Computing within Materials: Self-Adaptive Materials and Self-organizing Agents

From Macroscale to Microscale Computing

Stefan Bosse
University of Bremen, Dept. Mathematics & Computer Science
University of Koblenz, Fac. Computer Science
11.4.2018
sbosse@uni-bremen.de

1. Inhalt

1. Inhalt 1
2. Material Informatics: Material-integrated Computing 2
2.1. Smart Materials 2
2.2. Material Informatics: Computing within Materials 4
2.3. Computing Power and Efficiency 4
2.4. Technologies 5
2.5. Comparison of Computers 6
3. Optimization and Adaptation 6
3.1. Physical Model 7
3.2. Optimization Goals 7
3.3. Algorithms 9
4. Multi-Agent Systems 9
4.1. Sensor Data Distribution 9
4.2. Distributed Observable Computation 10
4.3. Distributed Adaptation 11
5. Simulation 12
5.1. Multi-domain Simulation Framework 12
5.2. Simulation Example 13
5.3. Simulation Results 14
6. Conclusions and Outlook 14
7. References 14

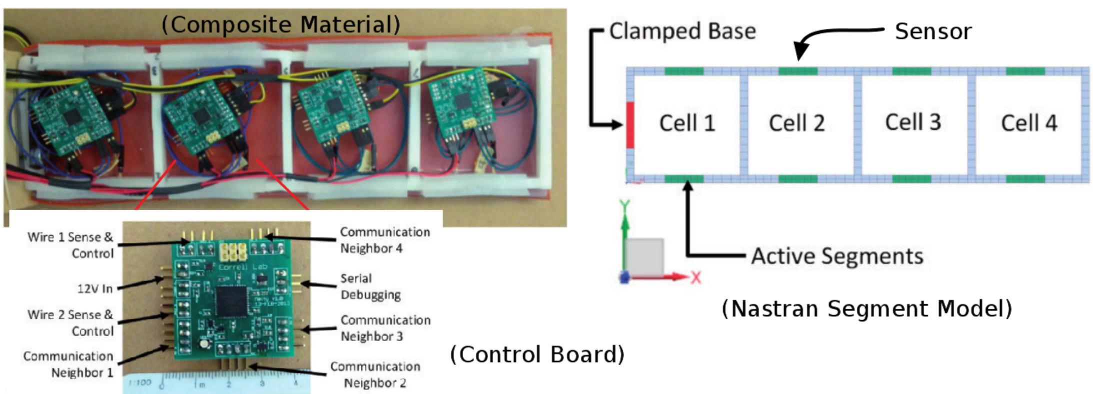
2. Material Informatics: Material-integrated Computing

2.1. Smart Materials

Smart Materials: Fusion of Sensorial- and Adaptive Materials with Information Processing

Sensorial Material
: A material or structure with integrated **sensing** and **data processing** (ICT)

Adaptive Material
: A material or structure with integrated sensing, data processing and **actuation** that can control and change material or structure **properties**



McEvoy, 2015 [1]

In our understanding, a smart material provides the following major features:

1. **Perception** using various kinds of sensors, e.g., measuring of strain, displacement, temperature, pressure, forces;
2. **Changing** of local material and structure properties by actuators, e.g., stiffness or damping variation;

3. **Integrated Information and Communication Technologies** (ICT);
4. **Distributed** Approach: Local sensor processing and actuator control - Global cooperation and coordination.
5. **Robustness** and **Self-Organization**

