Forschungsprofil

Priv.-Doz. Dr. Stefan Bosse
Universität Bremen, FB Mathematik & Informatik
sbosse@uni-bremen.de

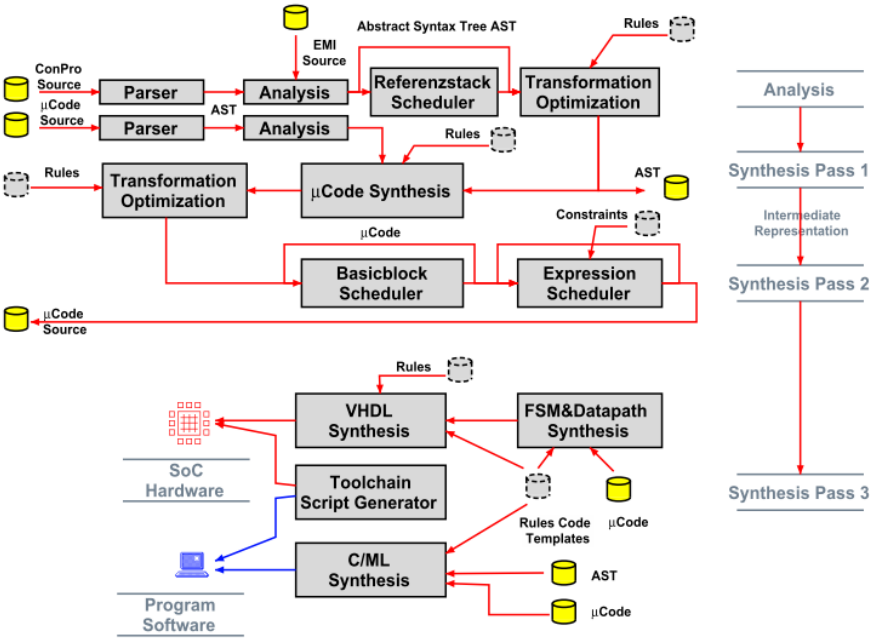
1. Inhalt

1. Inhalt 1
2. High-level Synthese, Compilerbau, Simulation 1
2.1. Entwurf eingebetteter paralleler Systeme 2
2.2. Komplexe Modellierungswerkzeuge und Frameworks 3
2.3. Multi-domain Simulation 3
3. Verteilte Systeme 4
3.1. Verteilte Funktionale Programmierung 4
3.2. Environmental Sensing: IoT & Clouds 4
3.3. Algorithmische Skalierung 5
3.4. Material-integrierte Intelligente Systeme (MIIS) 6
3.5. MIIS: Sensorische Materialien 6
3.6. MIIS: Adaptive Materialien 7
3.7. Social & Crowd Sensing 8
4. Agentenbasierte Verfahren und Plattformen 8
4.1. Mobile Agenten ⇔ Mobiler Code 9
4.2. Plattformen 10
4.3. Kommunikation & Interaktion 11
5. Self-* Systeme 11
5.1. Mustererkennung 11
5.2. Energiemanagement durch selbstorganiserte Verhandlung 12
5.3. Ereignisbasierte Sensordatenverarbeitung 12
6. Data Mining / Maschinelles Lernen 13
6.1. Strukturüberwachung 13
6.2. Agentenbasiertes und Verteiltes Lernen 14
6.3. Seismische Netzwerke 14

2. High-level Synthese, Compilerbau, Simulation

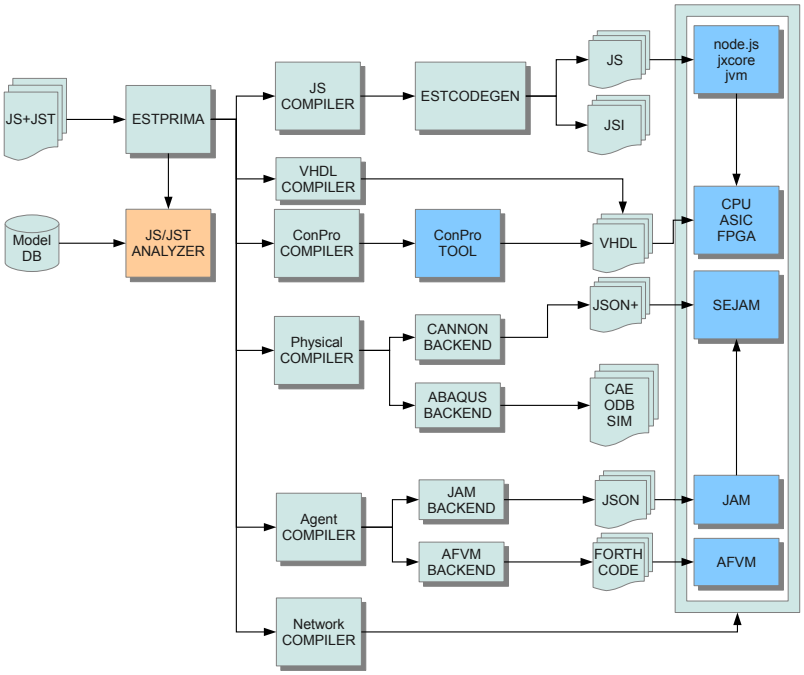
2.1. Entwurf eingebetteter paralleler Systeme

