A Framework for Self-Awareness in Artificial Subjects

Axel Jantsch

Vienna University of Technology
Austria

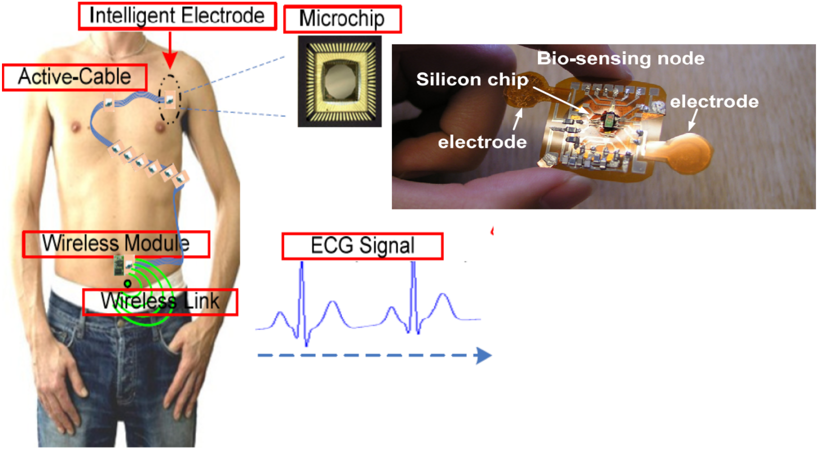
Seminar at the Brno University of Technology
19 March 2015

Sometimes Machines Behave Silly

The Benefits of Awareness

- Better functionality in different contexts
- Context depending performance
- Appropriate reaction in presence of faults

Self-Awareness for Resource Constrained, Insect-like Gadgets



Properties of Awareness

- Not all information is necessary
- More information does not imply more awareness

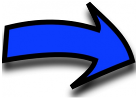
- Raw data is interpreted/abstracted
- Data interpretation is “meaningful”
- The drawn conclusions are “robust”
- The reaction is appropriate

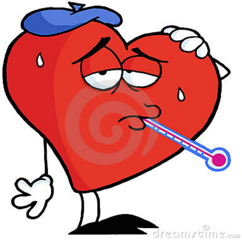
BioPatch: Temperature Sensor

$$\text{measured temperature} \begin{cases} < 20 & \to \text{person is dead} \\ \text{in } [20, 32] & \to \text{alive, life threatening} \\ \text{in } [32, 36] & \to \text{worrying, not life threatening} \\ \text{in } [36, 37] & \to \text{normal} \\ \text{in } [37, 37.5] & \to \text{elevated, not worrying} \\ \text{in } [37.5, 39.5] & \to \text{fever} \\ \text{in } [39.5, 43] & \to \text{high fever, life threatening} \\ > 43 & \to \text{person is dead} \end{cases}$$

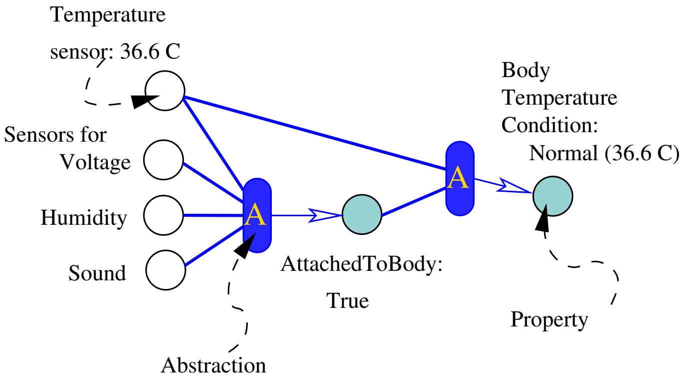
Abstractions and Models

Abstraction: Mapping of Measurements $\Rightarrow$ Properties



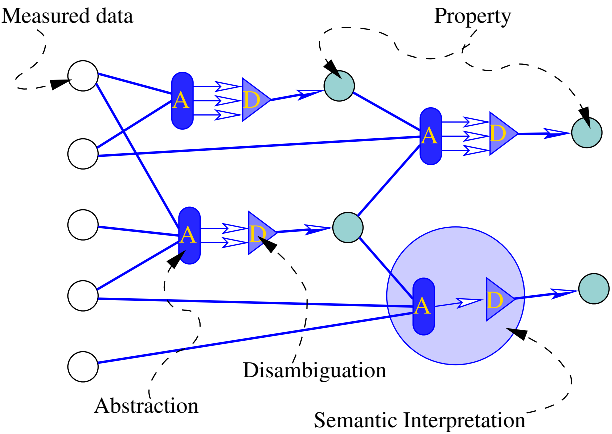


Abstractions and Models



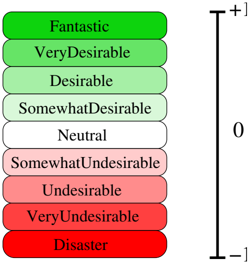
Disambiguation

Selection among several interpretations



Desirability Scale

A value range that captures the desirability of something



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

History

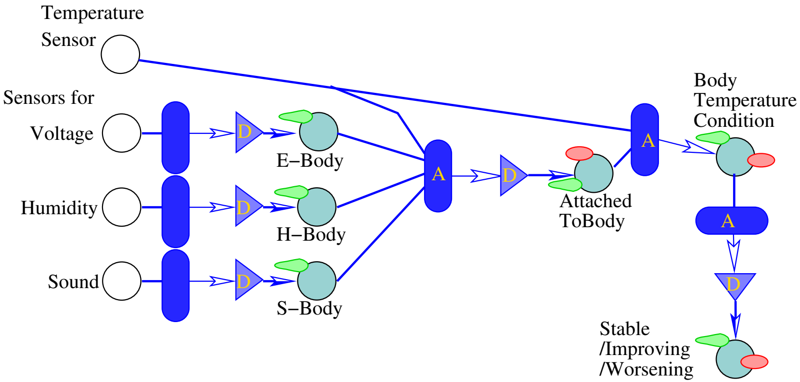
History of a Property The evolution of the values of a property.

