TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

Institut für Computertechnik
Institute of Computer Technology

Self Awareness and Resilience in CPS

Axel Jantsch

Testmethoden und Zuverlässigkeit von Schaltungen und Systemen

Freiburg, Germany, March 2018

Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

Outline

1. Motivation

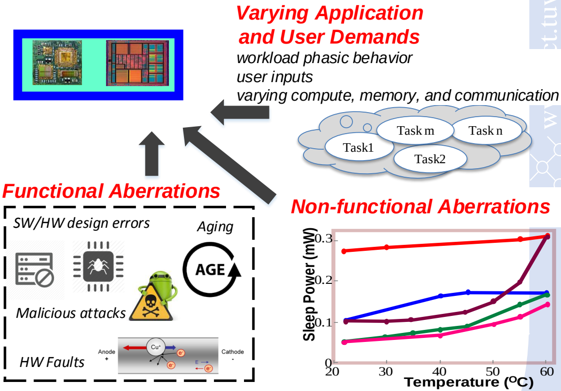
2. Concepts of Self-Awareness

3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management

4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score

5. Conclusions

The Problem



Varying Application and User Demands

workload phasic behavior

user inputs

varying compute, memory, and communication

Functional Aberrations

SW/HW design errors

Aging

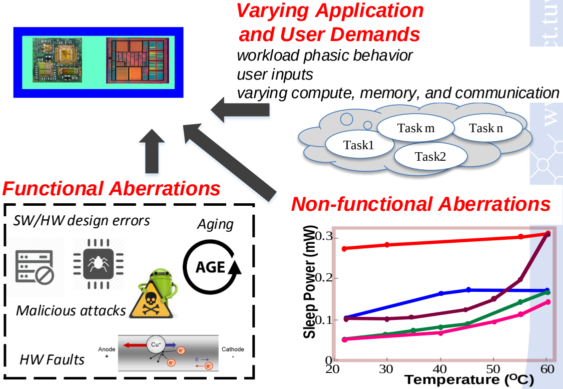
Malicious attacks

HW Faults

Non-functional Aberrations

The Problem

- Large number of resources



Varying Application and User Demands

workload phasic behavior

user inputs

varying compute, memory, and communication

Functional Aberrations

SW/HW design errors

Aging

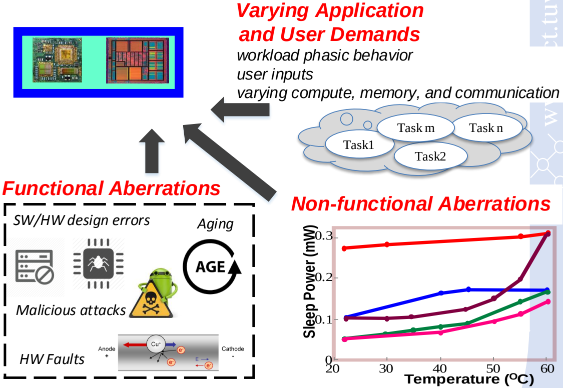
Malicious attacks

HW Faults

Non-functional Aberrations

The Problem

- Large number of resources
- Many tight constraints



Varying Application and User Demands

workload phasic behavior

user inputs

varying compute, memory, and communication

Functional Aberrations

SW/HW design errors

Aging

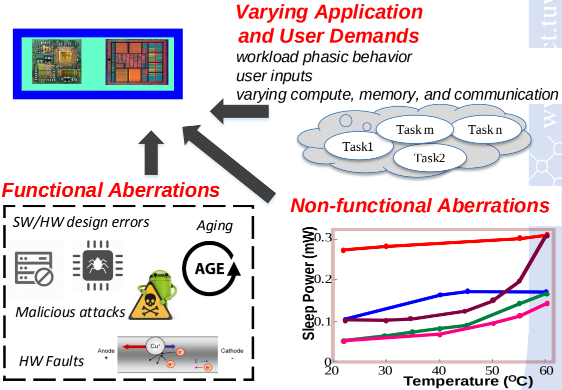
Malicious attacks

HW Faults

Non-functional Aberrations

The Problem

- Large number of resources
- Many tight constraints
- Varying application demands, both within and between applications;



Varying Application and User Demands

workload phasic behavior
user inputs
varying compute, memory, and communication

Functional Aberrations

SW/HW design errors

Aging

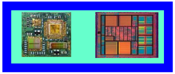
Malicious attacks

HW Faults

Non-functional Aberrations

The Problem

- Large number of resources
- Many tight constraints
- Varying application demands, both within and between applications;
- Functional Aberrations:
  - Design errors or omissions;
  - Malicious attacks;
  - Aging;
  - Soft errors;
- Non-functional Aberrations:
  - Performance;
  - Power consumption;

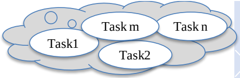


Varying Application and User Demands

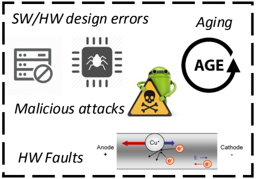
workload phasic behavior

user inputs

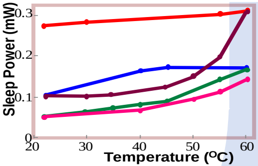
varying compute, memory, and communication



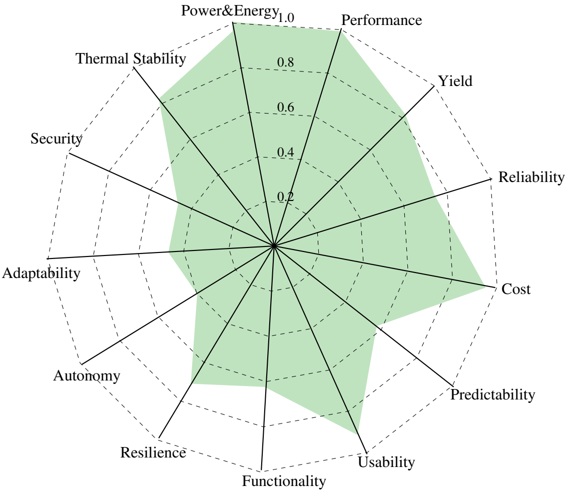
Functional Aberrations



Non-functional Aberrations

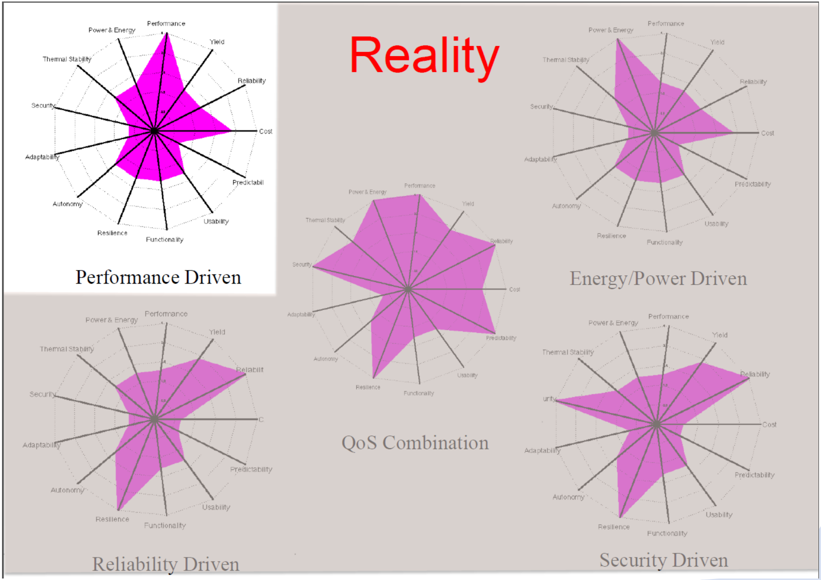


The SoC Radar

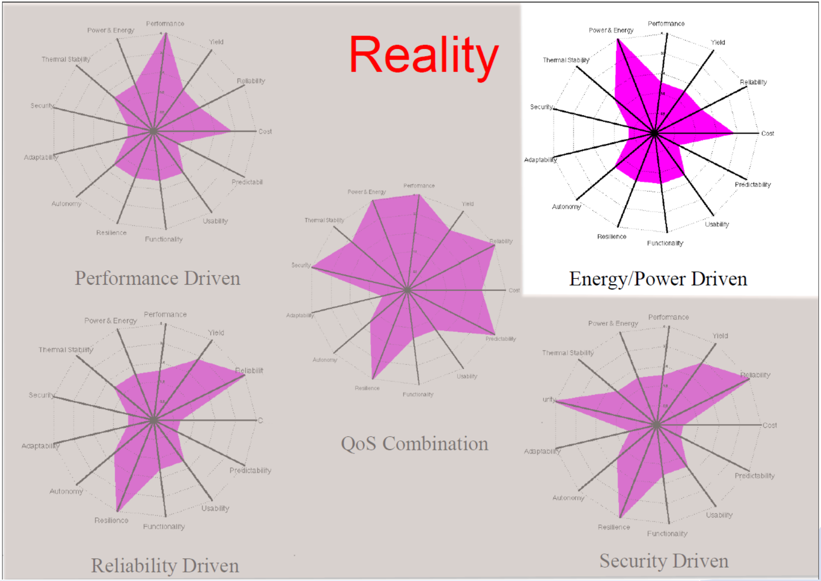


Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. “On-Chip Self-Awareness Using Cyberphysical-Systems-On-Chip (CPSoC)”. . In: *Proceedings of the 12th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. New Delhi, India, Oct. 2014