- Hardware-Software Co-Design mit ConPro (→ OCAML)
- ConPro: Concurrent Processing Synthesis & Programming Language
- Paralleles Modell: Hoare CSP mit geteilten Ressourcen und Konkurrenz



2.2. Komplexe Modellierungswerkzeuge und Frameworks

- Modellierung und Implementierung komplexer sensorischer und mechatronischer Systeme mit einer einheitlichen Modellierungs- und Programmiersprache → **JS+**: JavaScript + JavaScript Semantic Type System (**JST**)
- Ein einziges Tool für Software, Hardware, Agenten, Physikalische Systeme, Simulation usw.



2.3. Multi-domain Simulation

- Simulation von komplexen gekoppelten Systemen: Multiagentensystem + Physikalisches Systeme (Multikörperphysik) → Adaptive Materialien
- Hardware-in-the-loop Simulation möglich

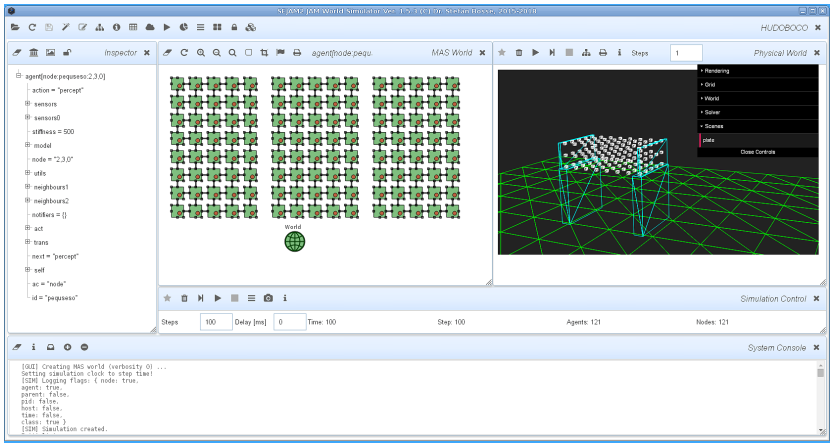


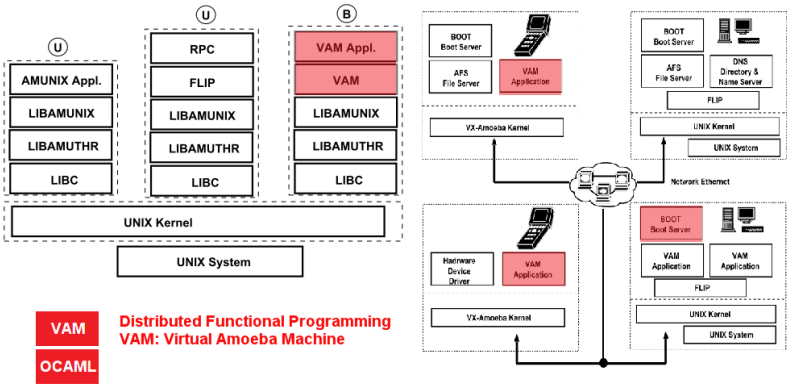
Abb. 1. SEJAM2: Simulation Environment for the JavaScript Agent Machine