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.

Sensors and properties of the BioPatch



Expectations

Expectation on Environment

- all implicit and explicit assumptions about the environment;
- a value range for each of the monitored properties.

Expectation on Subject

- all implicit and explicit assumptions about the subject;
- a value range for each of its monitored properties.

Goals

Sub-Goal A sub-goal of the subject is a desired value range of a property of the subject or its environment.

Goal A goal consists of one or several sub-goals.

Purpose The purpose of a subject is to achieve all its defined goals.

Inspection and Simulation

Self Inspection Engine is a mapping from a set of properties onto a desirability scale;

Model Transformation Given a model and a set of actions, a transformation applies actions and derives the new values for all properties.

Simulation Given a model and a set of potential actions, a simulation is a sequence of transformations applied onto the model resulting in a new, updated model.

Awareness of a Property

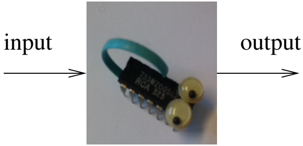
- The subject makes observations and derives the property by means of a meaningful semantic interpretation (**Meaning Condition**).
- The semantic interpretation is robust (**Robustness Condition**).
- There is a meaningful semantic attribution into a desirability scale (**Attribution Condition**).
- The subject reacts appropriately to its perception of the property (**Appropriateness Condition**).
- A history of the evolution of the property over time is maintained (**History Condition**).

Awareness of a Subject

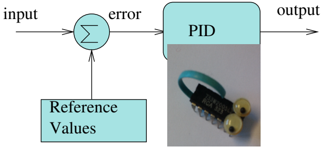
- The subject can assess how well it meets all its goals (**Goal Condition**).
- The subject can assess how well the goals are achieved over time and when its performance is improving or deteriorating (**Goal History Condition**).

Levels of Awareness

Level 0 - Functional: Behaviour is an immediate function of input.

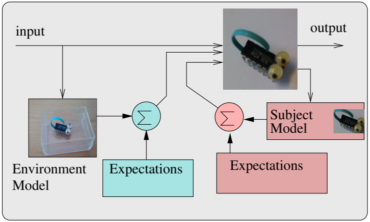


Level 1 - Adaptive: Output is an adaptive reaction to the input and a reference value (PID controller).

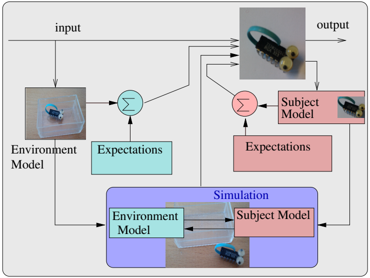


Levels of Awareness

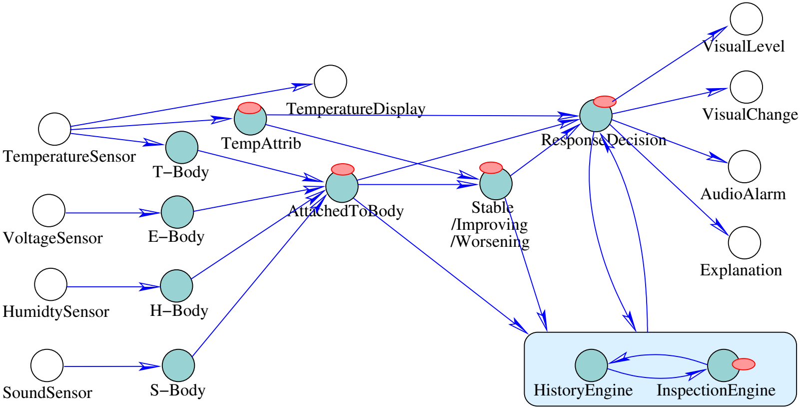
Level 2 - Self-aware: System represents some of its own properties and its environment as an abstraction. The models are related to desirable reference points.



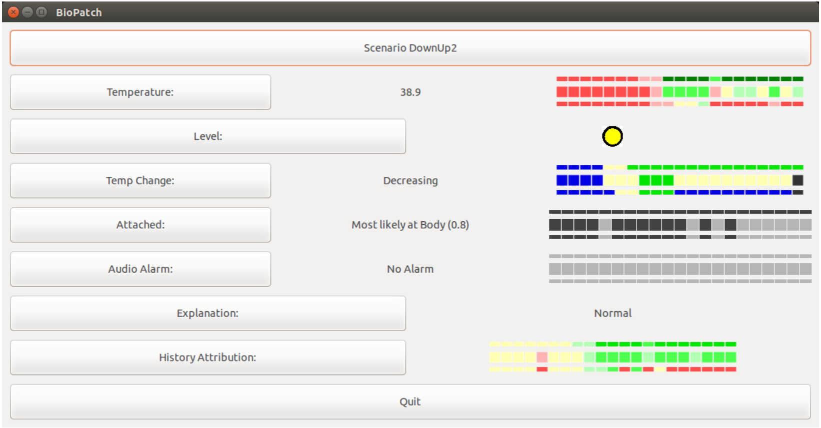
Level 3 - Predictive: System can simulate the effect of future input and of its own actions on the Self-Rep and the environment.