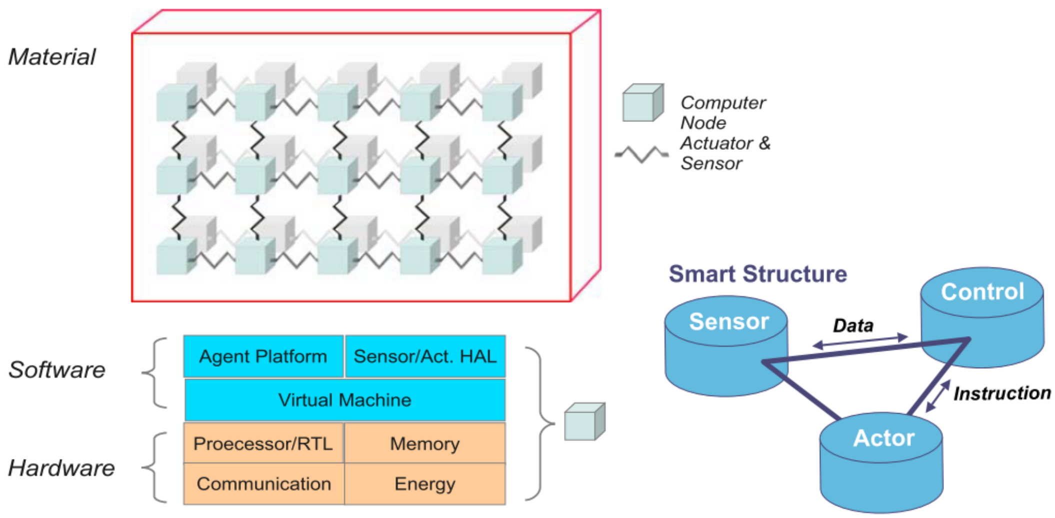
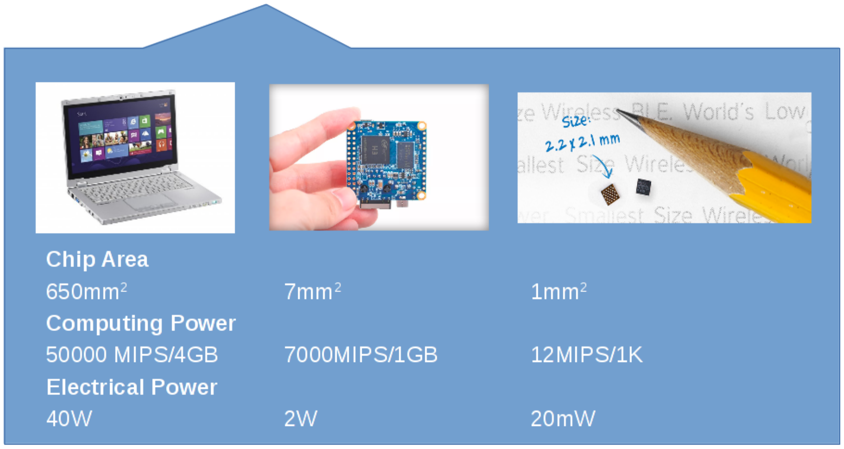


Fig. 1. (Left) General architecture of a Sensorial- and Adaptive Material = Smart Material (Right) Functional Decomposition: Sensing, Acting, Processing ↔ Data + Instruction Streams

2.2. Material Informatics: Computing within Materials

- Traditionally computation is separated from sensing and control
- Smart Materials poses the **tight coupling** of computation, communication, sensing, and control with **loosely coupled nano computers**
- **Algorithmic scaling** and **distribution** are required



2.3. Computing Power and Efficiency

➤ A normalized computing efficiency of a computer (considering only the data processing unit) can be defined by

$$\varepsilon = \frac{C}{AP} \tag{1}$$

A

Chip Area in square millimetres

C

Computing Power in (Integer) Mega Instructions Per Second (MIPS)

P

Electrical Power in Watt

➤ The computing efficiency can be used to compare different computers and devices, i.e., giving a scaling factor:
$s=\epsilon_x/\epsilon_y$

