Smart Energy Management and Energy Distribution in Decentralized Self-Powered Sensor Networks Using Artificial Intelligence Concepts

Stefan Bosse[1,2], **Frank Kirchner**[1,2]
[1] University of Bremen, Department of Computer Science, Workgroup Robotics
[2] ISIS Sensorial Materials Scientific Centre, Bremen

1. Introduction and Related Work

Sensorial materials equipped with embedded miniaturized smart sensors provide environmental information required for advanced machine and robotics applications. With increasing miniaturization and sensor-actuator density, decentralized self-supplied energy concepts and energy distribution architectures are preferred and required.

Self-powered sensor nodes collect energy from local sources, but can be supplied additionally by external energy sources. Nodes in a sensor network can use communication links to transfer energy, for example, optical links are capable of transferring energy using Laser or LE diodes in conjunction with photo diodes on the destination side, with a data signal modulated on an energy supply signal.

We propose and demonstrate a decentralized sensor network architecture with nodes supplied by 1. energy collected from a local source, and 2. by energy collected from neighbour nodes using smart energy management (SEM). Nodes are arranged in a two-dimensional grid with connections to their four direct neighbours. Each node can store collected energy and distribute energy to neighbour nodes.

Each autonomous node provides communication, data processing, and energy management. There is a focus on single System-On-Chip (SoC) design satisfying low-power and high miniaturization requirements.

Energy management is performed 1. for the control of local energy consumption, and 2. for collection and distribution of energy by using the data links to transfer energy.

Typically, energy management is performed by a central controller in which a program is implemented **[5]**, with limited fault robustness and the requirement of a well-known environment world model for energy sources, sinks, and storage. Energy management in a network envolves the transfer of energy.

The loss of energy ε (in the range between 0 and 1) at each node occurring each time when "energy" is routed along different nodes from a source to a destination node (assuming N intermediate nodes) reduces overall efficiency dramatically in the order of $\eta=\varepsilon^N$.

By using electrical connections, only negligible loss of energy can be expected in a distributed network, in contrast to optical and radio wave connections which have significant loss in the order of $\varepsilon \cong$10-30% per node. Additionally, in the latter case there is no physical interaction between a source and a sink node requesting energy, thus requiring active management (routing).

To overcome these limitations and to increase operational robustness, this work proposes smart energy management performed by using concepts from artificial intelligence. Initially, the sensor network is a distributed group of independent computing nodes. Interaction between nodes is required to manage and distribute information and energy. One common interaction model is the mobile agent.

Different kinds of agents with different behaviours are used to negotiate energy demands and energy distribution and to implement group communication. A multi-agent system is a decentralized and self-organizing approach for data processing in a distributed system like a sensor network.

Recent work shows the benefit and suitability of multi-agent systems used for energy management **[5]**.

Section **2.** describes the communication architecture and some aspects of the technical implementation required for a communication link which is capable of transferring and receiving data and energy. Section **3.** gives a short introduction to the multi-agent approach and the agent implementation used for smart energy management, targeting single microchip technologies (SoC designs). Section **4.** finally discusses some results retrieved from simulation, showing the benefits of smart energy management using agents.

2. Communication, Data and Energy Transport

Nodes, arranged for example in a two-dimensional mesh-network (see figure **1**), can exchange data with their neighbours by using serial communication links. There are different kinds of physical transmission technologies: electrical, optical, and radio-wave based.

In contrast to electrical interconnect technologies, optical and radio-wave technologies have the disadvantage of lower efficiency ε. This is negligible for the exchange of information, but significant for the distribution and exchange of energy required for the supply of nodes. Optical communication has clear advantages such as extremely small and light-weight hardware, ultra-low power consumption, and the ability to optimally focus and match the beam to the transmission medium (optical fibre) **[4]**.

Sharing of one interconnect medium for both data communication and energy transfer significantly reduces node and network resources and complexity, a prerequisite for a high degree of miniaturization required in high-density sensor networks embedded in sensorial materials. Point-to-point connections and mesh-network topologies are preferred in high-density networks because they allow good scalability (and maximal path length) in the order of *O(log N)*, with N as the number of nodes.

Figure **1** shows the main building blocks of a sensor node, the proposed technical implementation of the optical serial interconnect modules, and the local energy management module collecting energy from a local source, for example a thermo-electric generator, and energy retrieved from the optical communication receiver modules.