3. Verteilte Systeme

3.1. Verteilte Funktionale Programmierung

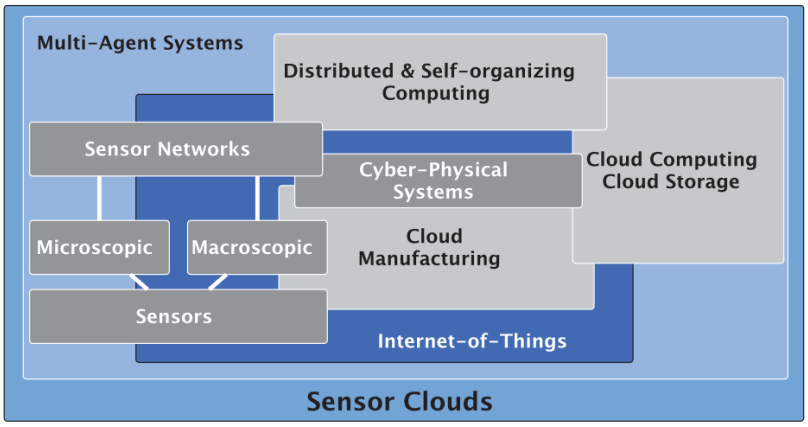
VAMNET: Virtual Amoeba Machine Network

Verteilte Funktionale Programmierung mit OCAML und dem objektorientierten verteilten Betriebssystem Amoeba (Konzepte)



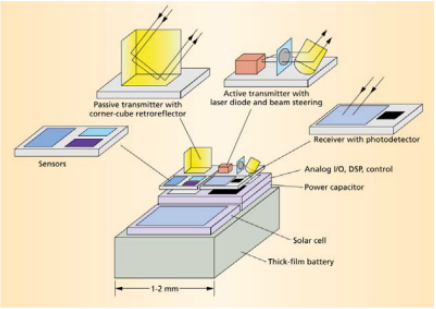
3.2. Environmental Sensing: IoT & Clouds

Lösung des verteilten Datenverarbeitungsproblems in stark heterogenen, unzuverlässigen, und dynamischen Umgebungen mittels autonomer und self-* Agentensysteme

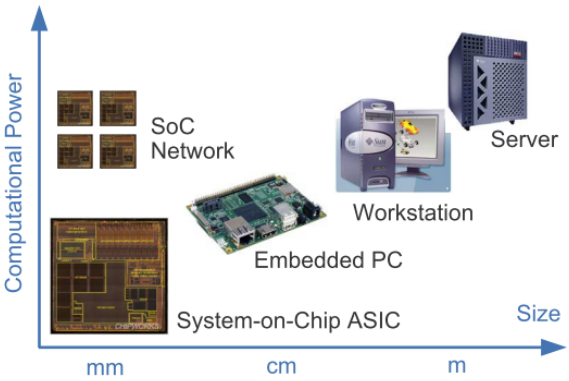


3.3. Algorithmische Skalierung

- Skalierung etablierter Algorithmen und Softwarearchitekturen wie
  - Kommunikation
  - Datenfusion
  - Maschinelles Lernen
- auf unzuverlässige Systeme unter Robustheitsaspekten bestehend aus:
  - Mikroskalige Computersystemen (1mm^3 Computer) die
  - großskalige Netzwerke bilden (Millionen von Knoten)!



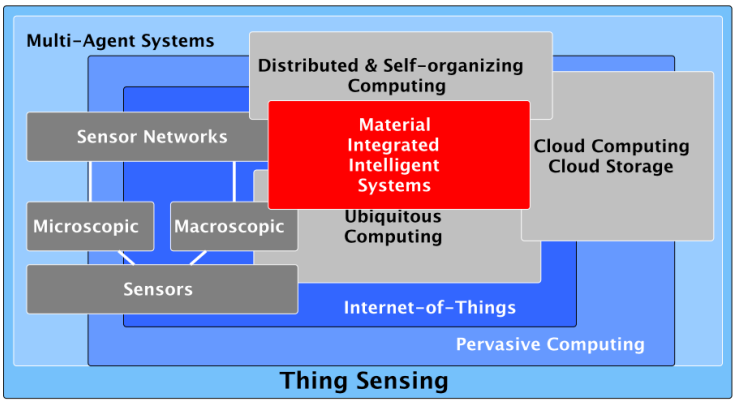
Smart Dust Sensor Mote



3.4. Material-integrierte Intelligente Systeme (MIIS)

- Großskalige Netzwerke aus mikroskaligen Computern die in Materialien und technische Strukturen integriert sind
- Hohe Anforderungen an Algorithmen: Wenig Speicher, geringe Rechenleistung, evtl. nur Ganzzahlarithmetik, Fehler, nur sporadisch aktiv
- Sie bieten intelligente Perzeption und Steuerungsmöglichkeiten / Interaktion
- Sie können "fühlen" und mit der Umgebung interagieren
- Intelligenz durch folgende Eigenschaften:
  - Autonomie und Robustheit

- Adaptivität und Rekonfiguration
- Organisation und Kooperation
- Lernen
- Resilienz