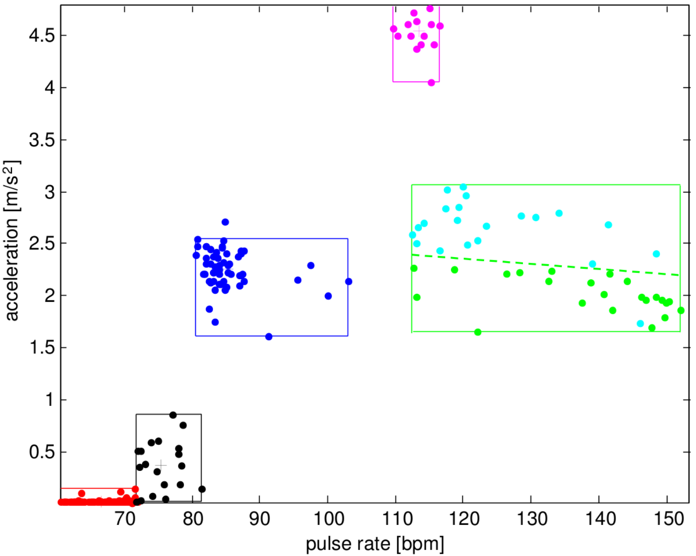
BioPatch Example



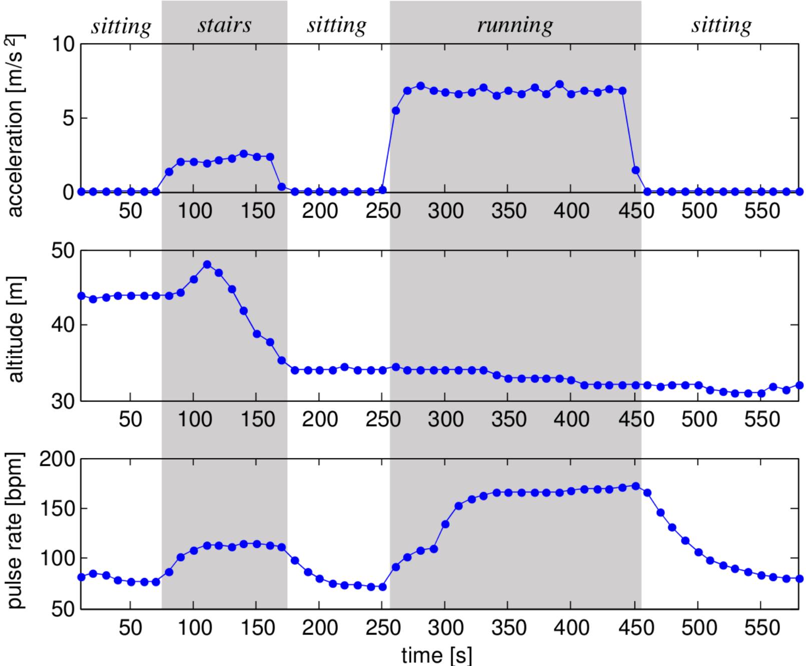
BioPatch Example



BioPatch Monitoring



BioPatch Monitoring



Example Approaches

- HAMSoC: Hierarchical Agent Monitored Systems on Chip
- SEEC: A Framework for Self-Aware Computing
- CPSoC: A Sensor-rich SoC Platform

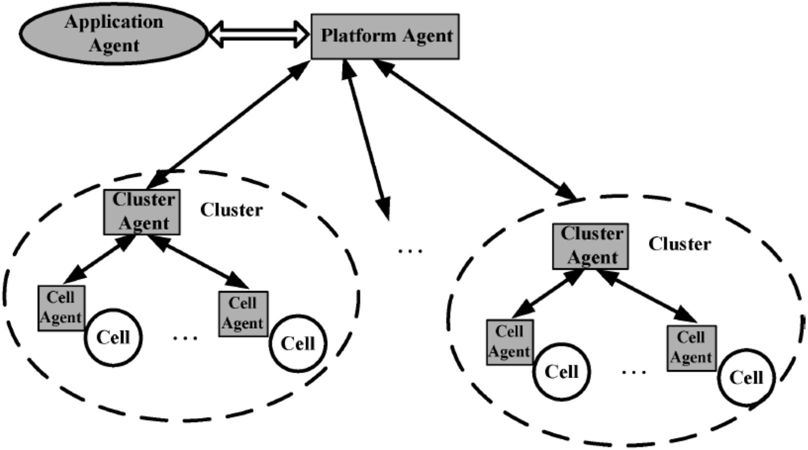
HAMSoC - A Hierarchical Agent Monitored System on Chip

- Self-monitoring design platform for multi-core SoCs
- Three levels of agents: cell, cluster, platform
- Dedicated design layer for self-awareness and adaptivity
- Application: Power management in NoC based multi-core SoC