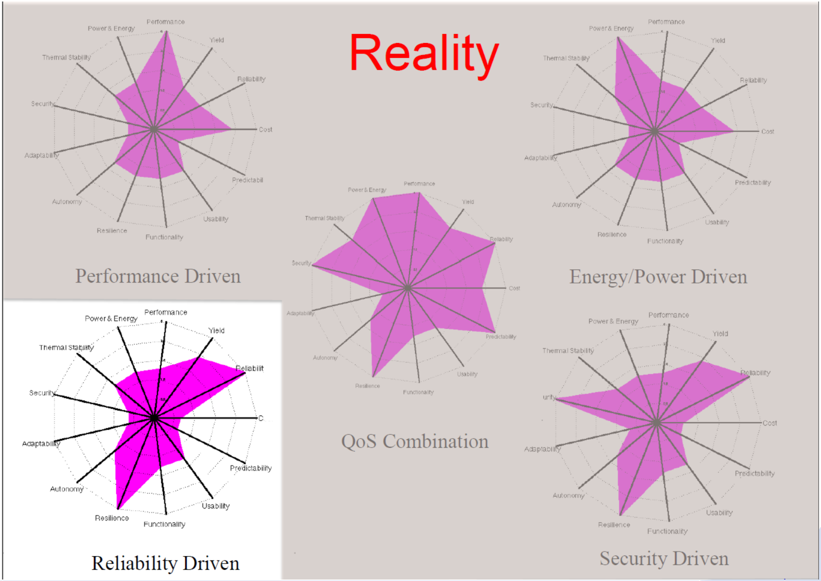
The SoC Radar



The SoC Radar



The SoC Radar



Reality

Performance Driven

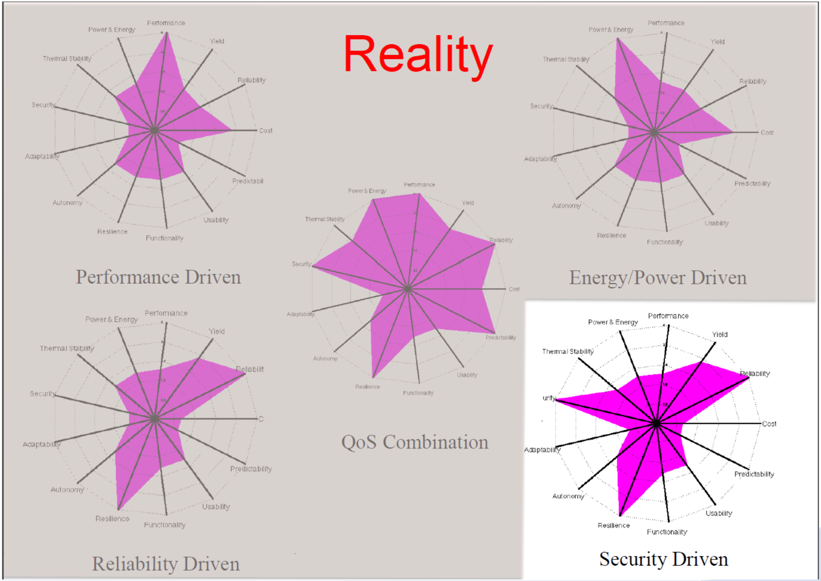
Energy/Power Driven

QoS Combination

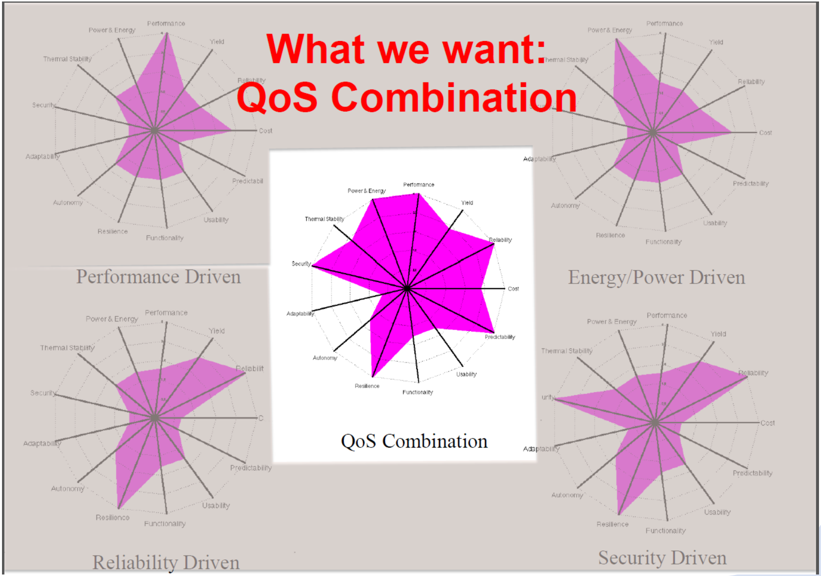
Reliability Driven

Security Driven

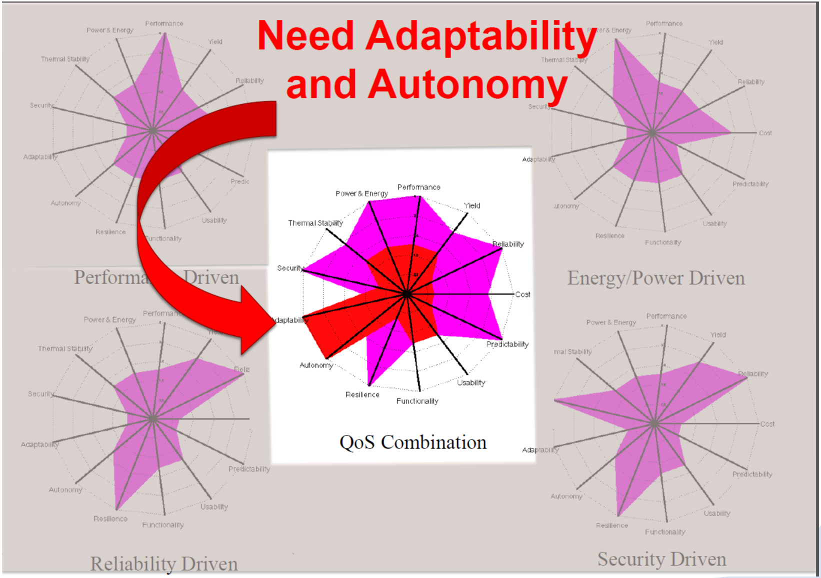
The SoC Radar



The SoC Radar



The SoC Radar



Need Adaptability and Autonomy

Performance Driven

Energy/Power Driven

QoS Combination

Reliability Driven

Security Driven

Self-Awareness

Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

Which Ingredients Lead to Awareness ?

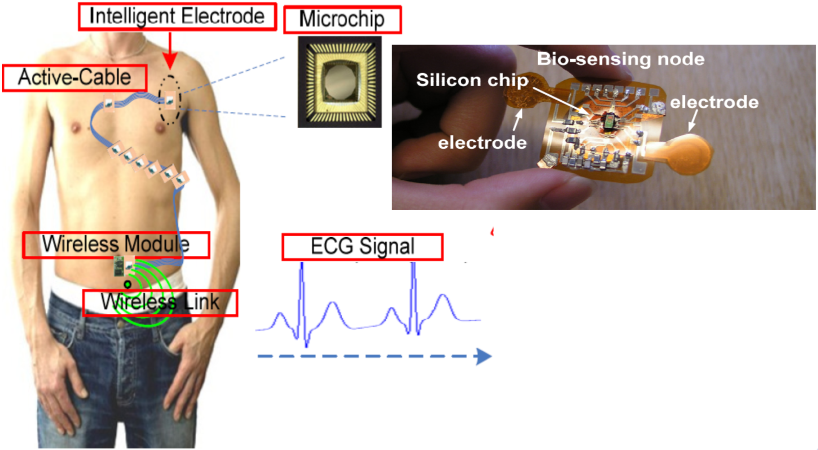
Johan Moreelses "Der Alchemist", 1630

Which Ingredients Lead to Awareness ?

- Data abstraction
- Disambiguation
- Desirability scale
- History
- Goals
- Attention
- Learning
- Introspection

Johan Moreelses “Der Alchemist”, 1630

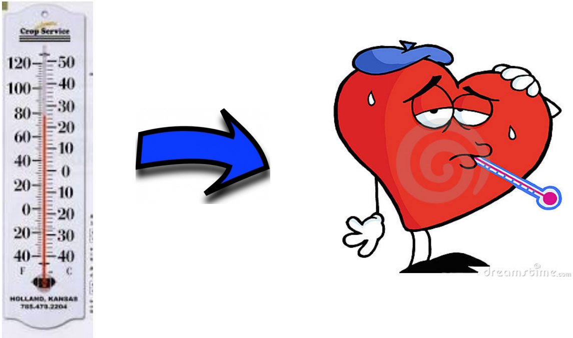
Awareness for Resource Constrained, Insect-like Gadgets



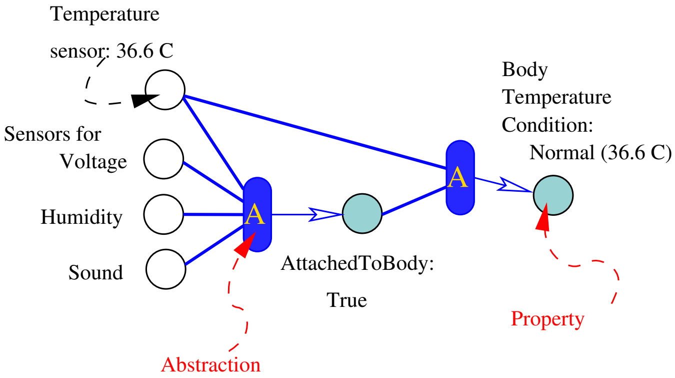
Courtesy of KTH

Abstractions and Models

Abstraction: Mapping of Measurements ⇒ Properties



Abstractions and Models



Temperature sensor: 36.6 C

Sensors for Voltage

Humidity

Sound

AttachedToBody: True

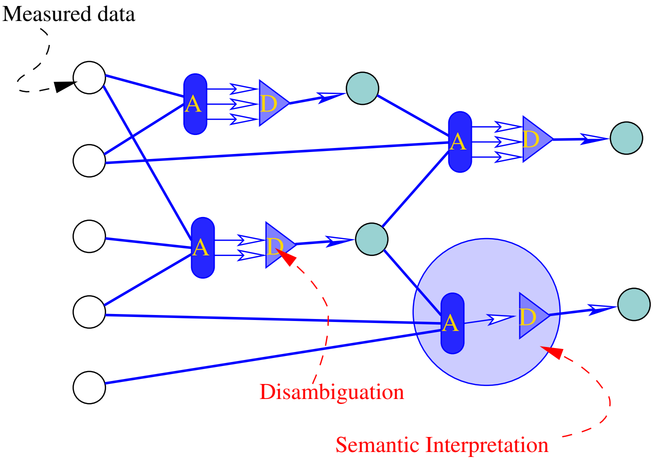
Body Temperature Condition: Normal (36.6 C)

Abstraction

Property

Disambiguation

Selection among several interpretations



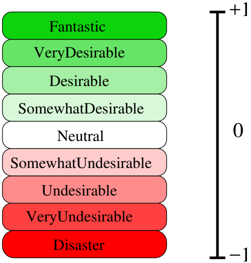
Measured data

Disambiguation

Semantic Interpretation

Desirability Scale

Desirability is the common, internal currency.

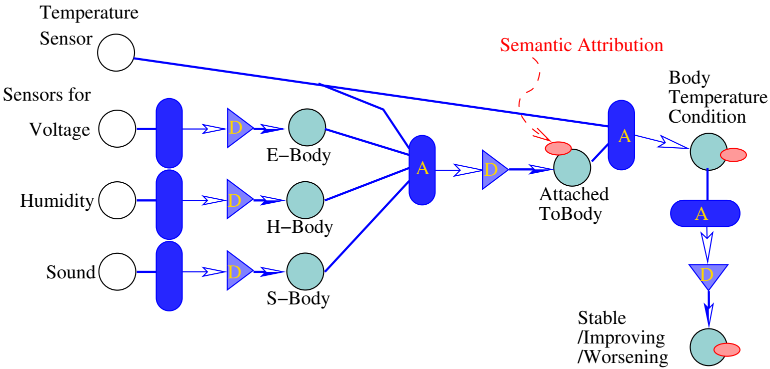


Fantastic
VeryDesirable
Desirable
SomewhatDesirable
Neutral
SomewhatUndesirable
Undesirable
VeryUndesirable
Disaster

+1
0
−1

Semantic Attribution maps the values of a property to a point in the desirability scale.

BioPatch with Semantic Attribution



History

History of a Property The evolution of the values of a property.