3.5. MIIS: Sensorische Materialien

- Material-integriete Sensorische Systeme bieten voll integrierte Datenverarbeitung, Kommunikation, Energieversorgung, Sensoren
- Hohe Anforderungen an Informationsgewinnung → Merkmalsexktraktion und Datenreduktion vor Ort
- Neue Konzepte der Interaktion/Kommunikation einzelner Knoten (Gruppenverhalten, datenzentrische Verfahren, ereignisbasierte Verfahren, Informationsclusterbildung) → IP Netzwerke ungeeignet

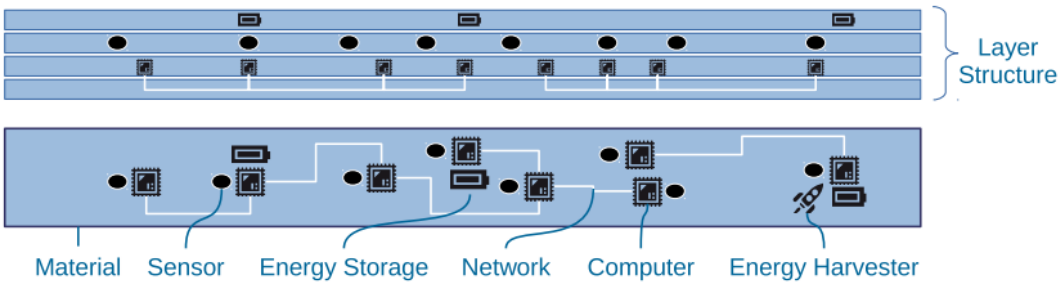


Abb. 2. Sensorisches Material im Schichtaufbau → technisch realisierbar durch System-in-Foil Technologie

3.6. MIIS: Adaptive Materialien

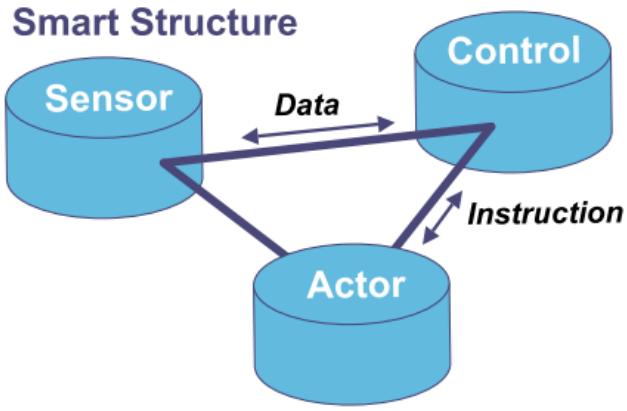


Abb. 3. Smarte Adaptive Materialien verarbeiten Sensor- und Instruktionsströme

- Physische Netzwerke aus Volumenkörpern mit Sensoren und aktuierten Verbindungen zwischen den Körpern → Feder-Masse Systeme!
- Kopplung von physikalischen und rechnenden Knoten