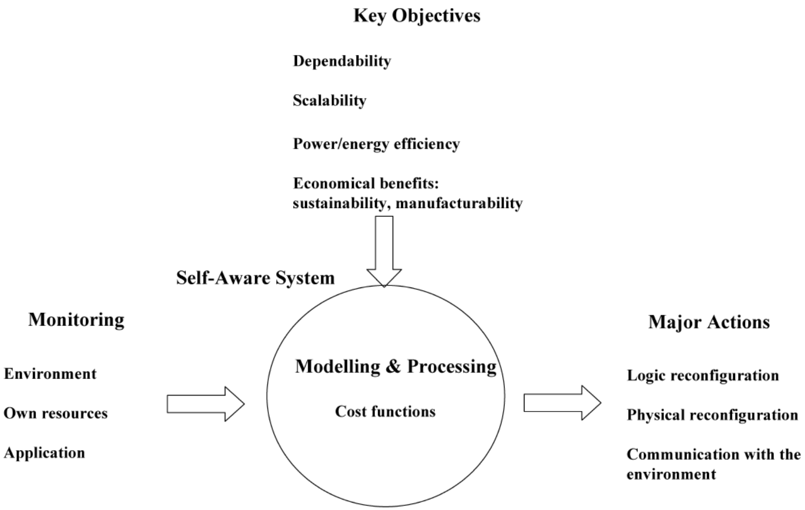
Liang Guang, Ethiopia Nigussie, Pekka Rantala, Jouni Isoaho, and Hannu Tenhunen. "Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip". In: *ACM Trans. Embed. Comput. Syst.* 9.3 (2010), pp. 1–24

Liang Guang. "Hierarchical Agent-based Adaptation for Self-Aware Embedded Computing Systems". PhD thesis. Turku, Finland: University of Turku, 2012

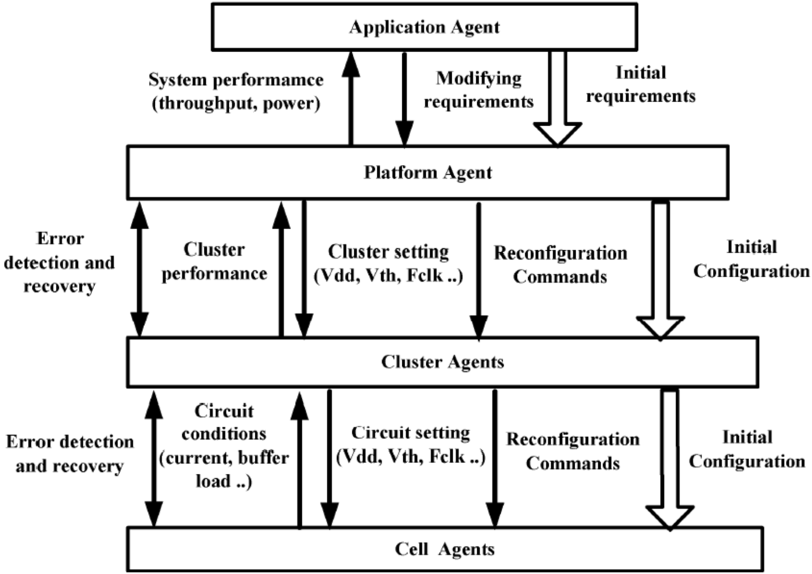
HAMSoC - A Hierarchical Agent Monitored SoC



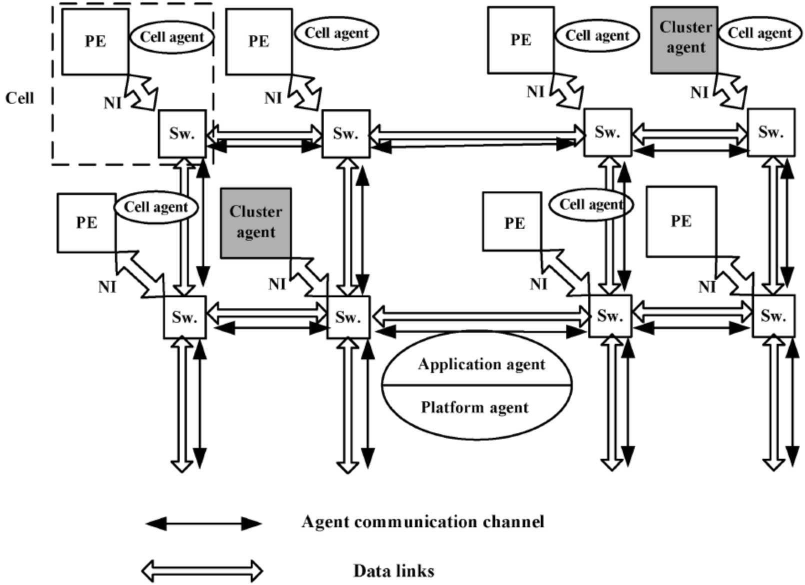
HAMSoC - A Hierarchical Agent Monitored SoC