Abstracted History The history stores abstracted values.

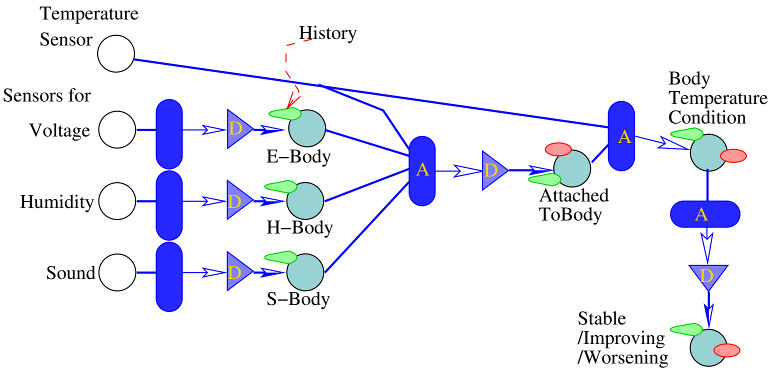
Attributed History The history is annotated with attributions.

Fading History If the property values are more abstracted the longer ago they have occurred.

Consolidating History Relevant memories are enforced, irrelevant memories are cleaned.

Evolving History Memories are adjusted to fit later observations.

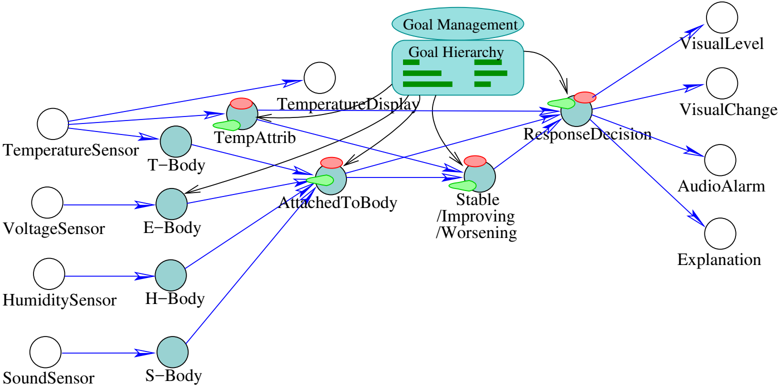
BioPatch with History



Expectations and Goals

- Expectations on Environment
- Expectations on Subject
- Sub-Goals
- Goals
- Purpose

Acting BioPatch

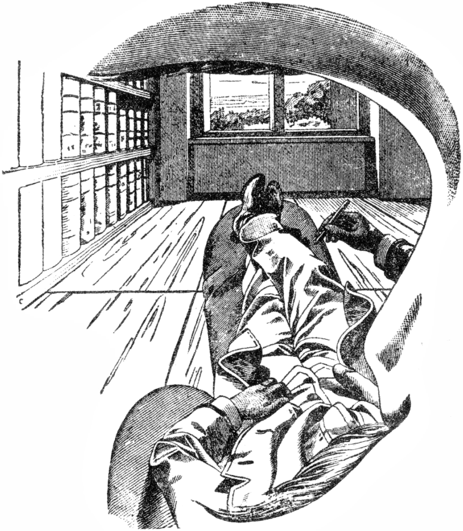


Introspection and Simulation

Self Inspection Engine

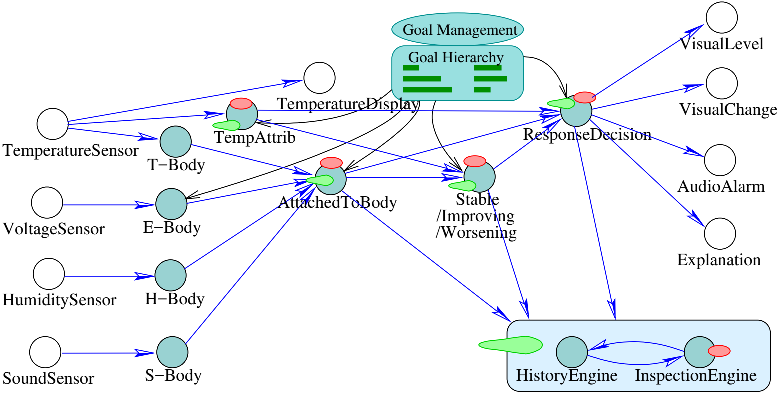
Model Transformation

Simulation

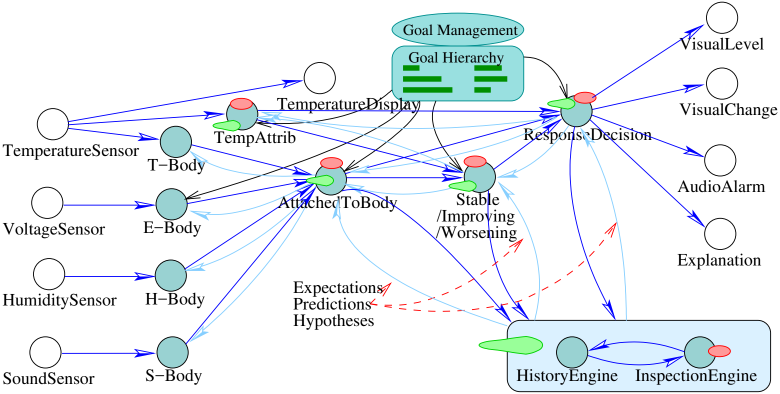


Ernst Mach “Innenperspektive”, 1886

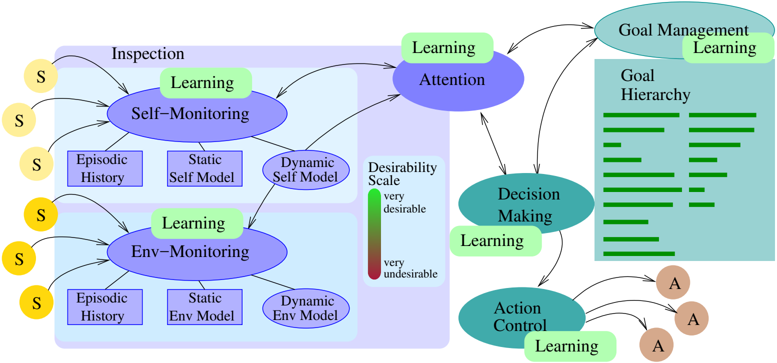
Self-inspecting BioPatch



BioPatch with Top-down Prediction

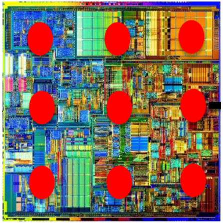


Self-Awareness Architecture



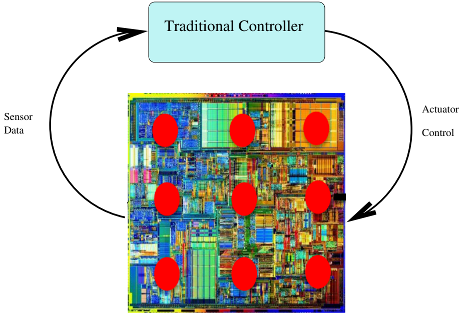
Cyber-Physical SoC

Cyber-Physical SoC



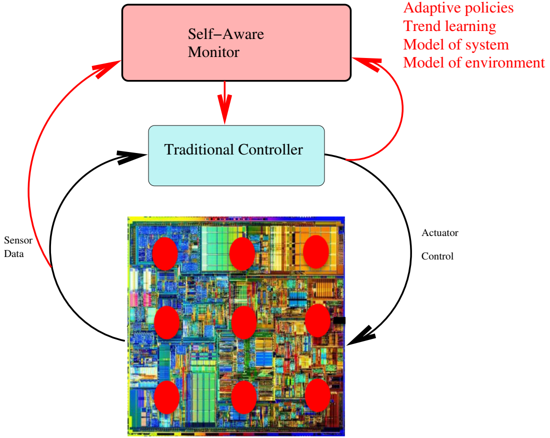
Self-monitoring chip

Cyber-Physical SoC



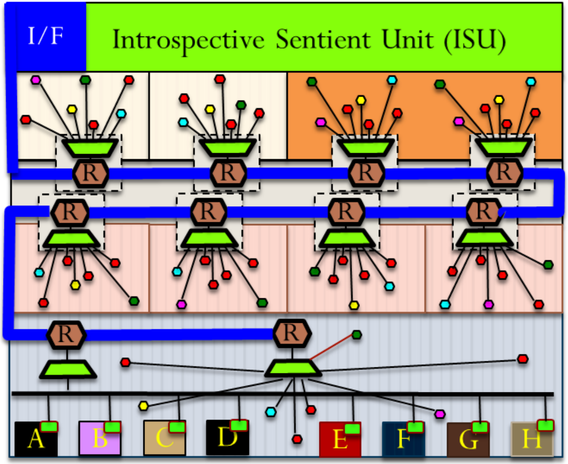
Self-monitoring chip

Cyber-Physical SoC



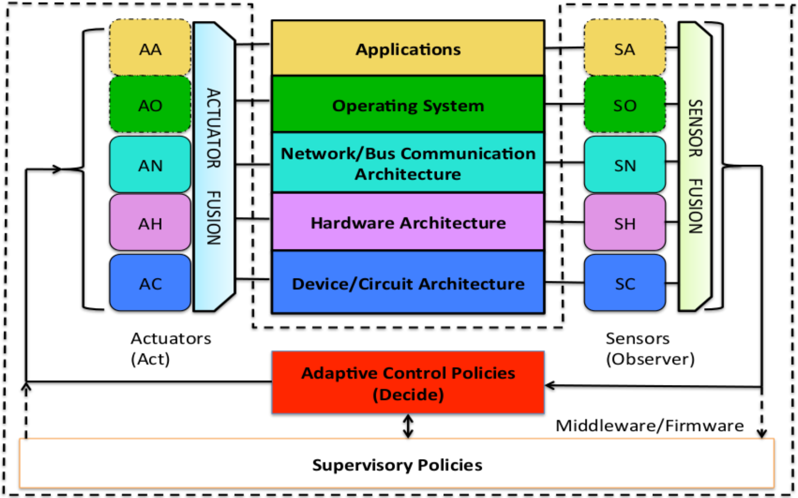
Self-monitoring chip

CPSoC - A Sensor Rich SoC Platform



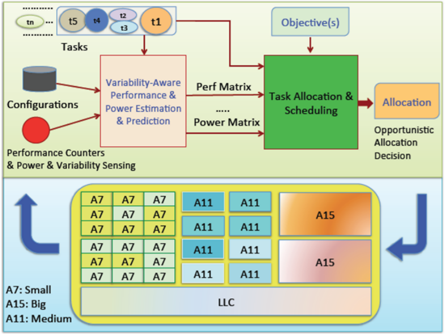
Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. “CyberPhysical-System-On-Chip (CPSoC): A Self-Aware MPSoC Paradigm with Cross-Layer Virtual Sensing and Actuation”. In: *Proceedngs of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*. Grenoble, France, Mar. 2015