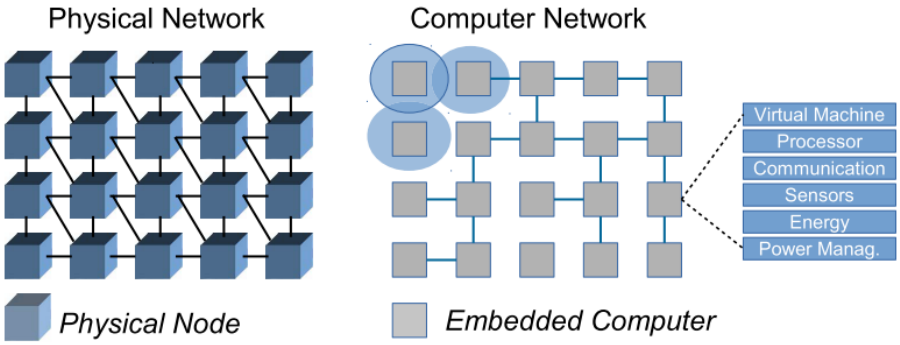


Abb. 4. (Links) Physisches/Meachanisches Netzwerk (Rechts) Berechnungsnetzwerk

3.7. Social & Crowd Sensing

- Social & Crowd Sensing mit Agenten und virtuellen Sensorarchitekturen

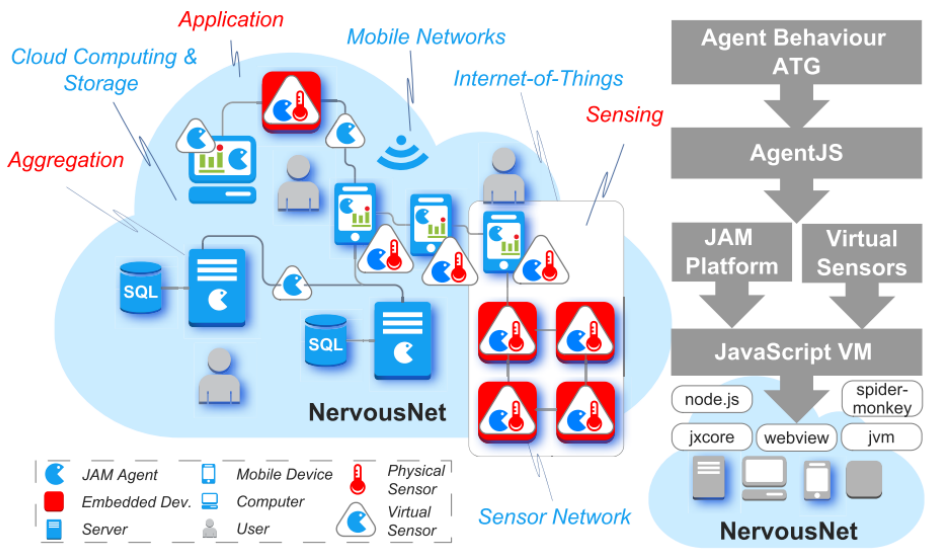
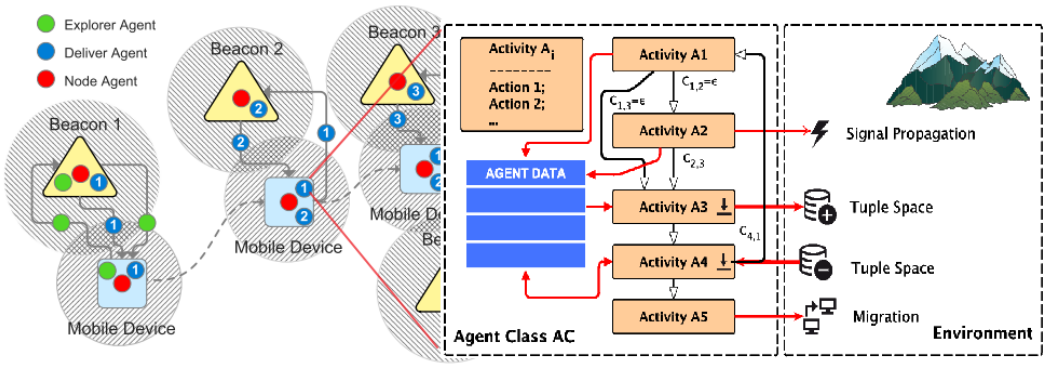


Abb. 5. Vereinheitlichte Datenverarbeitung in Clouds und mobilen Netzwerken

4. Agentenbasierte Verfahren und Plattformen

4.1. Mobile Agenten ⇔ Mobiler Code

- Reaktive mobile Agenten mit Verhaltensmodell: Dynamische Aktivitäts-Übergangsgraphen (**DATG**)
- DATG können von Agenten modifiziert werden → Adaptivität / Rekonfiguration
- Beschreibung der DATG und des Agentenverhaltens durch Metasprache **AAPL**
- Diverse Implementierungen von APPL mit konkreten Programmiersprachen: **AgentScript** (JavaScript), **AgentFORTH**, ..