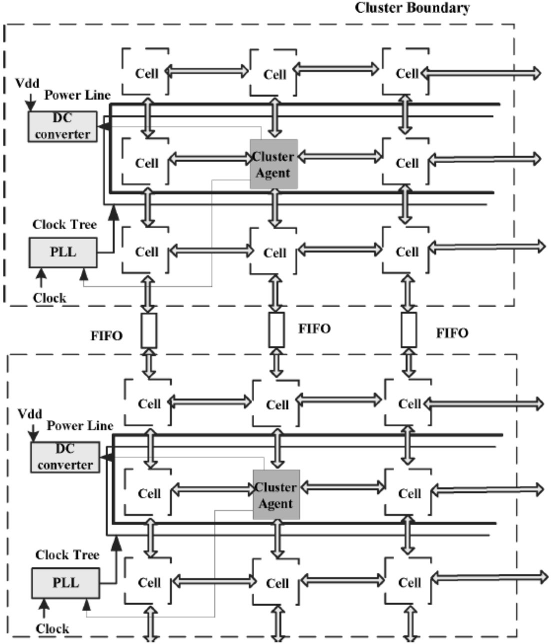
HAMSoC - A Hierarchical Agent Monitored SoC



HAMSoC - A Hierarchical Agent Monitored SoC



HAMSoC - A Hierarchical Agent Monitored SoC



HAMSoC - A Hierarchical Agent Monitored SoC

Normalized Communication Energy of Three Energy-Efficient Architectures

Traffic Pattern	Cluster-based DVFS	Single-domain DVFS	Static Voltage Island
1	80.90%	106.29%	1
2	79.36%	101.98%	1
3	96.21%	100.41%	1
4	90.18%	106.52%	1

HAMSoC - A Hierarchical Agent Monitored SoC

Normalized Communication Latencies of Three Energy-Efficient Architectures

Traffic Pattern	Cluster-based DVFS	Single-domain DVFS	Static Voltage Island
1	165.34%	131.63%	1
2	144.37%	142.44%	1
3	123.59%	108.44%	1
4	124.00%	121.38%	1

HAMSoC - A Hierarchical Agent Monitored SoC

Area Overhead of Three Energy-Efficient Architectures (in mm^2)

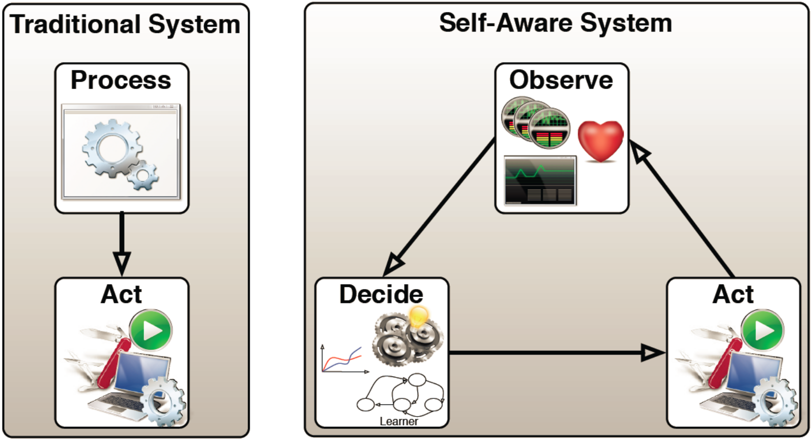
Architecture	Links	Switches	DC Regulators & PLLs	Total	% of a Chip Size
Cluster-based DVFS	23.35	12.88	10.63	46.86	17.04%
Single-domain DVFS	23.35	12.88	0.38	36.61	13.31%
Static voltage island	22.63	12.88	0	35.51	12.91%

SEEC - A Framework for Self-Aware Computing

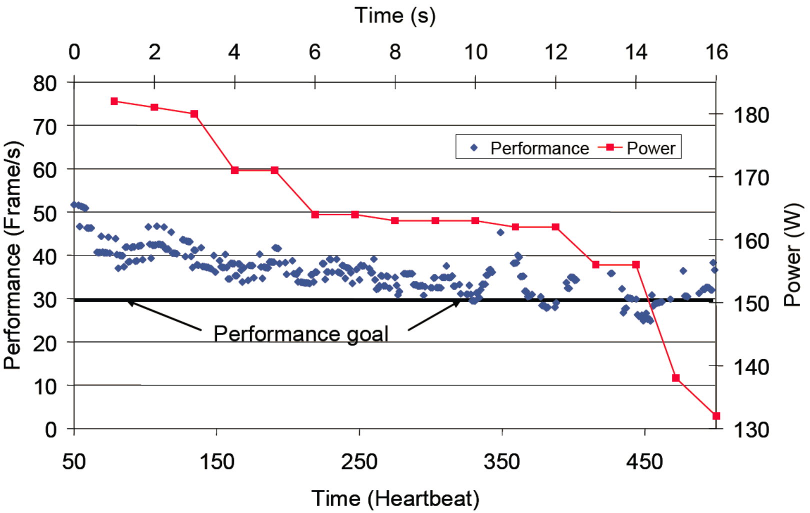
- The applications specify goals
- The platform provides possible actions
- SEEC monitors the application and decides upon actions
- Observe - Decide - Act based control loop

Henry Hoffmann, Martina Maggio, Marco D Santambrogio, Alberto Leva, and Anant Agarwal. *Seec: A framework for self-aware computing*. Tech. rep. MIT-CSAIL-TR-2010-049. Cambrige, Massachusetts: MIT, Oct. 2010

SEEC - A Framework for Self-Aware Computing



SEEC - A Framework for Self-Aware Computing



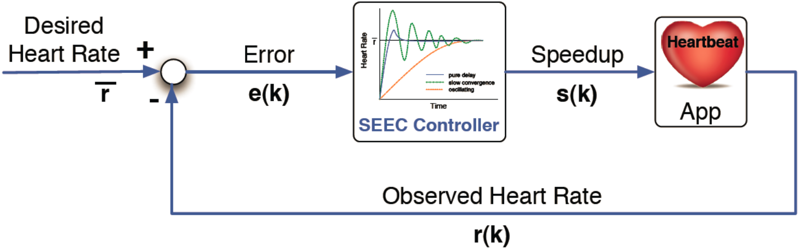
x264 encoder with 30 frames/sec performance goal.