CPSoC - A Sensor Rich SoC Platform

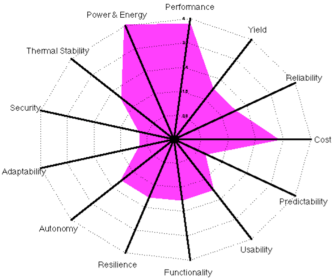


Nikil Dutt, Axel Jantsch, and Santanu Sarma. “Self-Aware Cyber-Physical Systems-on-Chip”. In: *Proceedings of the International Conference for Computer Aided Design*. invited. Austin, Texas, USA, Nov. 2015

Improvement of Energy Efficiency



tn
t5
t4
t2
t3
t1
Tasks
Objective(s)
Configurations
Variability-Aware Performance & Power Estimation & Prediction
Perf Matrix
.....
Power Matrix
Task Allocation & Scheduling
Allocation
Opportunistic Allocation Decision
Performance Counters & Power & Variability Sensing
A7
A11
A15
LLC
A7: Small
A15: Big
A11: Medium

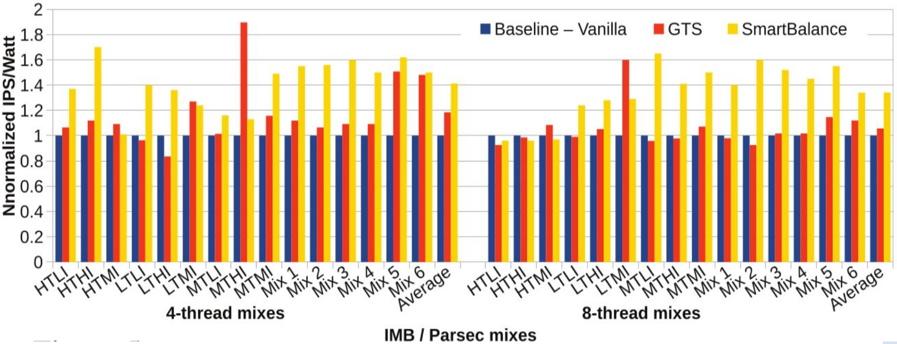


Goal:

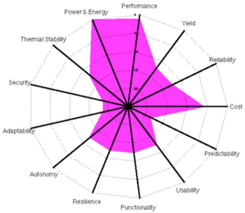
- **Energy Efficiency**

Santanu Sarma and Nikil Dutt. “Cross-Layer Exploration of Heterogeneous Multicore Processor Configurations”. In: *VLSI Design Conference*. 2015

Improvement of Energy Efficiency

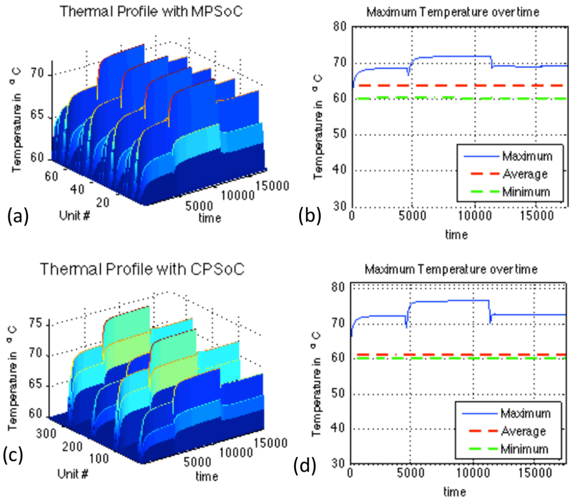


The benefit comes from actually measuring energy efficiency.



Santanu Sarma, T. Muck, L. A.D. Bathen, N. Dutt, and A. Nicolau. "SmartBalance: A Sensing-Driven Linux Load Balancer for Energy Efficiency of Heterogeneous MPSoCs". In: *Proceedings of the Design Automation Conference*. July 2015

Thermal-Aware Performance



Throughput improvement by 70%-300% for same power and temperature.

Benefit is due to accurate and fine-grain measurement and tight tracking.

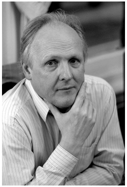
Santanu Sarma, Nikil Dutt, N. Venkatasubramaniana, A. Nicolau, and P. Gupta. *CyberPhysical-System-On-Chip (CPSoC): Sensor-Actuator Rich Self-Aware Computational Platform*. Tech. rep. CECS Technical Report No: CECS TR–13–06. Irvine, CA 92697-2620, USA: Center for Embedded Computer Systems University of California, Irvine, May 2013.

Hardware Faults

Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

HW Faults



Steve Furber in a keynote at ETS in 2006 predicted that

> within a decade we will see 100 billion transistor chips. That is the good news. The bad news is that 20 billion of those transistors will fail in manufacture and a further 10 billion will fail in the first year of operation.

Steve Furber. “Living with failure: Lessons from nature?” In: *Proceedings of the European Test Symposium (ETS)*. 2006

Outline

1 Motivation

2 Concepts of Self-Awareness

3 Hardware Faults
- Fault Types
- On-line Diagnosis
- Reliable NoC Design
- Health Management

4 Self-Aware Monitoring
- Context Aware Health Monitoring
- Early Warning Score

5 Conclusions

What can fail?

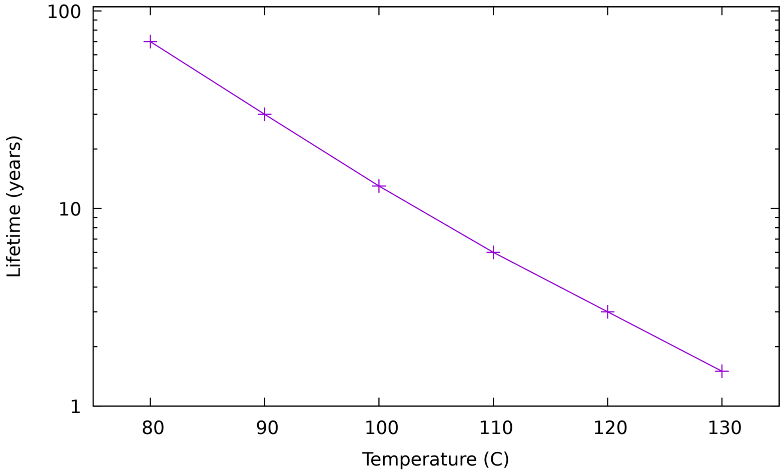
- Sensors
- Computing components
- Communication links
- Actuators
- SoC components

Fault causes and effects

Physical cause		Accelerated by process variation	Fault class	Burstiness
Radiation	Neutrons	No	Transient logic	Yes
	α-particles	No	Transient logic	Yes
Electromagnetic interference	cross coupling of parallel wires	No	Intermittent delay	No
	self coupling, Skin effect leading to higher resistance	No	Intermittent delay	No
Aging	Electromigration	Yes	intermittent → permanent delay and logic fault	No
	Bias Temperature Instability (BTI)	Yes	intermittent → permanent delay fault	No
	Hot carrier injection	Yes	intermittent → permanent delay faults	No
	Oxide breakdown/Time Dependent Dielectric Breakdown (TDDB)	Yes	intermittent → permanent logic fault	No
Power density variation and heat flux	leakage power variation due to temperature differences	Yes	Intermittent → permanent delay and logic faults	Yes
	performance variation due to temperature differences;	Yes	intermittent delay failures	Yes
	variations in wear-out effects due to temperature differences	Yes	intermittent and permanent, delays, opens and shorts	Maybe

Martin Radetzki, Chaochao Feng, Xueqian Zhao, and Axel Jantsch. “Methods for Fault Tolerance in Networks-on-Chip”. In: *ACM Computing Surveys* 46.1 (July 2013), 8:1–8:38

Aging is temperature dependent

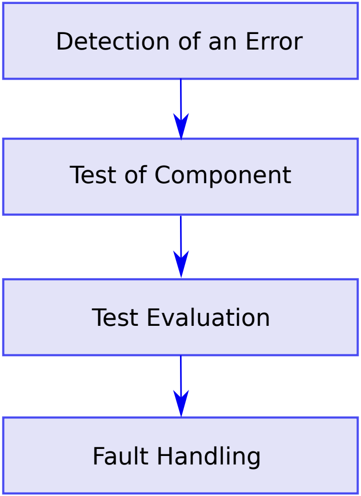


Lifetime of an inverter chain decreases by a factor of 2.2 for every 10°C increase in operating temperature due to NBTI.

Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

Flow of Diagnosis



Detection of an Error

Error detection codes
Periodic self-test
Modular redundancy

Test of Component

BIST
Initiated by system

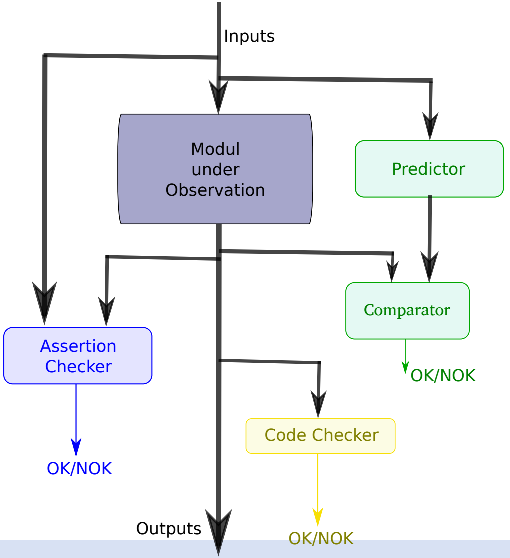
Test Evaluation

Analysis of response
Localization of fault

Fault Handling

Reconfiguration
Update of health map

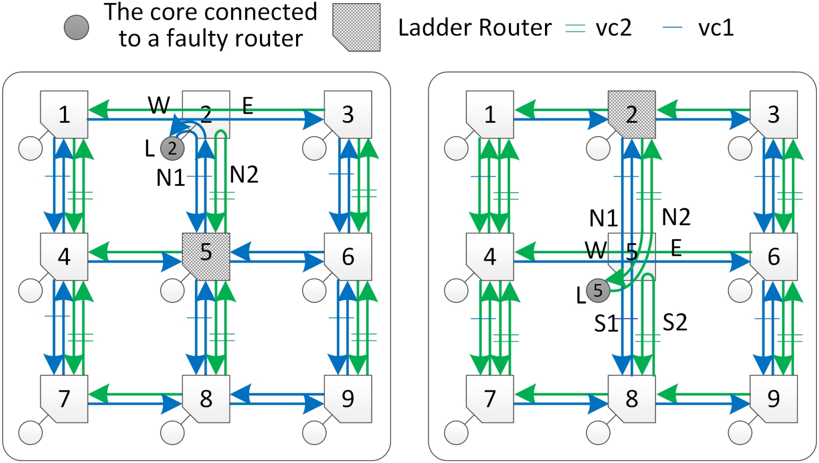
Monitoring during Operation



Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

Minimal Intrusion Testing

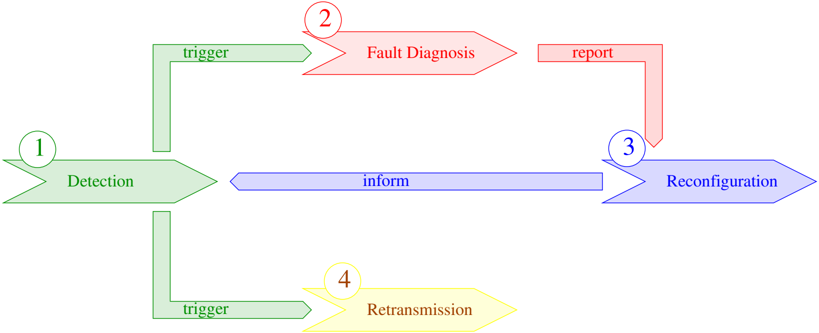


(a)