The data processing system can use the communication unit to transfer data (D) and superposed energy (E) pulses using a light-emitting or laser diode. The diode current, driven by a differential-output sum amplifier, and the pulse duration time determine the amount of energy to be transferred. The data pulses have a fixed intensity several orders lower than the adjustable energy pulses. On the receiver side, the incoming light is converted into an electrical current by using a photo diode. The data part is separated by a high-pass filter, the electrical energy is stored by the harvester module. Information and energy is encapsulated in messages routed in the network from a source to a destination node using a simple delta-routing protocol. The routing algorithm is explained in figure **2**. An alternative advanced smart routing protocol, which allows incomplete mesh-networks and compensate link failures by using different routing rules, is described in **[3]**.

Fig. 1. **Network topology (left) and sender and receiver blocks (right) used for data and energy transmission between neighbour nodes arranged in the network. Each**

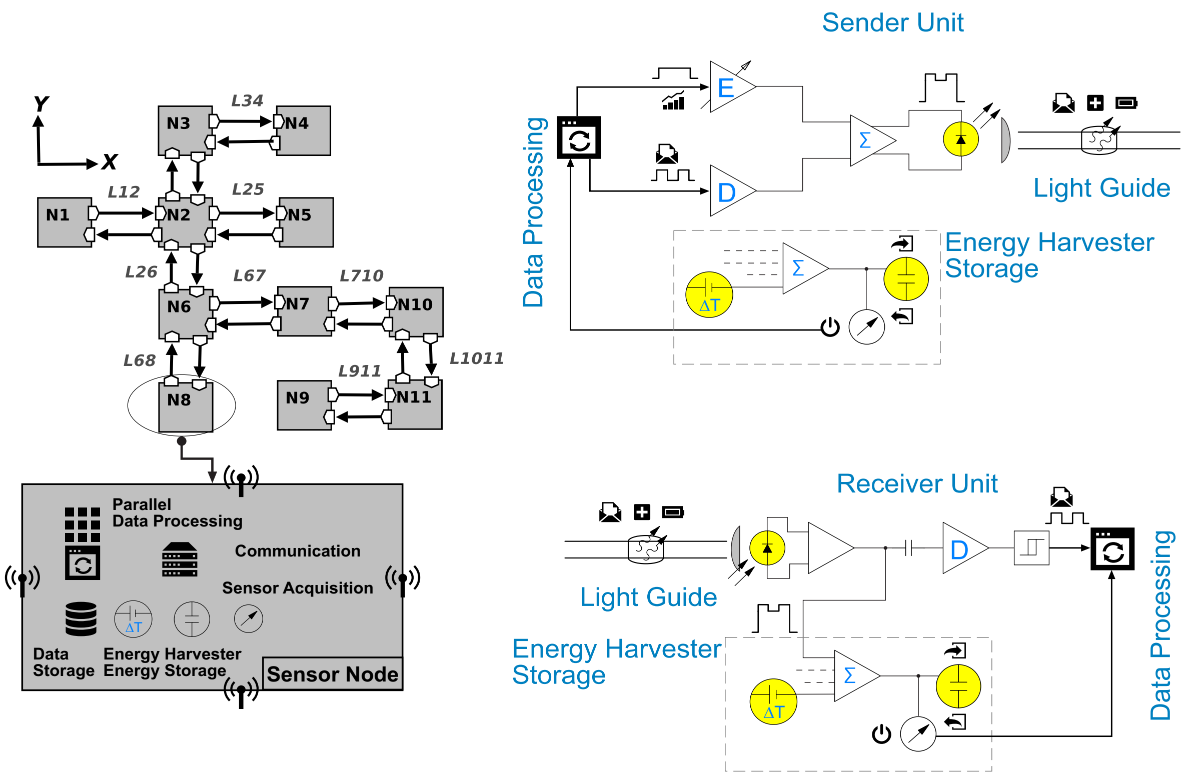
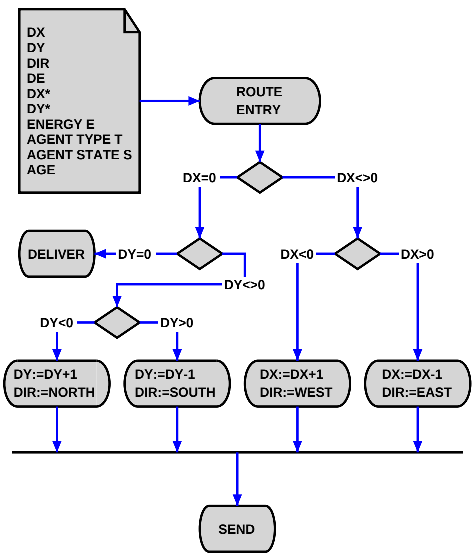


Fig. 2. **UML activity diagram of the simplified message routing algorithm. Simple delta routing rules are applied to messages until the propagation vector is Δ=(DX,DY)=(0,0).**



3. Multi-Agent Interaction Model and Implementation

Initially, the sensor network is a collection of independent computing nodes. Interaction between nodes is required to manage and distribute information and energy. One common interaction model is the mobile agent. An agent is capable of autonomous action in an environment with the goal to meet its delegated objectives. An agent is a data processing system, a program executed on a computer system, that is situated in this environment **[1]**.