SEEC - A Framework for Self-Aware Computing

Phase	Applications Developer	Systems Developer	SEEC Framework
Observation	Specify application goals and performance	-	Read goals and performance
Decision	-	-	Determine how much to speed up the application
Action	-	Specify actions and a function that performs actions	Initiate actions based on result of decision phase

Roles in the SEEC development framework.

SEEC - A Framework for Self-Aware Computing

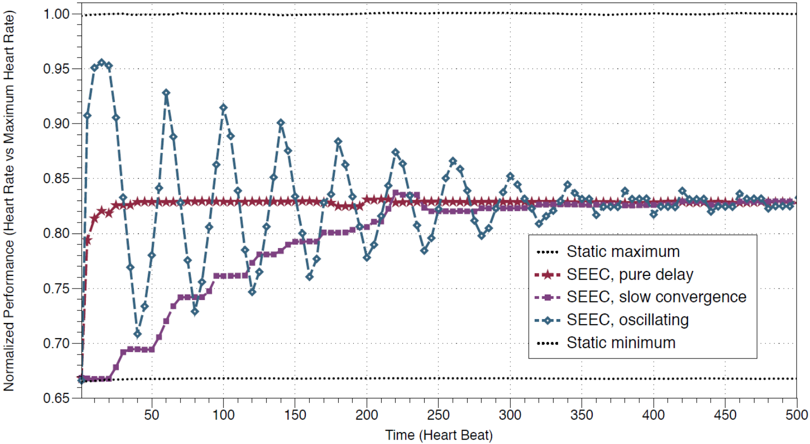


SEEC - A Framework for Self-Aware Computing

Controller	Action Set	Actuation Function	Tradeoffs
Frequency Scaler	CPU Speeds	Change CPU speed	Power vs Speed
Core Allocator	Number of available cores	Change affinity masks	Power vs Speed
DRAM Allocator	Number of available DRAM controllers	Change NUMA page allocation	Power vs Speed
Power Manager	CPU speed and in-use cores	Change CPU speed and affinity masks	Power vs Speed
Adaptive Video Encoder	Encoding Parameters and Algorithm	Change parameters, use different algorithms	Video Quality vs Speed

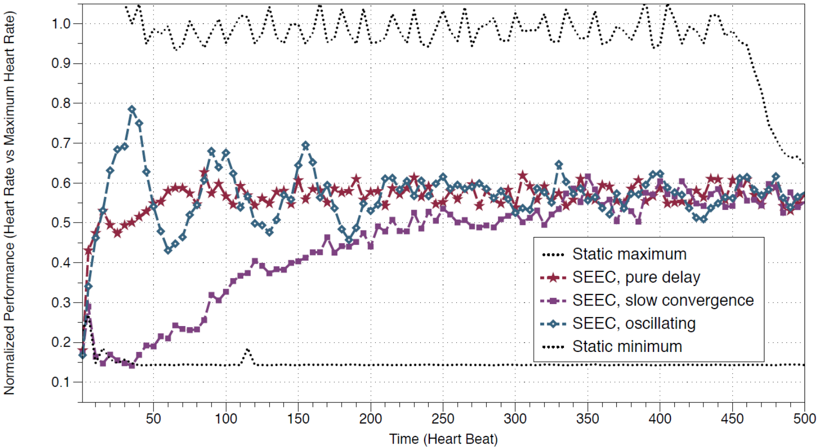
Application examples

SEEC - A Framework for Self-Aware Computing



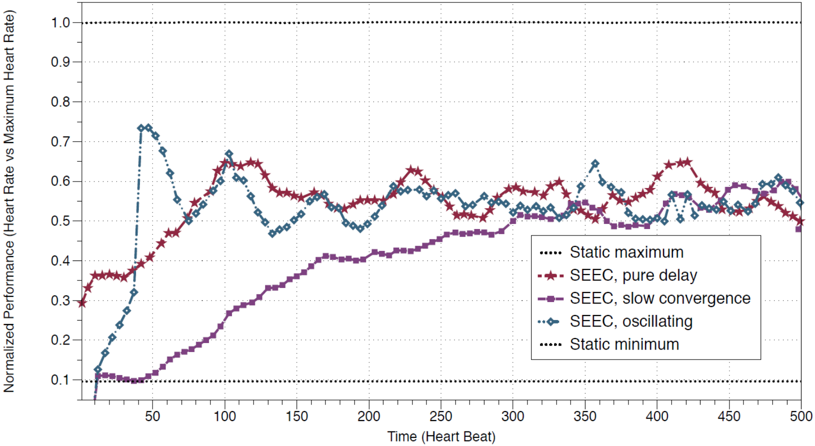
Frequency scaling for the swaptions application (PARSEC benchmark)