(b)

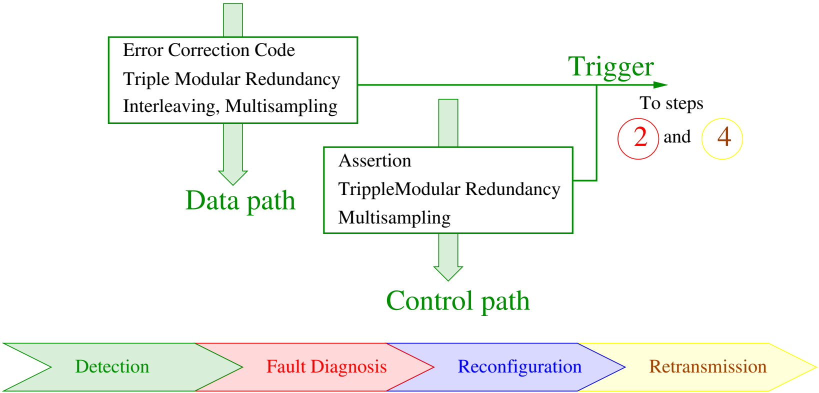
Junshi Wang, Masoumeh Ebrahimi, Letian Huang, Qiang Li, Guangjun Li, and Axel Jantsch. "Minimizing the System Impact of Router Faults by Means of Reconfiguration and Adaptive Routing". In: *Microprocessors and Microsystems* 51 (June 2017), pp. 252–263

NoC Fault Reliability Flow



Junshi Wang, Masoumeh Ebrahimi, Letian Huang, Axel Jantsch, and Guangjun Li. “Design of Fault-Tolerant and Reliable Networks-on-Chip”. In: *IEEE Annual Symposium on VLSI (ISVLSI)*. Montpelllier, France, July 2015

Fault Detection



Error Correction Code

Triple Modular Redundancy

Interleaving, Multisampling

Data path

Assertion

TrippleModular Redundancy

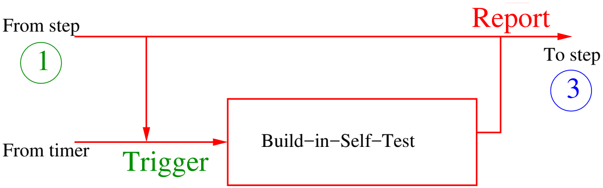
Multisampling

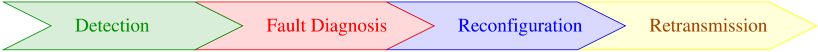
Control path

Trigger

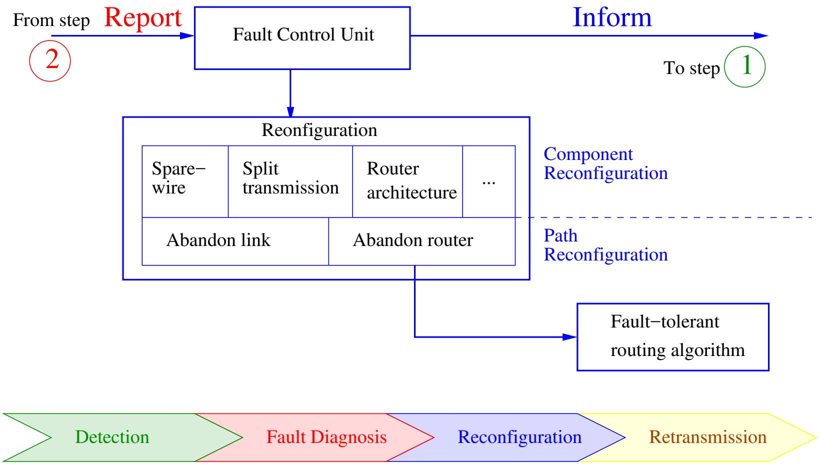
To steps 2 and 4

Fault Diagnosis

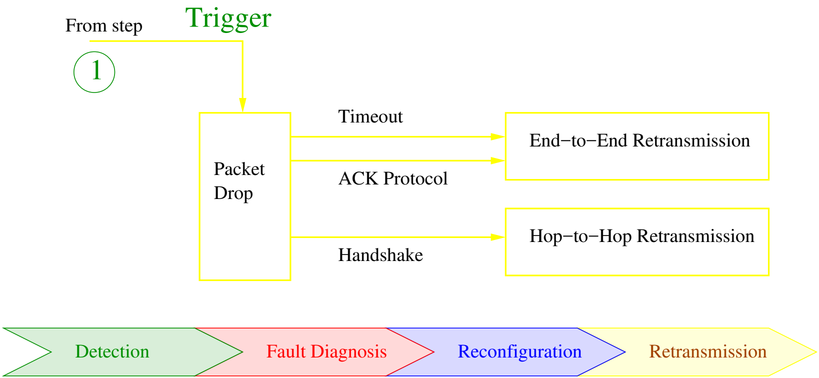




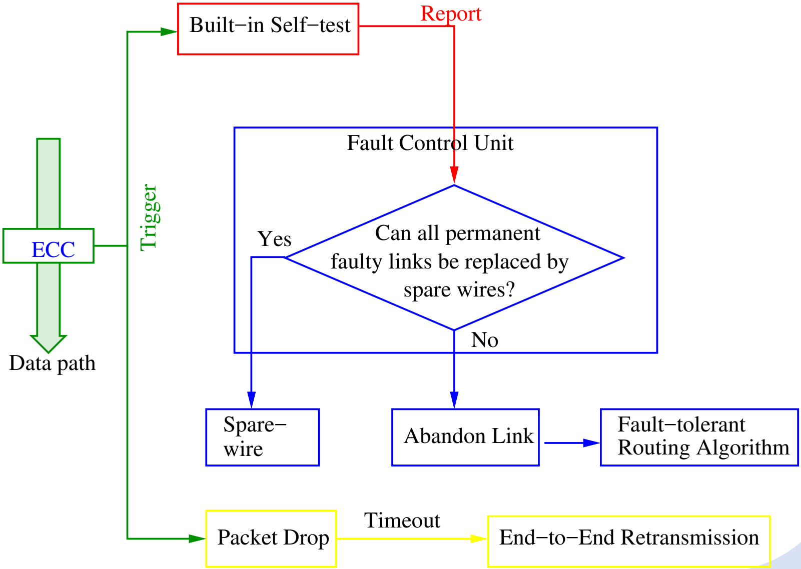
Reconfiguration



Retransmission



Example: Reliability Design for Faulty Links

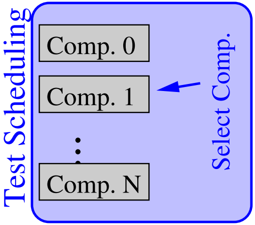

Built−in Self−test
Report
Fault Control Unit
Can all permanent faulty links be replaced by spare wires?
Yes
No
ECC
Trigger
Data path
Spare−wire
Abandon Link
Fault−tolerant Routing Algorithm
Packet Drop
Timeout
End−to−End Retransmission

Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

CASP

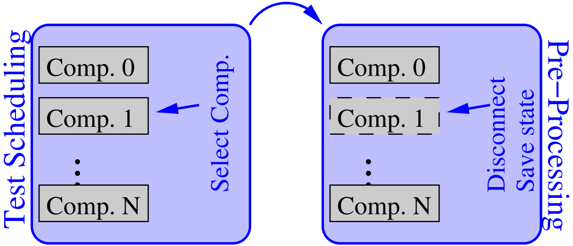
Concurrent Autonomous Chip Self-Testing using Stored Test Patterns



Y. Li, S. Makar, and S. Mitra. “CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns”. In: *2008 Design, Automation and Test in Europe*. Mar. 2008, pp. 885–890

CASP

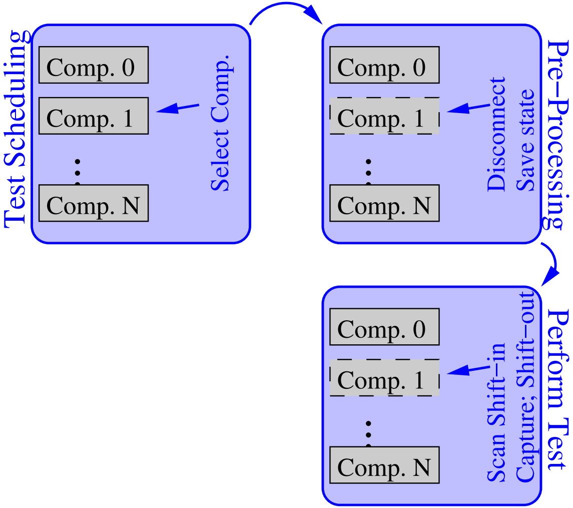
Concurrent Autonomous Chip Self-Testing using Stored Test Patterns



Y. Li, S. Makar, and S. Mitra. "CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns". In: *2008 Design, Automation and Test in Europe*. Mar. 2008, pp. 885–890

CASP

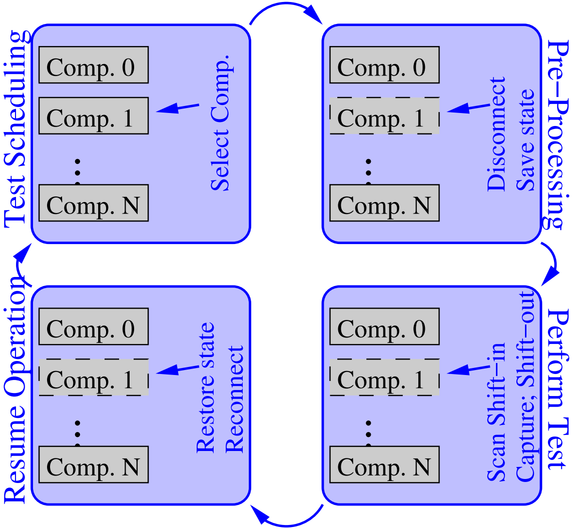
Concurrent Autonomous Chip Self-Testing using Stored Test Patterns



Y. Li, S. Makar, and S. Mitra. "CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns". In: *2008 Design, Automation and Test in Europe*. Mar. 2008, pp. 885–890

CASP

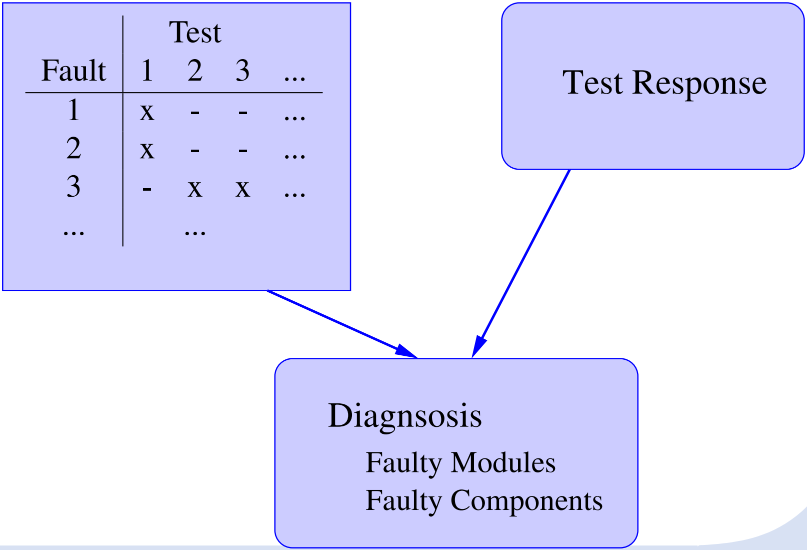
Concurrent Autonomous Chip Self-Testing using Stored Test Patterns



Y. Li, S. Makar, and S. Mitra. "CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns". In: *2008 Design, Automation and Test in Europe*. Mar. 2008, pp. 885–890

Diagnosis

Fault Dictionary



	Test			
Fault	1	2	3	...
1	x	-	-	...
2	x	-	-	...
3	-	x	x	...
...		...		