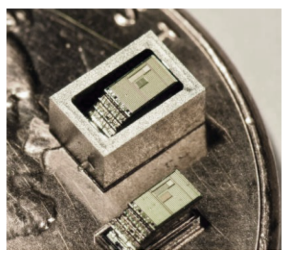
2.4. Technologies

Existing "Nano" Computers

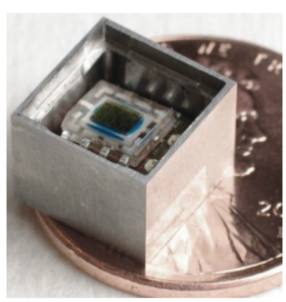
➤ **Smart Dust** → Millions of loosely coupled nano computers, e.g.,

- ❏ embedded in materials
- ❏ scattered on surfaces
- ❏ dispersed in liquids, foils, ..
- ❏ about 10mm^3 volume

Micro Mote M3



ELM System



2.5. Comparison of Computers

	Micro Mote (M3)	**ELM System**	**Atmel Tiny 20**	**Freescale KL03**	**ARM Cortex Smart Phone**
Processor	Arm Cortex M0	C8051F990 (SL)	AVR	Arm Cotex M0+	Arm Cortex A9
Clock	740kHz max.	32kHz	-	48MHz	1GHz
CPU Chip Area	$0.1mm^2$	$9mm^2$	$1mm^2$	$4mm^2$	$7mm^2$/ROM
Sensors	Temperature		-	-	Temp, Light, Sound, Accel., Press., Magn.
Communication	900MHz radio, optical	optical	electrical	-	3G/4G, WLAN, USB, Blue-tooth, NFC
Harvester, Battery	Solar cell, Thin film	Solar cell, Coin	-	-	-
Power Consumption	70mW / CPU	160mW / CPU	20mW	3mW @ 48MHz	100mW avg.,
Manufacturing	180nm CMOS	-	-	-	40nm CMOS
Package	Wire bonded	Silicon Stack	PCB	Single Chip	Single Chip
Computing Eff. ϵ	**150**	**0.02**	**0.6**	**4.0**	**0.53**

3. Optimization and Adaptation

3.1. Physical Model

➤ A component composed of a Smart Material consists of:

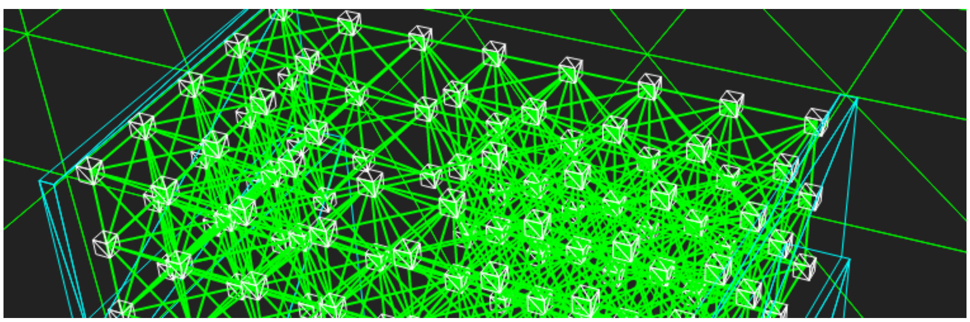
- ❏ Mass (body) nodes or material regions
- ❏ Interconnection elements with parameterizable properties

Multi-body Physics Model

Solid body mass nodes with springs connecting nodes:

➤ A spring is an actuator with two *optimization target variables*: Stiffness s, Damping d

➤ Each spring is a strain sensor delivering the sensor value σ for the computation of the *observation variable*



3.2. Optimization Goals

Reduction of global and local stress, strain, or forces of arbitrariely shaped components under varying load situations

---

Control and Optimization Cycle

1. Perception using sensors
2. Comparison of local and global observation variables
3. Modification of material parameters by actuators

3.3. Algorithms

Global

- A global observation variable x is used to compute a correction of the target variable s
- The correction function k_x uses the ratio of the local and the global observable

```
do with x ∈ {ε,σ,U}
 X:=0; ∀n∈ℕ do X := X + x_n
 X := X/|ℕ|
 ∀n∈ℕ do
  r_n := k_x(x_n/X, s_n)
  s_n := s_n * r_n
until |Err| < Err0
```