SEEC - A Framework for Self-Aware Computing



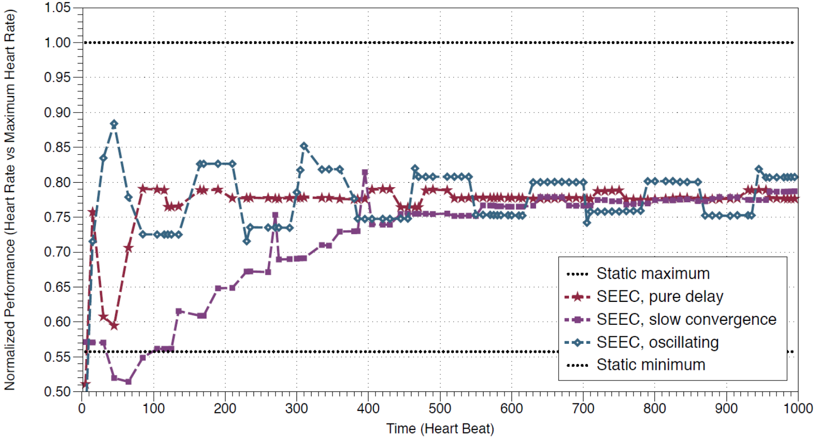
Core allocator for swaptions

SEEC - A Framework for Self-Aware Computing



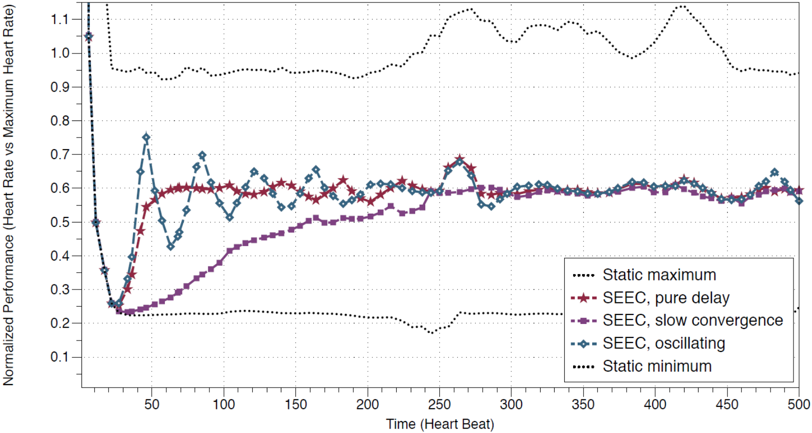
Power manager (DRAM controllers, number of cores, frequency) for swaptions

SEEC - A Framework for Self-Aware Computing



Memory allocator for STREAM (PARSEC benchmark)

SEEC - A Framework for Self-Aware Computing



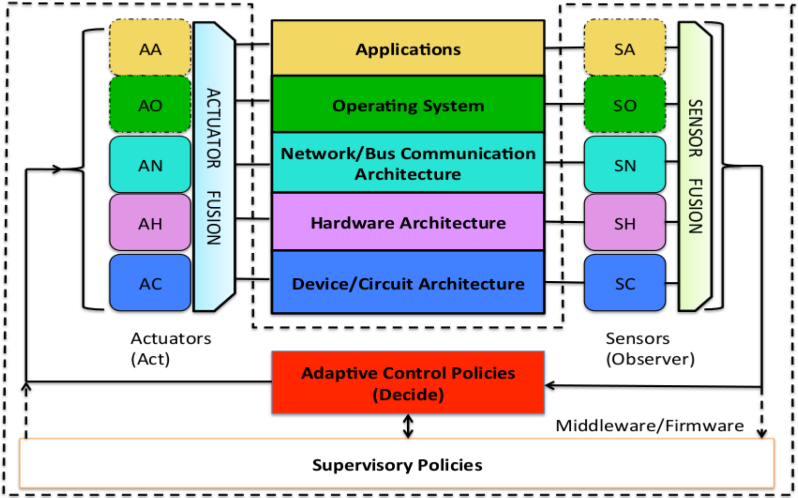
Adaptive video encoder

CPSoC - A Sensor Rich SoC Platform

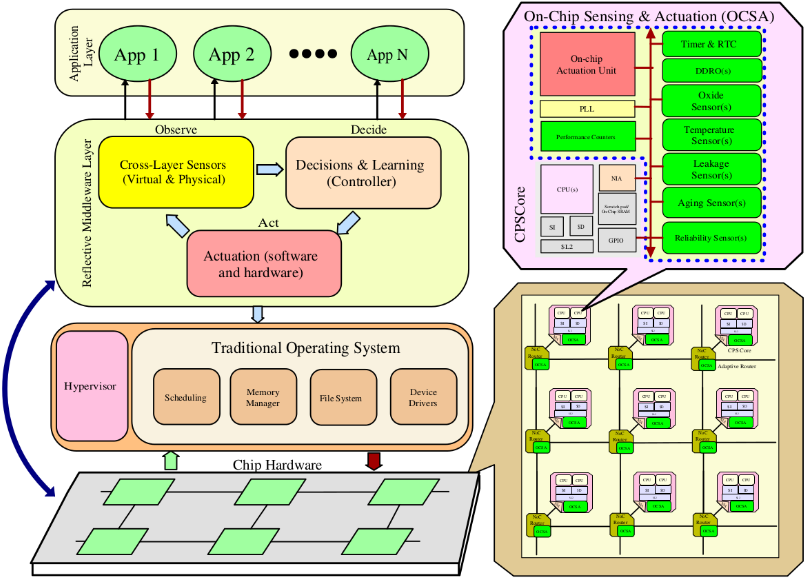
- Sensors and actuators at five layers:
  - Device/ circuit architecture
  - Hardware architecture
  - Network/Bus communication architecture
  - Operating system
  - Application
- Observe-decide-act paradigm
- Codesign of control, communication and computing