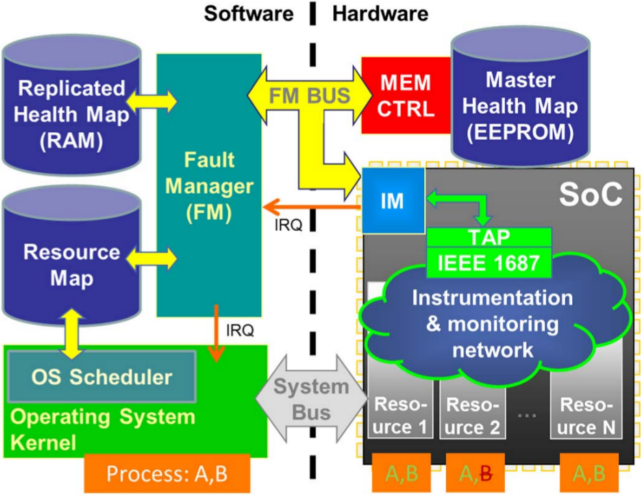
Test Response

Diagnsosis

Faulty Modules

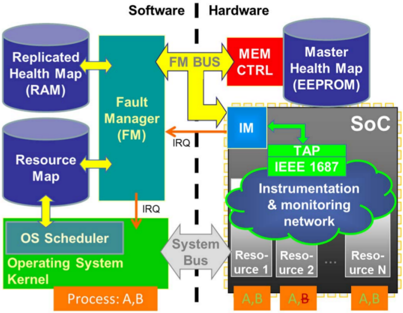
Faulty Components

Health Management



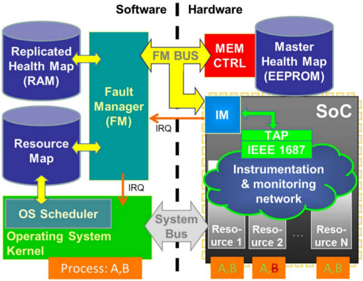
A. Jutman, K. Shibin, and S. Devadze. “Reliable health monitoring and fault management infrastructure based on embedded instrumentation and IEEE 1687”. In: *2016 IEEE AUTOTESTCON*. Sept. 2016, pp. 1–10

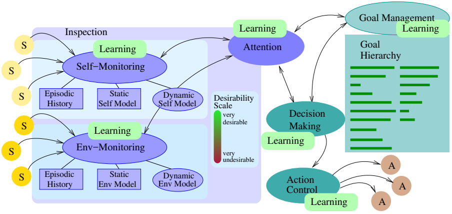
Health Management



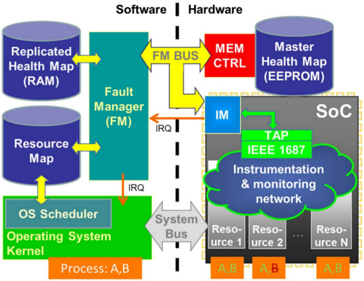
- Health Map: detailed information about faults
- Resource Map: List of healthy resources
- Fault Manager: Updates HM and RM
- Instrument Manager: interface to the instrumentation and monitoring network
- Fault detection and diagnosis: Embedded monitors and instrumentation

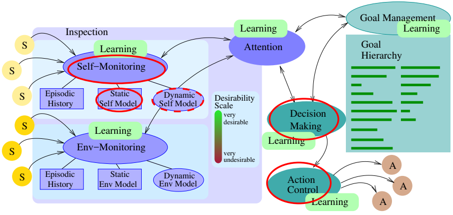
Self-Awareness for Health Management



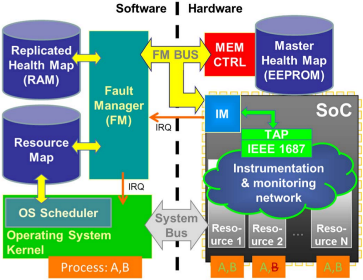


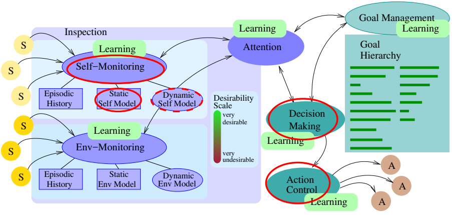
Self-Awareness for Health Management





Self-Awareness for Health Management





What is missing for Self-Awareness?

- History
- Learning
- Attention
- Goal management
- Comprehensive assessment

Self-Aware Monitoring

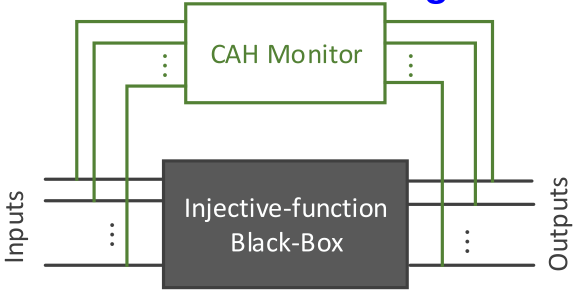
Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

Context Aware Health Monitoring of an AC Motor

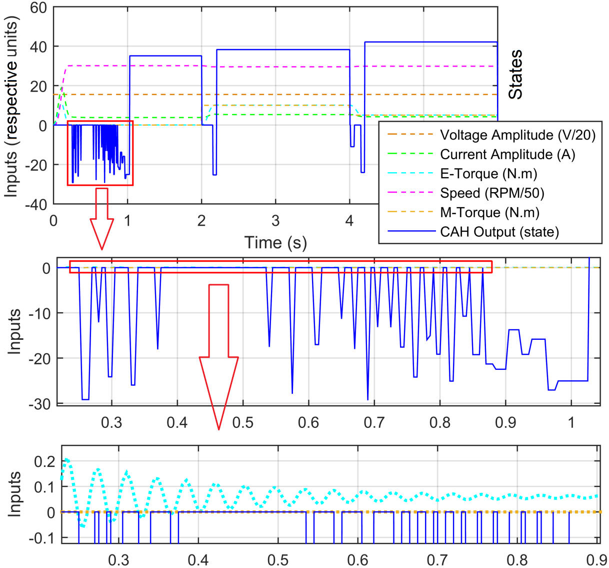


CAH Features

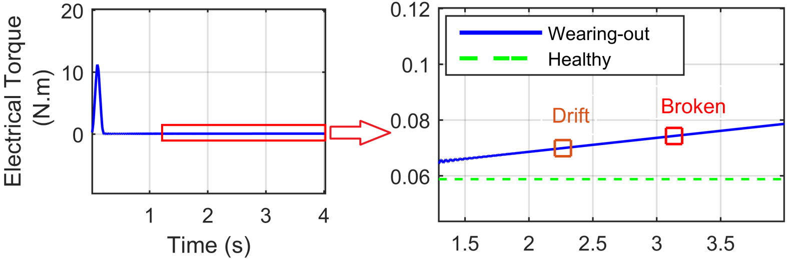
- No Model and minimal assumptions about the system
- Main assumption: injective function
- States are automatically inferred and learned
- Anomalies are detected when injectivity is violated

M. Götzinger, N. TaheriNejad, H. A. Kholerdi, and A. Jantsch. "On the design of context-aware health monitoring without a priori knowledge; an AC-Motor case-study". In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Apr. 2017, pp. 1–5

CAH Normal Mode



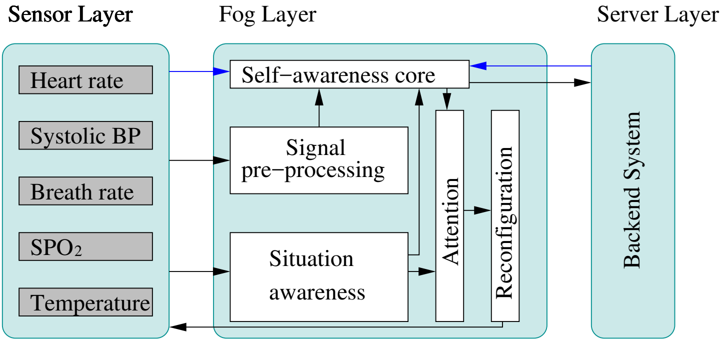
CAH Anomaly Detection



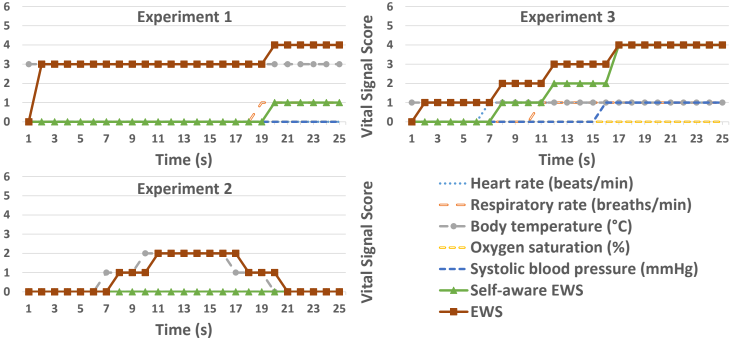
Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