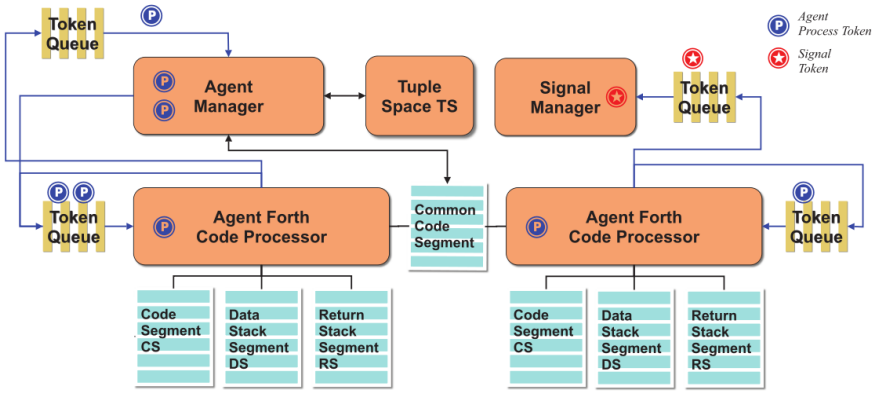


4.2. Plattformen

- Agentenausführungsplattformen: Vom Mikrochip bis zur Cloud

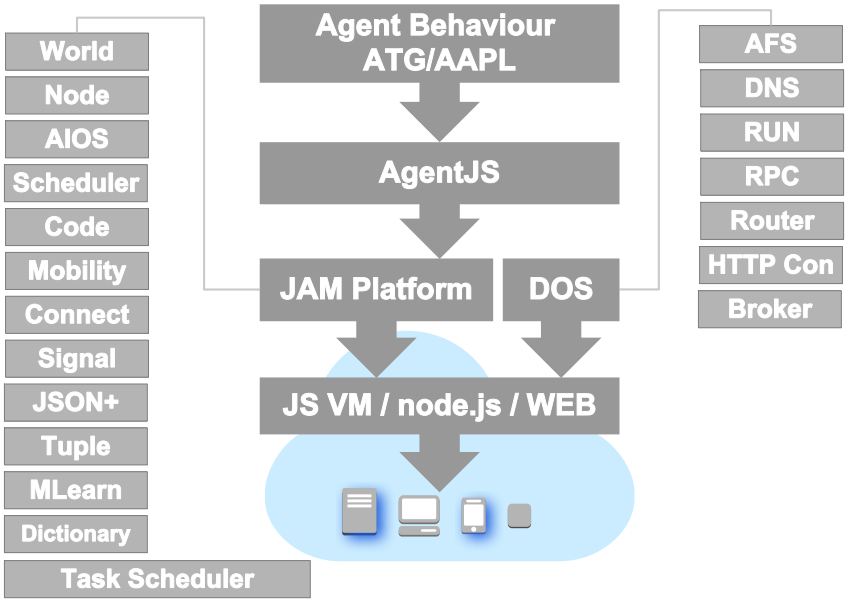
Pipelined AgentFORTH Virtual Machine (PAFVM)

- Parallele Rechnerarchitektur; hoch optimiertes Pipelining- und Schedulingverfahren
- Geeignet für Mikrochipimplementierung (1mm^2)
- Mobilität: Code Frames mit eingebetteten Daten



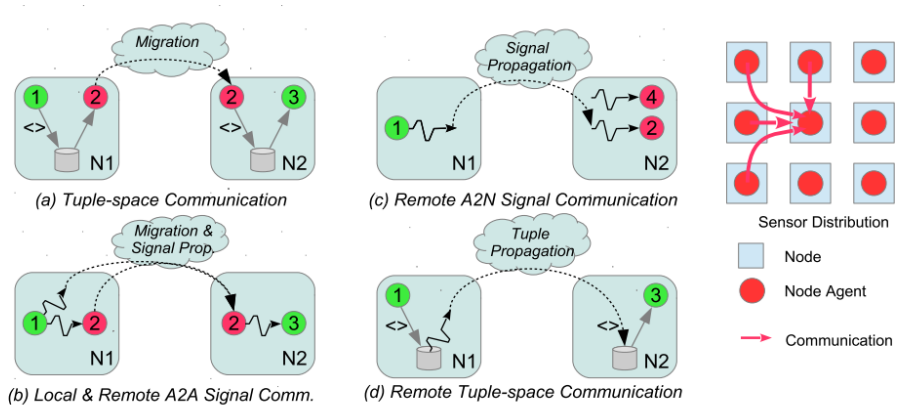
JavaScript Agent Machine (JAM)

- Ausführbar auf jeder JavaScript VM (Browser, Mobiles Gerät, Server)
- Mobilität: JSON+



4.3. Kommunikation & Interaktion

- Einfache und universelle Kommunikations- und Interaktionsschnittstellen für mobile Agenten:
- Tupleräume
- Signale
- Replikation
- Divide & Conquer



5. Self-* Systeme

5.1. Mustererkennung

- Region-of-Interest (ROI) Detektion in verrauschten und gestörten Messsignalen mit selbst-organisierenden Multiagentensytemen
- Divide&Conquer Ansatz mit Mehrheitsentscheid; Kombinierbar mit ML
- Selbstorg. MAS als Softwarearchitektur und adaptiver Algorithmus → Modellfrei!