Segment

- Network is partitioned in segments
- Observation variable is computed for each segment
- Target variable is computed for each segment

```
do with x ∈ {ε,σ,U}
 ∀Si∈𝕊 do
  Xs,i:=0;
  ∀n∈Si do X_s,i := X_s,i + x_n
  X_s,i := X_s,i / |S_i|
  ∀n∈S_i do
    r_n := k_x(x_n/X_s,i, s_n)
    s_n := s_n * r_n
until |Err| < Err0
```

Neighbour

- Neighbour node negotiation
- Observation variable is limited to node boundary

- ➤ Swapping of increments of target variable

```
do with x ∈ {ϵ,σ,U}
 ∀{n_i,n_j ∈ N | i≠j ∧
  |pos(n_i)-pos(n_j)|=1} do
  if x_i/x_j < 1 ∧
     s_i-Δs > s_0 ∧
     s_j+Δs < s_1 then
    s_i := s_i - Δs
    s_j := s_j + Δs
  end if
always
```

4. Multi-Agent Systems

4.1. Sensor Data Distribution

- ➤ Each node agent sends its sensor values to neighbour nodes within a range of radius 1
- ➤ This completes the set of sensor each node requires
- ➤ Remote signals are used for sensor distribution using &Delta-distance routing

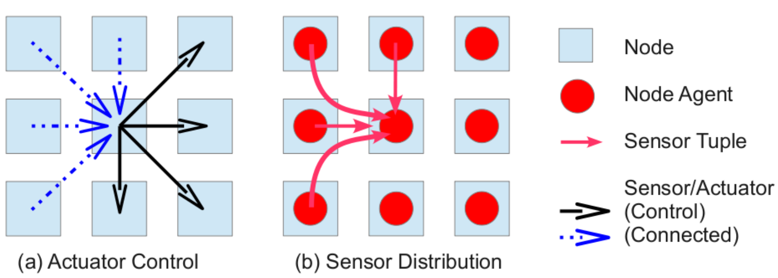


Fig. 2. Sensor Distribution by neighbour nodes using using remote signals (or tuples)

4.2. Distributed Observable Computation

- **Global observable** computation in a large-scale distributed network is expensive and difficult (failures)
  - Random walk and directed diffusion can be used to **approximate** a global observable
- **Segmented** approach requires network segmentation
  - Without central instance difficult;
  - Instead a floating segment window is placed around **each** node
  - Each node has observable from all direct neighbours (North, South, West, East, Up, Down)
  - A *chained distribution* of data is used in each segment (N nodes $\leftrightarrow$ N segments!)
  - Observable values with distance r=1 and r=2 are collected by each node to compute region observable

Distance r=1
: Observable from direct neighbours

Distance r=2
: Direct neighbours deliver also values form their neighbours (opposite to request direction)

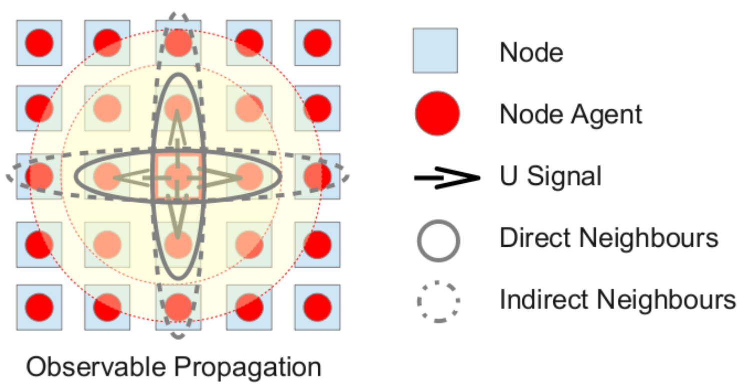


Fig. 3. Distribution of neighbour observable values for region accumulation using remote signals (or tuples)