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

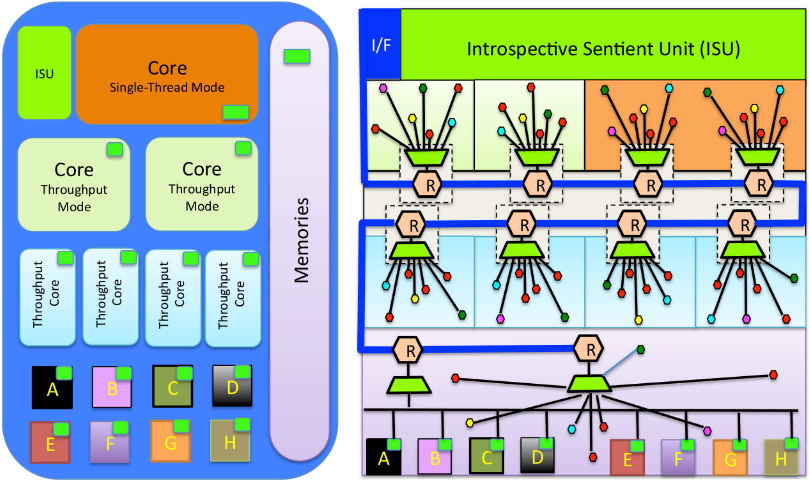
CPSoC - A Sensor Rich SoC Platform



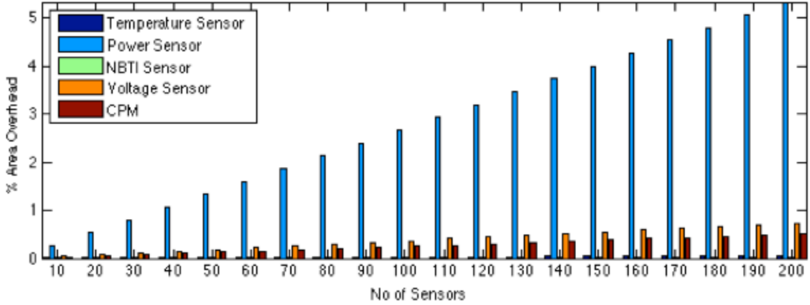
CPSoC - A Sensor Rich SoC Platform



CPSoC - A Sensor Rich SoC Platform

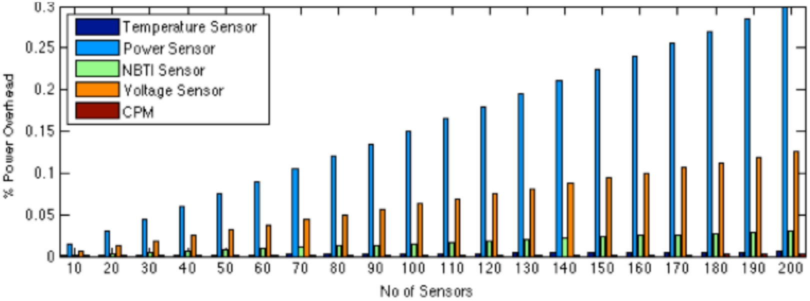


CPSoC - A Sensor Rich SoC Platform



Virtual sensing reduces the area overhead for 1000 sensors from 7.3% to 0.6%.

CPSoC - A Sensor Rich SoC Platform



Virtual sensing reduces the power overhead for 1000 sensors from 1.7% to 0.3%.

CPSoC - A Sensor Rich SoC Platform

Virtual/Physical Sensing and Actuations Across Layers

Layers	Virtual/Physical Sensors	Virtual/Physical Actuators
Application	Workload, Power, Energy and Execution Time, Phases	Loop Perforation, Approximation, Algorithmic Choice, Transformations
Operating System	System Utilization and Peripheral States	Task Allocation, Partitioning, Scheduling Migration, Duty Cycling
Network/ Bus Communication	Bandwidth, Packet/Flit Status and Channel Status, Congestion	Adaptive Routing,Dynamic BW Allocation and Ch. no and Direction Control
Hardware Architecture	Cache Misses, Miss Rate, Access Rate, IPC, Throughput, MLP	Cache & Issue-Width Sizing, Reconfiguration Resource Provisioning, Static/Dynamic Redundancy
Circuit/Device	Circuit Delay, Aging, Leakage Temperature, Oxide Breakdown	DVFS, ABB, Voltage Frequncy Island Clock Gating, Power Gating

Summary of Self-Aware Properties

- Awareness and self-awareness are useful properties
  - Context dependent functionality
  - Context dependent performance
  - Appropriate behavior in all situations
- Necessary features:
  - Data abstraction
  - Disambiguation
  - Desirability mapping
  - History maintenance
  - Expectations and goals
  - Self-inspection
  - Prediction and simulation

Challenges:

- Application specific selection and tuning of features
- Online learning and adaptation
- Efficient implementation