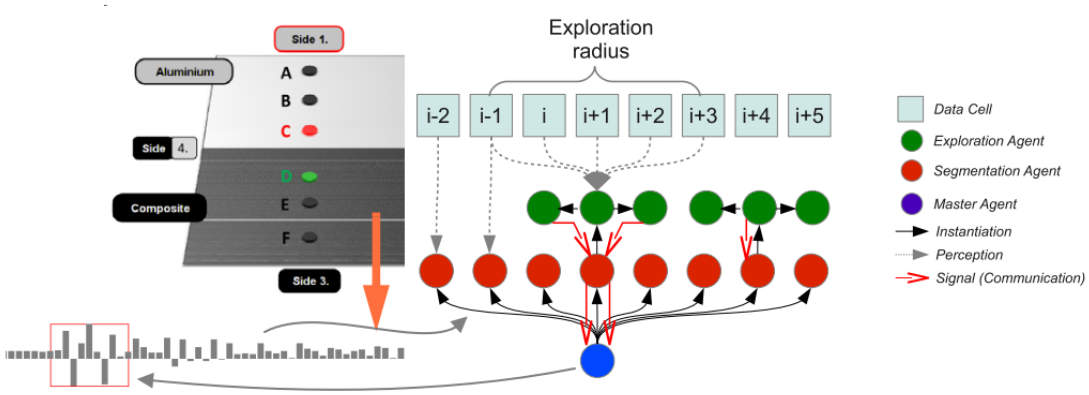


Abb. 6. ROI Extraktion in Ultraschallmesssignalen für die Schadensprüfung in Hybridwerkstoffen

5.2. Energiemanagement durch selbstorganiserte Verhandlung

- Verwendung von MAS für Energiemanagement und Verteilung, z.B. in energieautarken Sensornetzwerken

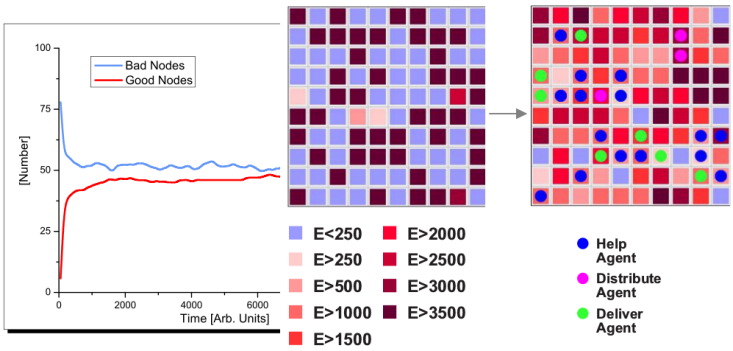


Abb. 7. Verbesserung der Energieverteilung in einem Netzwerk durch Verhandlung von Agenten

5.3. Ereignisbasierte Sensordatenverarbeitung

Automatische Erkennung von sensorischen Ereignissen und ereignisbasierte Verteilung von Sensordaten in großskaligen Sensornetzwerken

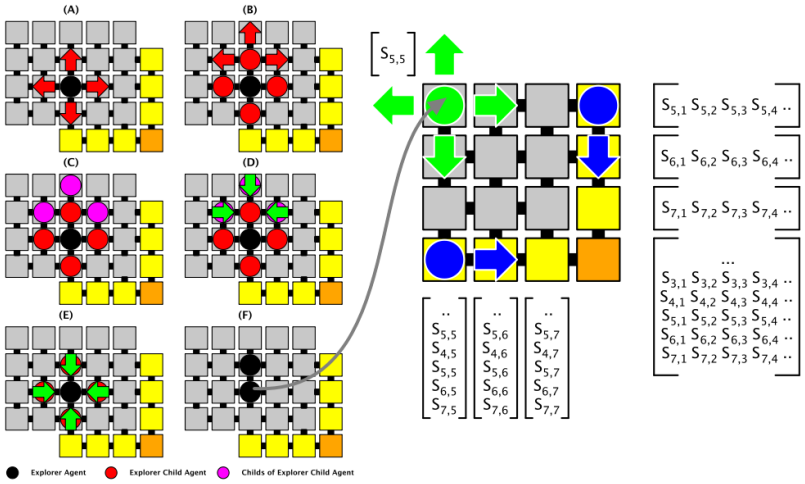


Abb. 8. (Links) Mustererkennung von stimulierten Sensorbereichen (Rechts) Sensordatenverteilung mit Agenten