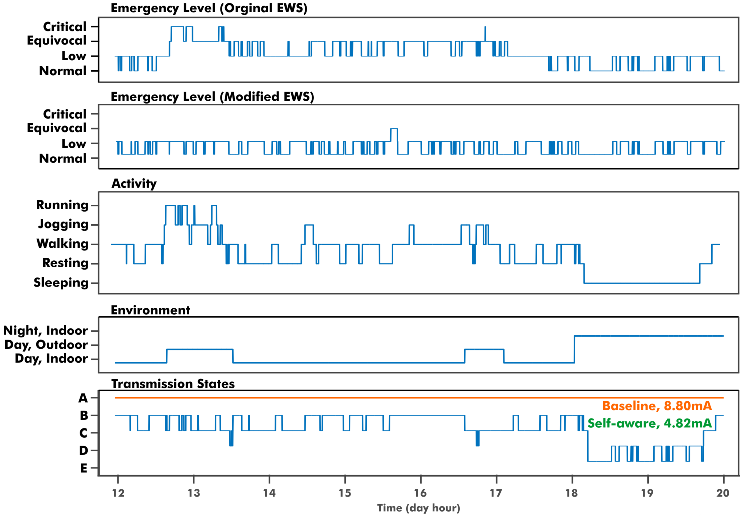
Early Warning Score



Assessing Data Reliability



Situation Awareness

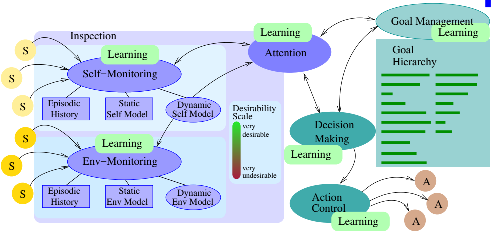


Conclusions

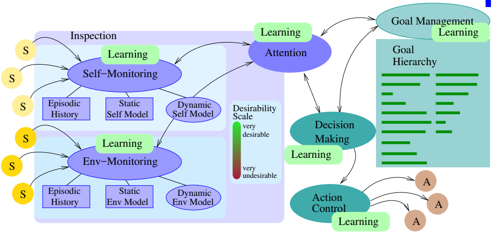
Outline

1. Motivation
2. Concepts of Self-Awareness
3. Hardware Faults
   - Fault Types
   - On-line Diagnosis
   - Reliable NoC Design
   - Health Management
4. Self-Aware Monitoring
   - Context Aware Health Monitoring
   - Early Warning Score
5. Conclusions

How can Self-Awareness Help?

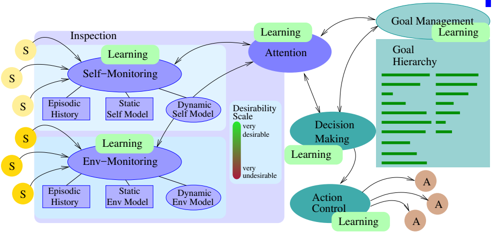


How can Self-Awareness Help?



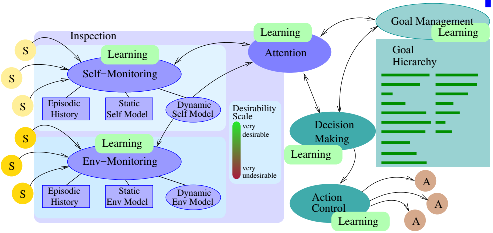
- Comprehensive self- and environment monitoring;

How can Self-Awareness Help?



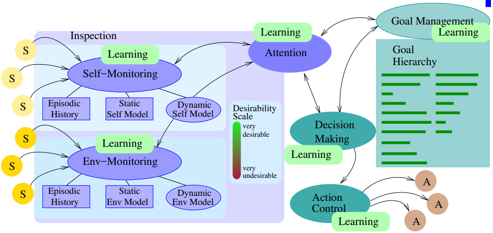
- Comprehensive self- and environment monitoring;
- Formalisms to express all desired and forbidden behavior;

How can Self-Awareness Help?



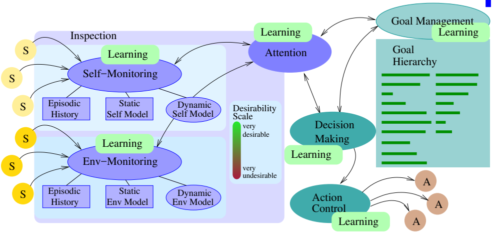
- Comprehensive self- and environment monitoring;
- Formalisms to express all desired and forbidden behavior;
- Track behavior and changes over time;

How can Self-Awareness Help?



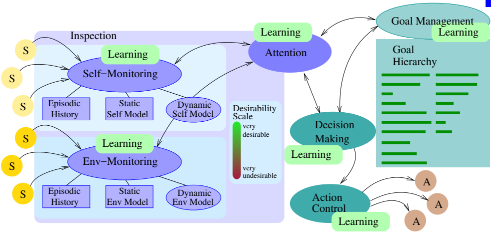
- Comprehensive self- and environment monitoring;
- Formalisms to express all desired and forbidden behavior;
- Track behavior and changes over time;
- Levels of monitoring details (attention directed monitoring);

How can Self-Awareness Help?



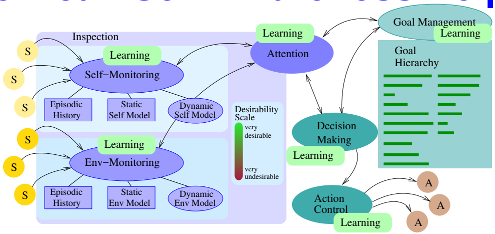
- Comprehensive self- and environment monitoring;
- Formalisms to express all desired and forbidden behavior;
- Track behavior and changes over time;
- Levels of monitoring details (attention directed monitoring);
- Relating anomalies in different domains (desirability scale);

How can Self-Awareness Help?



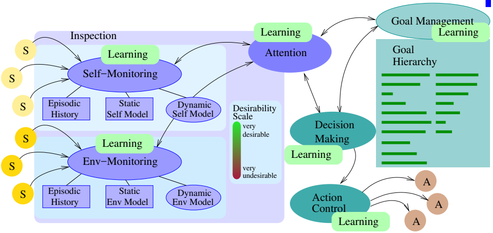
- Comprehensive self- and environment monitoring;
- Formalisms to express all desired and forbidden behavior;
- Track behavior and changes over time;
- Levels of monitoring details (attention directed monitoring);
- Relating anomalies in different domains (desirability scale);
- Comprehensive self- and environment assessment over time;

How can Self-Awareness Help?



- Comprehensive self- and environment monitoring;
- Formalisms to express all desired and forbidden behavior;
- Track behavior and changes over time;
- Levels of monitoring details (attention directed monitoring);
- Relating anomalies in different domains (desirability scale);
- Comprehensive self- and environment assessment over time;
- Dynamic goal management;

How can Self-Awareness Help?



- Comprehensive self- and environment monitoring;
- Formalisms to express all desired and forbidden behavior;
- Track behavior and changes over time;
- Levels of monitoring details (attention directed monitoring);
- Relating anomalies in different domains (desirability scale);
- Comprehensive self- and environment assessment over time;
- Dynamic goal management;
- Learning for continuous adaptation to specific environments, applications, and threats.

Challenges with Self-aware, Adaptive Systems

- How to assess and ensure the quality of sensor data?

Challenges with Self-aware, Adaptive Systems

- How to assess and ensure the quality of sensor data?
- How to express “correctness”?

Challenges with Self-aware, Adaptive Systems

- How to assess and ensure the quality of sensor data?
- How to express “correctness”?
- How to validate a smartly adapting system?

Challenges with Self-aware, Adaptive Systems

- How to assess and ensure the quality of sensor data?
- How to express “correctness”?
- How to validate a smartly adapting system?
- How to perform tradeoff analysis for smartness features?

Challenges with Self-aware, Adaptive Systems

- How to assess and ensure the quality of sensor data?
- How to express “correctness”?
- How to validate a smartly adapting system?
- How to perform tradeoff analysis for smartness features?
- How to quantify uncertainty, dynamicity, and variability in the platform, the environment, and the applications?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to reconcile autonomy with safety critical and real-time systems?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to reconcile autonomy with safety critical and real-time systems?

- How to develop efficient learning algorithms?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to reconcile autonomy with safety critical and real-time systems?
- How to develop efficient learning algorithms?
- How to formally model and formulate the goal management?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to reconcile autonomy with safety critical and real-time systems?
- How to develop efficient learning algorithms?
- How to formally model and formulate the goal management?
- How to verify it w.r.t. convergence, efficiency, robustness, QoS guarantees, etc.?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to reconcile autonomy with safety critical and real-time systems?
- How to develop efficient learning algorithms?
- How to formally model and formulate the goal management?
- How to verify it w.r.t. convergence, efficiency, robustness, QoS guarantees, etc.?
- How to handle a dynamic hierarchy of goals?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to make goal management lightweight?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to make goal management lightweight?
- How to scale self-awareness?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to make goal management lightweight?
- How to scale self-awareness?
- How to detect and handle design errors?

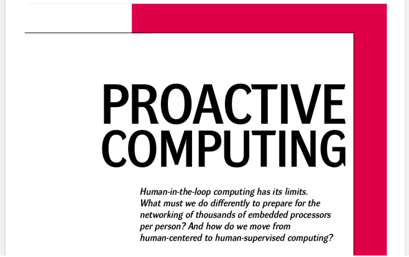
Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to make goal management lightweight?
- How to scale self-awareness?
- How to detect and handle design errors?
- How to detect and handle malicious attacks?

Challenges with Self-aware, Autonomous, Adaptive SoCs

- How to make goal management lightweight?
- How to scale self-awareness?
- How to detect and handle design errors?
- How to detect and handle malicious attacks?
- .....

Let’s Get Out



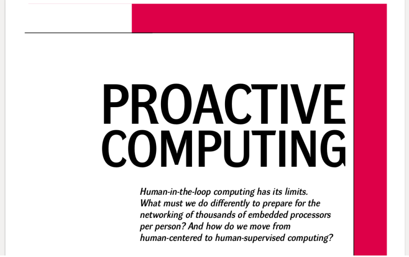
PROACTIVE COMPUTING

Human-in-the-loop computing has its limits. What must we do differently to prepare for the networking of thousands of embedded processors per person? And how do we move from human-centered to human-supervised computing?

David Tennenhouse. “Proactive Computing”. In: *Communications of the ACM* 43.5 (May 2000), pp. 43–50

Let’s Get Out

- Let’s get physical



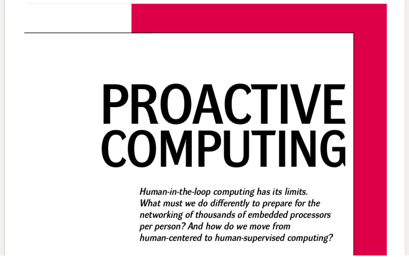
PROACTIVE COMPUTING

Human-in-the-loop computing has its limits. What must we do differently to prepare for the networking of thousands of embedded processors per person? And how do we move from human-centered to human-supervised computing?

David Tennenhouse. “Proactive Computing”. In: *Communications of the ACM* 43.5 (May 2000), pp. 43–50

Let’s Get Out

- Let’s get physical
- Let’s get real



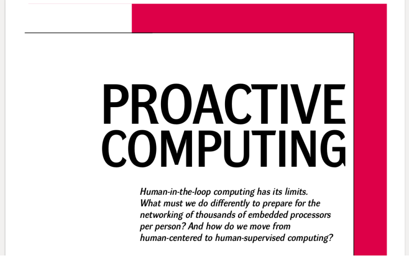
PROACTIVE COMPUTING

Human-in-the-loop computing has its limits. What must we do differently to prepare for the networking of thousands of embedded processors per person? And how do we move from human-centered to human-supervised computing?

