLD_CLASS { DECIMAL [0.001 ~ ) }
   }
}
```

Autonomic Computing

Inspiration from the human/mammalian autonomic nervous system.

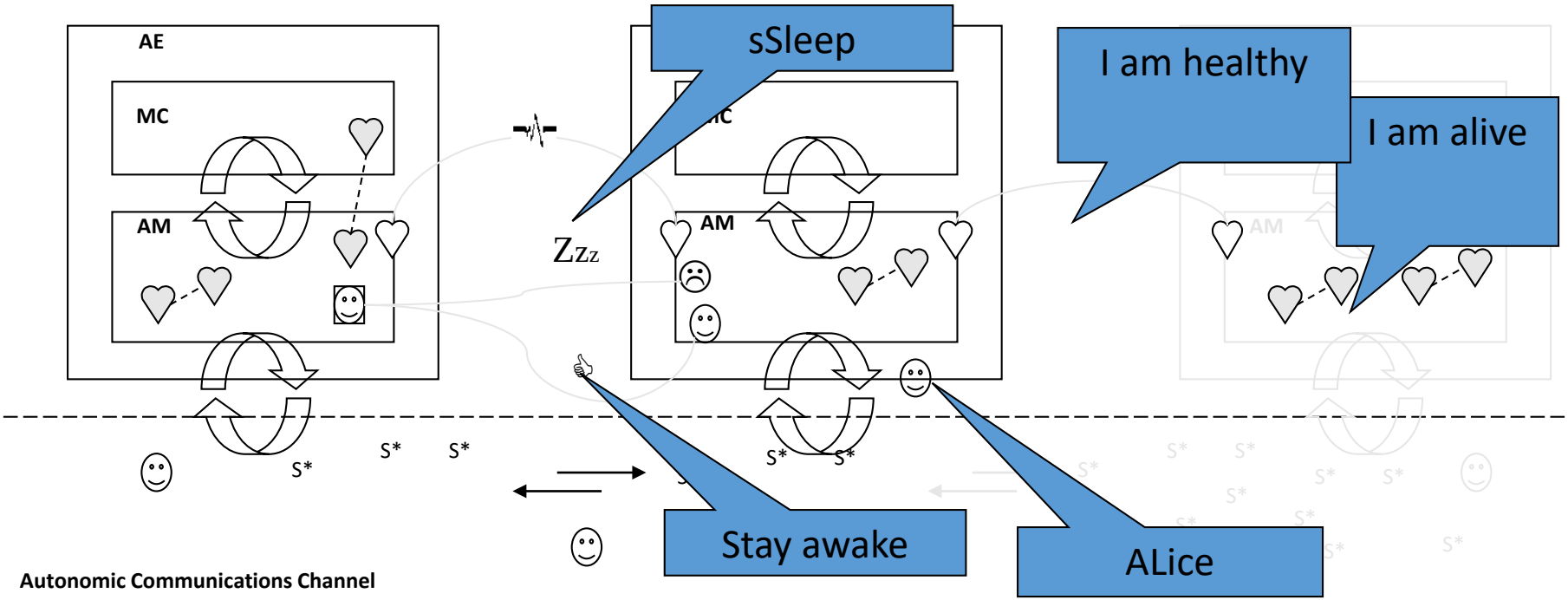
Fight or Flight

sympathetic
(SyNS)

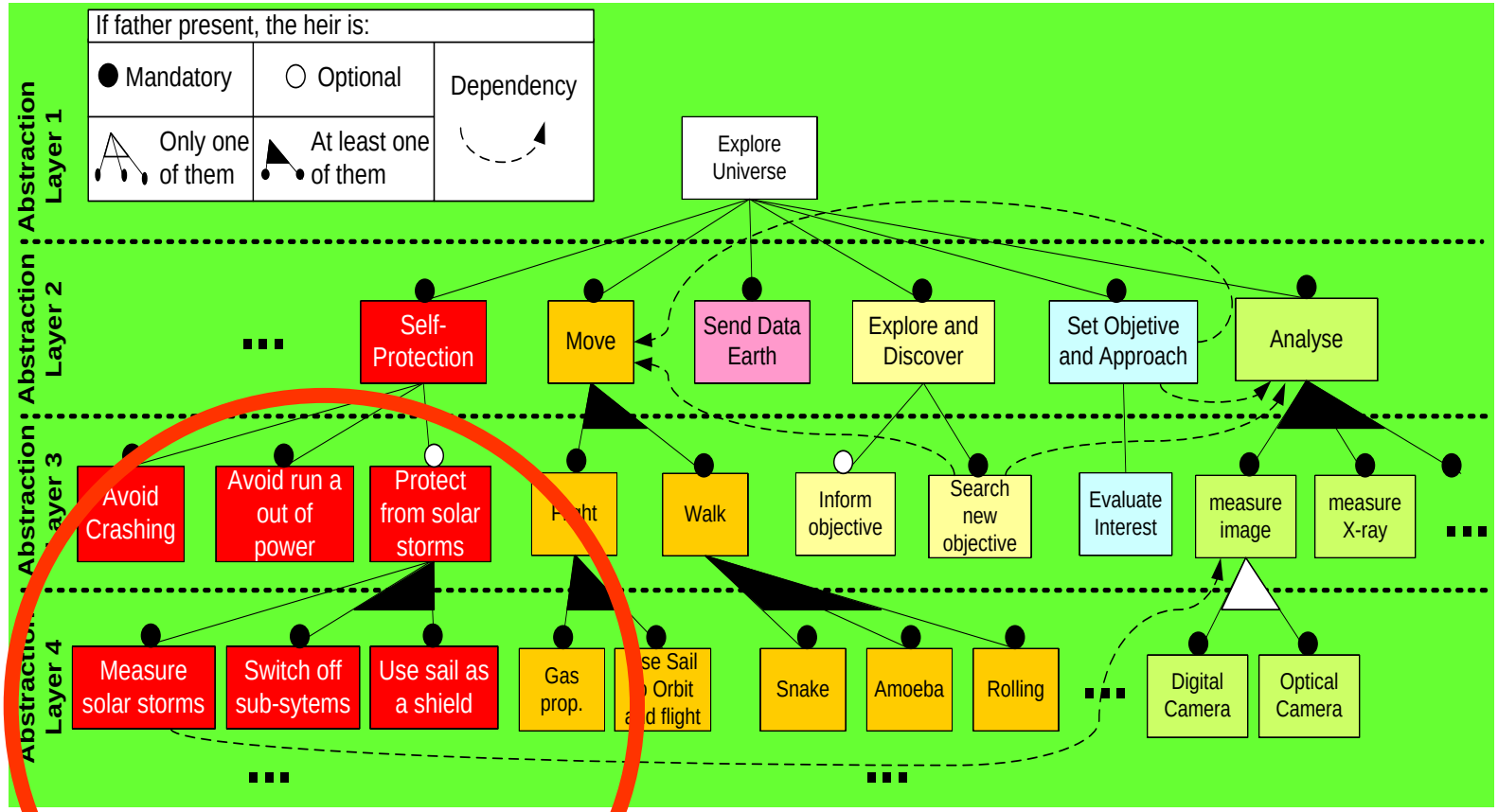
Rest and Digest

parasympathetic
(PaNS)

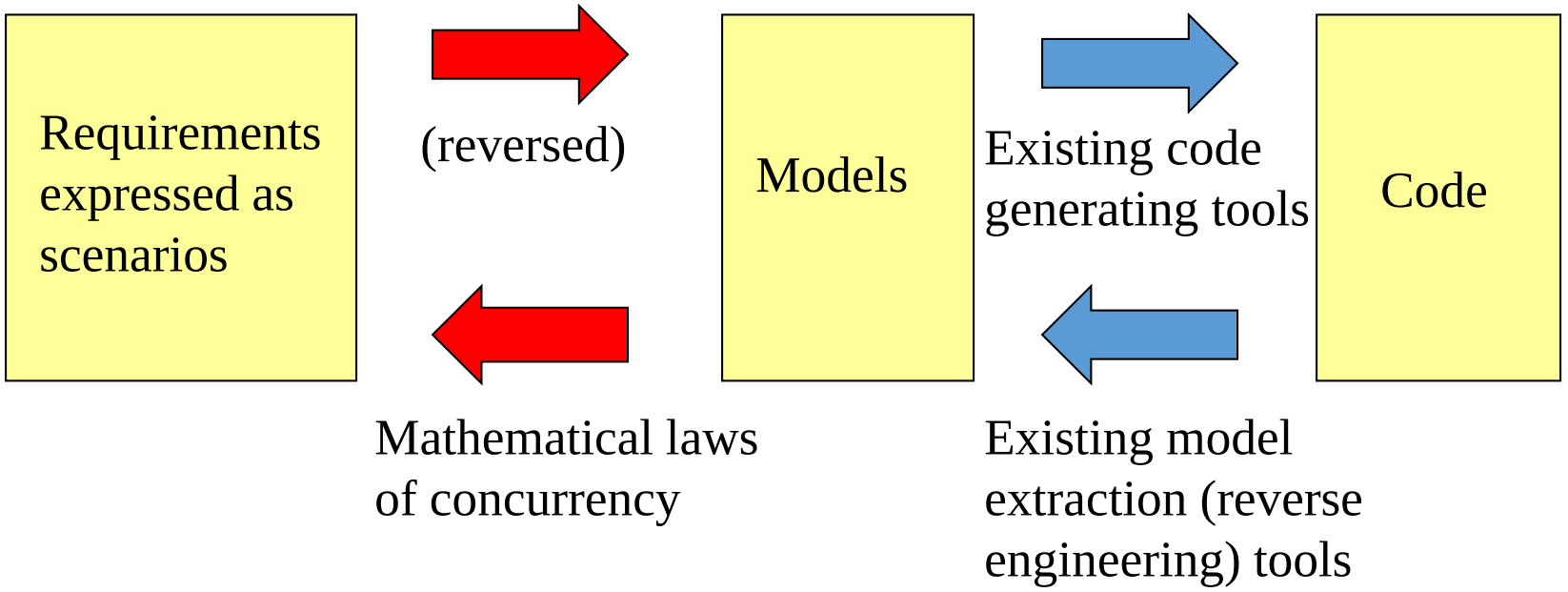
Autonomic Environment



SPL / Feature Model



Requirements to Design to Code (R2D2C)



Future Demand

- Demand for large, complex systems, is rising exponentially.
- Smart-grid, smart-cities, smart-buildings, smart-transportation, etc., all rely heavily on interconnected software-intensive systems.
- Ubiquitous systems, cloud, mobile devices etc., push demand for computing resources and applications, not just data.
- Autonomic Computing arose out of need to handle great amounts of data; what is needed for great amounts of code?
- World increasingly being seen as a *Cyber-Physical-Social* system.

Question

Q: Is there anything that isn't software?

A: Not really.

We need to....

- Migrate software development from a craft to a professional discipline with appropriately educated specialists;
- Develop new lifecycle models to enable the next generation of software development and higher level languages;
- Enable scalable development methods that address specific domains, organisations and processes;
- Enable active participation by customers in the software ecosystem and make software development customer-led (need-"pull" rather than technology "push");
- Develop a holistic approach to speedy development of highly-reliable software, able to express *physical*, *cyber* and *social* design objectives simultaneously.

Lero THE IRISH SOFTWARE RESEARCH CENTRE

Děkuji!
Go raibh maith agaibh!
Thank you!

Science Foundation Ireland
Ireland's European Structural and Investment Funds Programmes 2014-2020
Co-funded by the Irish Government and the European Union
European Union
European Regional Development Fund
SOUTHERN Regional Assembly
Promoting our Region