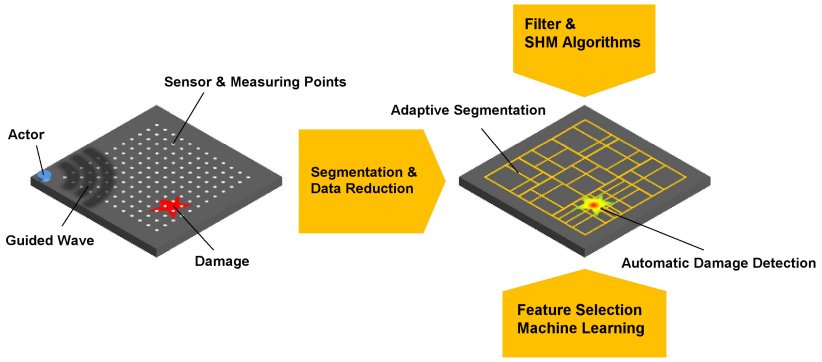
6. Data Mining / Maschinelles Lernen

Algorithmen

- DT Entscheidungsbäume mit Intervallarithmetik (Rauschstabilität)
- ANN Neuronale Netze (mit inneren Ebenen)
- kNN
- Verstärkungslernen
- Hybridverfahren & Methodenfusion
- Clustering

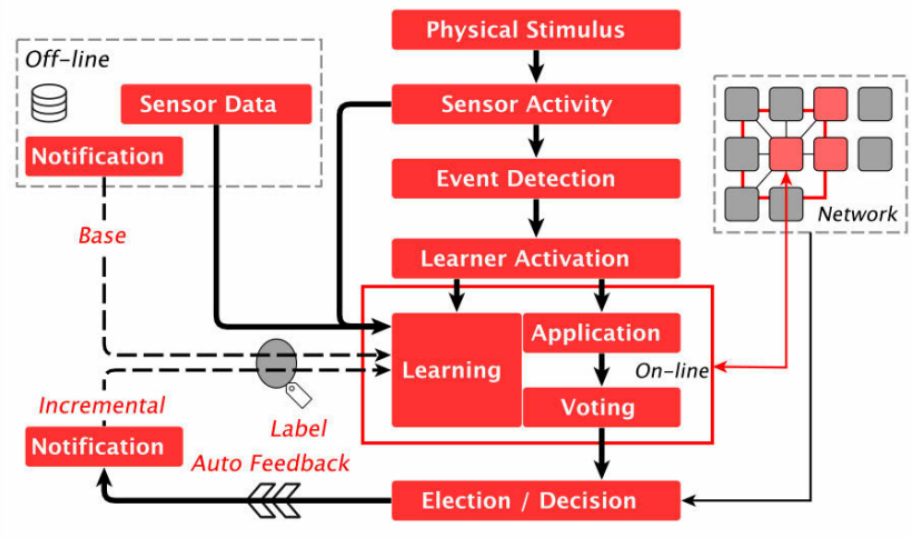
6.1. Strukturüberwachung

- Automatisierung von Structural Health Monitoring (**SHM**) zur Betriebszeit und Zerstörungsfreie Prüfung (**NDT**)
- Merkmalsselektion und Merkmalsextraktion u.a. mit agentenbasierten und Maschinelle Lernverfahren
- Zustandsvorhersage von Lastsituationen und Schäden



6.2. Agentenbasiertes und Verteiltes Lernen

- Verteiltes Lernen: Eine Vielzahl von verteilten Lerninstanzen auf lokalen Daten mit globalen Majoritätsentscheid
- Verteilte lernende Agentensysteme
- Inkrementelle Lernverfahren



6.3. Seismische Netzwerke

- Anwendung von verteilten agentenbasierten Lernen in Seismischen Sensornetzwerken für die Erdbebenklassifikation und Prädiktion

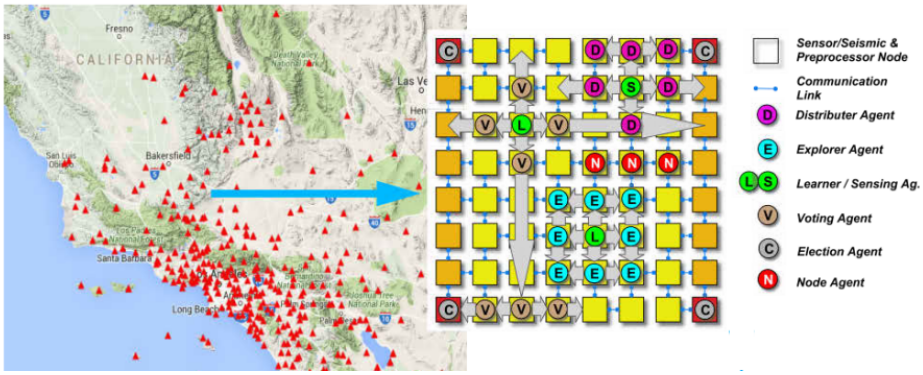


Abb. 9. Seismisches Netzwerk mit JAM Plattform und mobilen lernenden Agenten