David Tennenhouse. “Proactive Computing”. In: *Communications of the ACM* 43.5 (May 2000), pp. 43–50

Let’s Get Out

- Let’s get physical
- Let’s get real
- Let’s get out

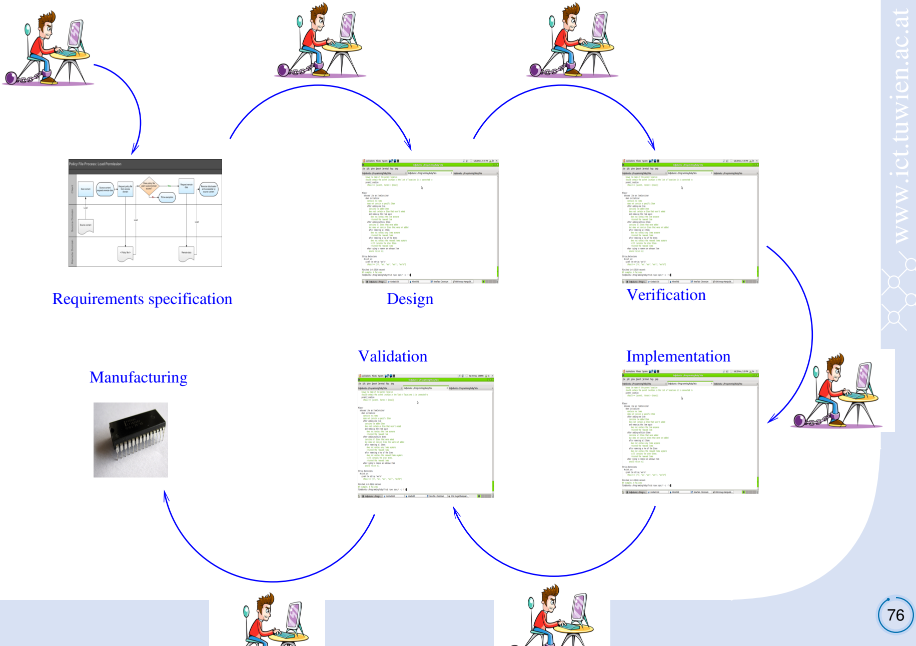


PROACTIVE COMPUTING

Human-in-the-loop computing has its limits. What must we do differently to prepare for the networking of thousands of embedded processors per person? And how do we move from human-centered to human-supervised computing?

David Tennenhouse. “Proactive Computing”. In: *Communications of the ACM* 43.5 (May 2000), pp. 43–50

Traditional Design Flow



Requirements specification

Design

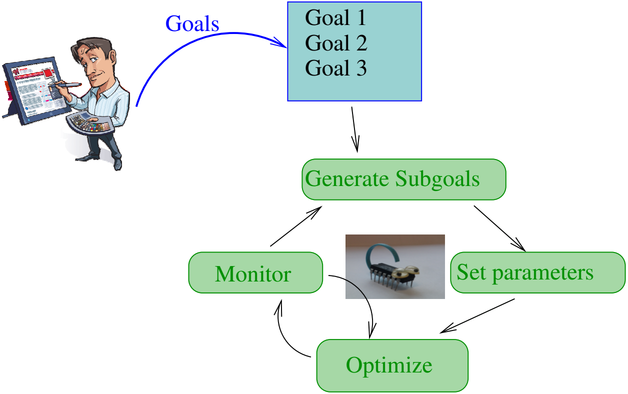
Verification

Implementation

Validation

Manufacturing

Design of Self-Aware Chips



Goals

Goal 1
Goal 2
Goal 3

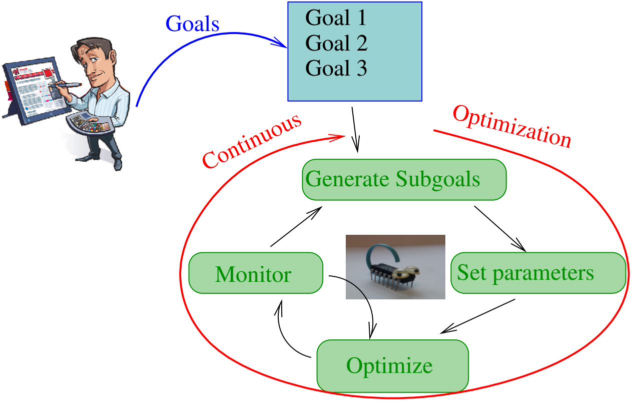
Generate Subgoals

Set parameters

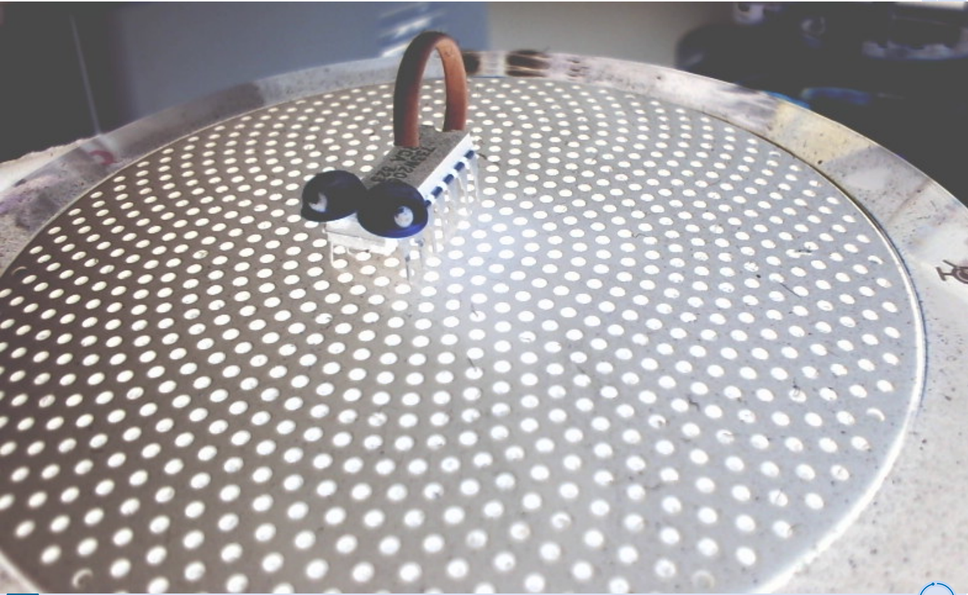
Optimize

Monitor

Design of Self-Aware Chips



Questions ?



References I

Nikil Dutt, Axel Jantsch, and Santanu Sarma. “Self-Aware Cyber-Physical Systems-on-Chip”. In: *Proceedings of the International Conference for Computer Aided Design*. invited. Austin, Texas, USA, Nov. 2015.

Nikil Dutt, Axel Jantsch, and Santanu Sarma. “Towards Smart Embedded Systems: A Self-Aware System-on-Chip Perspective”. In: *ACM Transactions on Embedded Computing Systems, Special Issue on Innovative Design Methods for Smart Embedded Systems* 15.2 (Feb. 2016). invited, pp. 22–27.

Steve Furber. “Living with failure: Lessons from nature?” In: *Proceedings of the European Test Symposium (ETS)*. 2006.

M. Götzinger, N. TaheriNejad, H. A. Kholerdi, and A. Jantsch. “On the design of context-aware health monitoring without a priori knowledge; an AC-Motor case-study”. In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Apr. 2017, pp. 1–5.

A. Jutman, K. Shibin, and S. Devadze. “Reliable health monitoring and fault management infrastructure based on embedded instrumentation and IEEE 1687”. In: *2016 IEEE AUTOTESTCON*. Sept. 2016, pp. 1–10.

References II

Axel Jantsch and Kalle Tammemäe. “A Framework of Awareness for Artificial Subjects”. In: *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis*. CODES ’14. New Delhi, India: ACM, 2014, 20:1–20:3.

S. Kounev, J.O. Kephart, A. Milenkoski, and X. Zhu, eds. *Self-Aware Computing Systems*. Springer, 2017.

Peter R. Lewis et al. “Architectural Aspects of Self-aware and Self-expressive Computing Systems”. In: *IEEE Computer* (Aug. 2015).

Peter R. Lewis, Marco Platzner, Bernhard Rinner, Jim Torresen, and Xin Yao, eds. *Self-Aware Computing Systems: An Engineering Approach*. Springer, 2016.

Y. Li, S. Makar, and S. Mitra. “CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns”. In: *2008 Design, Automation and Test in Europe*. Mar. 2008, pp. 885–890.

Martin Radetzki, Chaochao Feng, Xueqian Zhao, and Axel Jantsch. “Methods for Fault Tolerance in Networks-on-Chip”. In: *ACM Computing Surveys* 46.1 (July 2013), 8:1–8:38.

References III

Santanu Sarma, Nikil Dutt, N. Venkatasubramaniana, A. Nicolau, and P. Gupta. *CyberPhysical-System-On-Chip (CPSoC): Sensor-Actuator Rich Self-Aware Computational Platform*. Tech. rep. CECS Technical Report No: CECS TR–13–06. Irvine, CA 92697-2620, USA: Center for Embedded Computer Systems University of California, Irvine, May 2013.

Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. “On-Chip Self-Awareness Using Cyberphysical-Systems-On-Chip (CPSoC)”. In: *Proceedings of the 12th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. New Delhi, India, Oct. 2014.

Santanu Sarma, Nikil Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian. “CyberPhysical-System-On-Chip (CPSoC): A Self-Aware MPSoC Paradigm with Cross-Layer Virtual Sensing and Actuation”. In: *Proceedngs of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*. Grenoble, France, Mar. 2015.

Santanu Sarma, T. Muck, L. A.D. Bathen, N. Dutt, and A. Nicolau. “SmartBalance: A Sensing-Driven Linux Load Balancer for Energy Efficiency of Heterogeneous MPSoCs”. In: *Proceedings of the Design Automation Conference*. July 2015.

References IV

Santanu Sarma and Nikil Dutt. "Cross-Layer Exploration of Heterogeneous Multicore Processor Configurations". In: *VLSI Design Conference*. 2015.

David Tennenhouse. "Proactive Computing". In: *Communications of the ACM* 43.5 (May 2000), pp. 43–50.

Junshi Wang, Masoumeh Ebrahimi, Letian Huang, Axel Jantsch, and Guangjun Li. "Design of Fault-Tolerant and Reliable Networks-on-Chip". In: *IEEE Annual Symposium on VLSI (ISVLSI)*. Montpelllier, France, July 2015.

Junshi Wang, Masoumeh Ebrahimi, Letian Huang, Qiang Li, Guangjun Li, and Axel Jantsch. "Minimizing the System Impact of Router Faults by Means of Reconfiguration and Adaptive Routing". In: *Microprocessors and Microsystems* 51 (June 2017), pp. 252–263.

Youmin Zhang and Jin Jiang. "Bibliographical review on reconfigurable fault-tolerant control systems". In: *Annual reviews in Control* 32 (2008), pp. 229–252.