4.3. Distributed Adaptation

- The global and the segment algorithm need no further distribute coordination
- After the region observable (global or segment) is computed the actuators (springs) of each node can be modified basing on ratio of the local and the region observable value
- Neighbour negotiation approach do not require a region observable

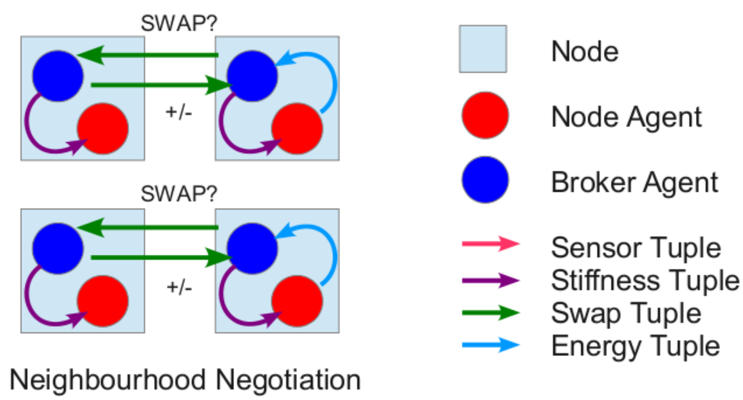


Fig. 4. Negotiation is used between two neighbour nodes to achieve a stiffness reconfiguration

5. Simulation

5.1. Multi-domain Simulation Framework

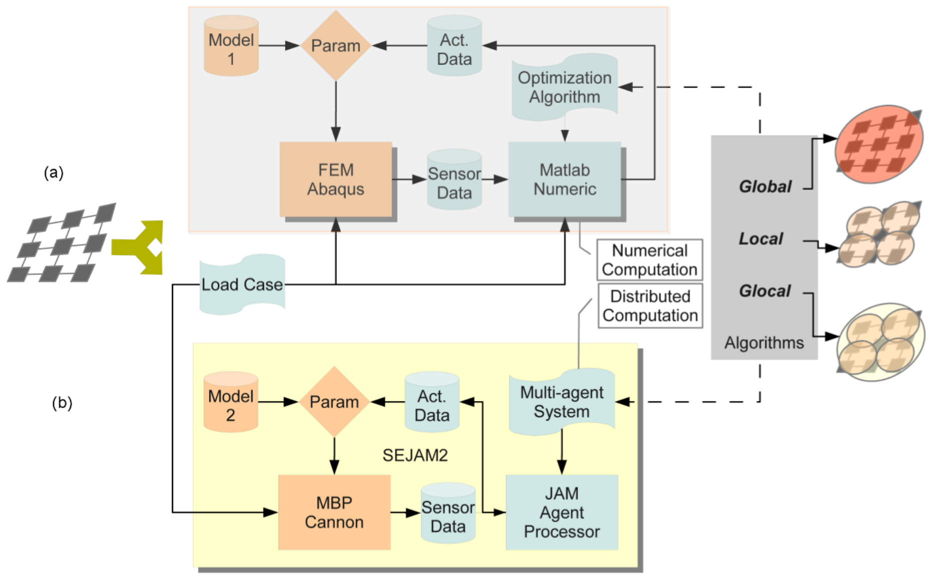


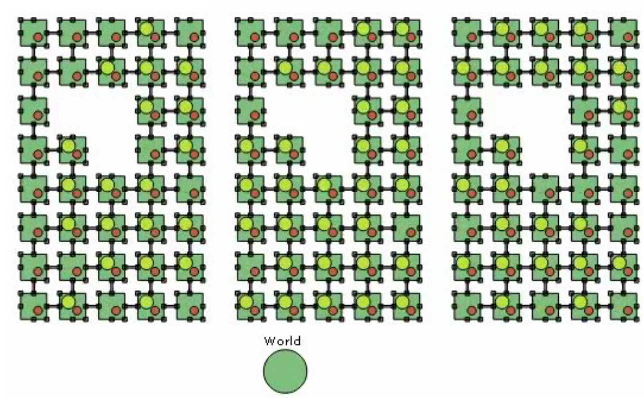
Fig. 5. Hybrid simulation environment with Abaqus, Matlab, and SEJAM2

5.2. Simulation Example

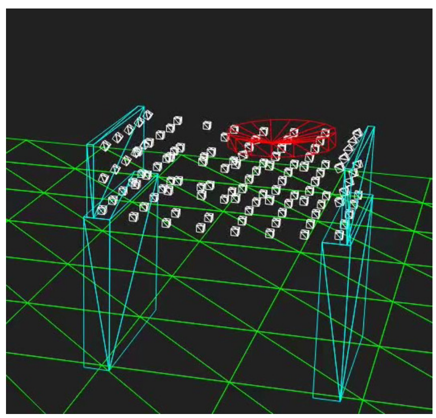
➤ Device under Test: Plate (8x5x3 nodes), large hole, external load

MAS World

➤ Event-based agent behaviour activates sensor processing, distribution, and adaptation only if something changes

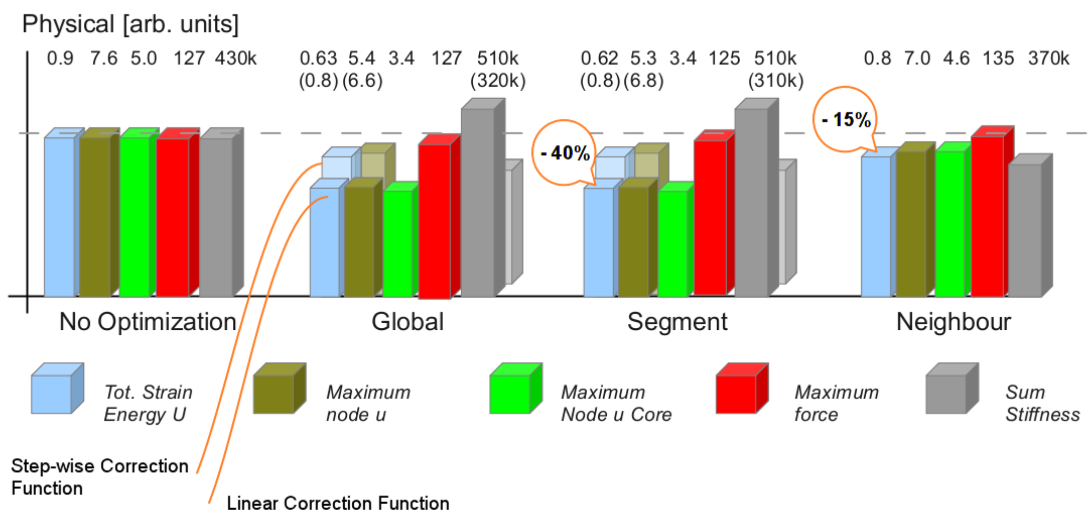


Physical World



5.3. Simulation Results

- Global and segment optimization achieves 40% decrease of total strain and maximum strain energy of mass elements using a linear correction function
- Neighbour negotiation is simple but not as efficient as segment approach



6. Conclusions and Outlook

- Smart Materials poses the **tight coupling** of computation, communication, sensing, and control with **loosely coupled nano computers**
- **Algorithmic scaling** and **distribution** are required
- Distributed information processing paradigma: Multi-agent Systems
- Multi-domain simulation enables the development and evaluation of different optimization strategies for smart adaptive materials and structures
- The SEJAM simulator enables the simulation and analysis of coupled physical and computational systems
- Global and segment optimization achieves 40% decrease of total strain and maximum strain energy of mass elements using a linear correction function

7. References

[1] M. A. McEvoy and N. Correll, “Thermoplastic variable stiffness composites with embedded, networked sensing, actuation, and control,” Journal of Composite Materials, vol. 49, no. 15, 2015.