Having the technical abilities explained in the previous section, it is possible to use active messaging to transfer energy from good nodes having enough energy towards bad nodes, requiring energy. An agent can be sent by a bad node to explore and exploit the near neighbourhood. The agent examines sensor nodes during path travel or passing a region of interest (perception) and decides to send agents holding additional energy back to the original requesting node (action). Additionally, a sensor node is represented by a node agent, too. The node and the energy management agents must negotiate the energy request.

An agent consists of a state, holding data variables and the control state, and a reasoning engine, implementing behaviours and actions. Table **1** explains different agent behaviours required for smart energy management.

A node having an energy level below a threshold can send a help agent with a delta-distance vector specifying the region of interest in a randomly chosen direction. The help agent hops from one sensor node to the next until the actual delta-vector is zero (see figure **2**). If there is a good node found, with local energy above a specified threshold, the help agent persists on this node and tries to send periodically deliver agents transferring additional energy to the originator node. An additional behaviour, help-on-way, changes the deliver agent into a exploration agent, too. Such a modified agent examines the energy level of nodes along the path to the destination. If a bad node was found, the energy carried with the agent is delivered to this node, instead to the original destination node.

The state of an agent is completely kept in the message transferred in the network, but not the functional behaviour. Figure **3** shows the execution environment for the energy management agents. There is a message module implementing delta-distance routing, and several finite-state-machines implementing the agent behaviour and providing virtual machines able to process incoming agents. All parts are mappable to digital logic on RTL and SoC system architecture.

Tab. 1. Agents with different behaviours used to manage and distribute energy (ROI: region of interest).

Agent Type	Behaviour
`Request`	*Point-to-point agent*: this agent requests energy from a specific destination node, returned with a `Reply` agent.
`Reply`	*Point-to-point agent*: Reply agent created by a `Request` agent, which has reached its destination node. This agent carries energy from one node to another.
`Help`	*ROI agent*: this agent explores a path starting with an initial direction and searches a good node having enough energy to satisfy the energy request from a bad node. This agent resides on the final good node for a couple of times and creates multiple deliver agents periodically in dependence of the energy state of the current node.
`Deliver`	*Path agent*: this agent carries energy from a good node to a bad node (response to `Help` agent). Depending on selected sub-behaviour (`HELPONWAY`), this agent can supply bad nodes first, found on the back path to the original requesting node.
`Distribute`	*ROI agent*: this agent carries energy from and is instantiated on a good node and explores a path starting with an initial direction and searches a bad node supplying it with the energy.

Fig. 3. Sensor node building blocks providing mobility and processing of multi-agent systems: parallel agent virtual machines, agent-processing scheduler, communication, and data processing.

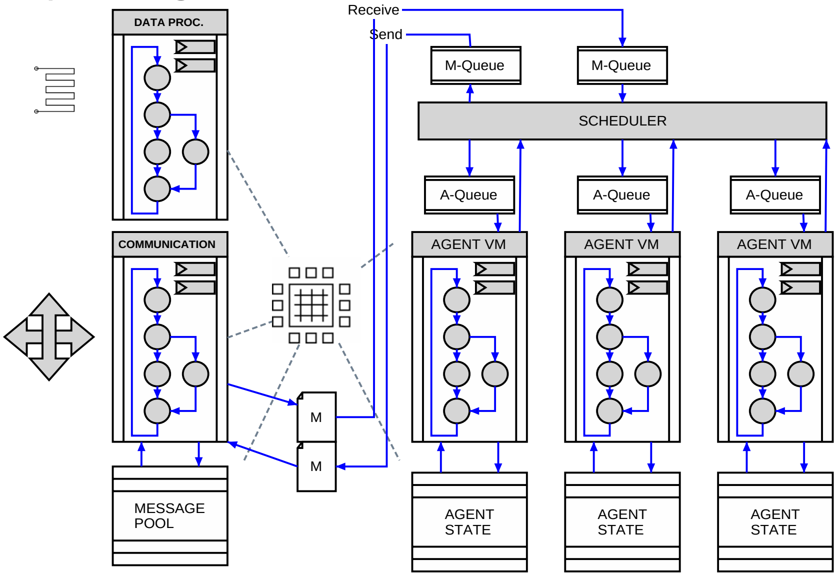


Fig. 4. **UML activity diagram for the energy management agent. This super agent contains request, reply, help, deliver, and distribute agents implementing different behaviours.**

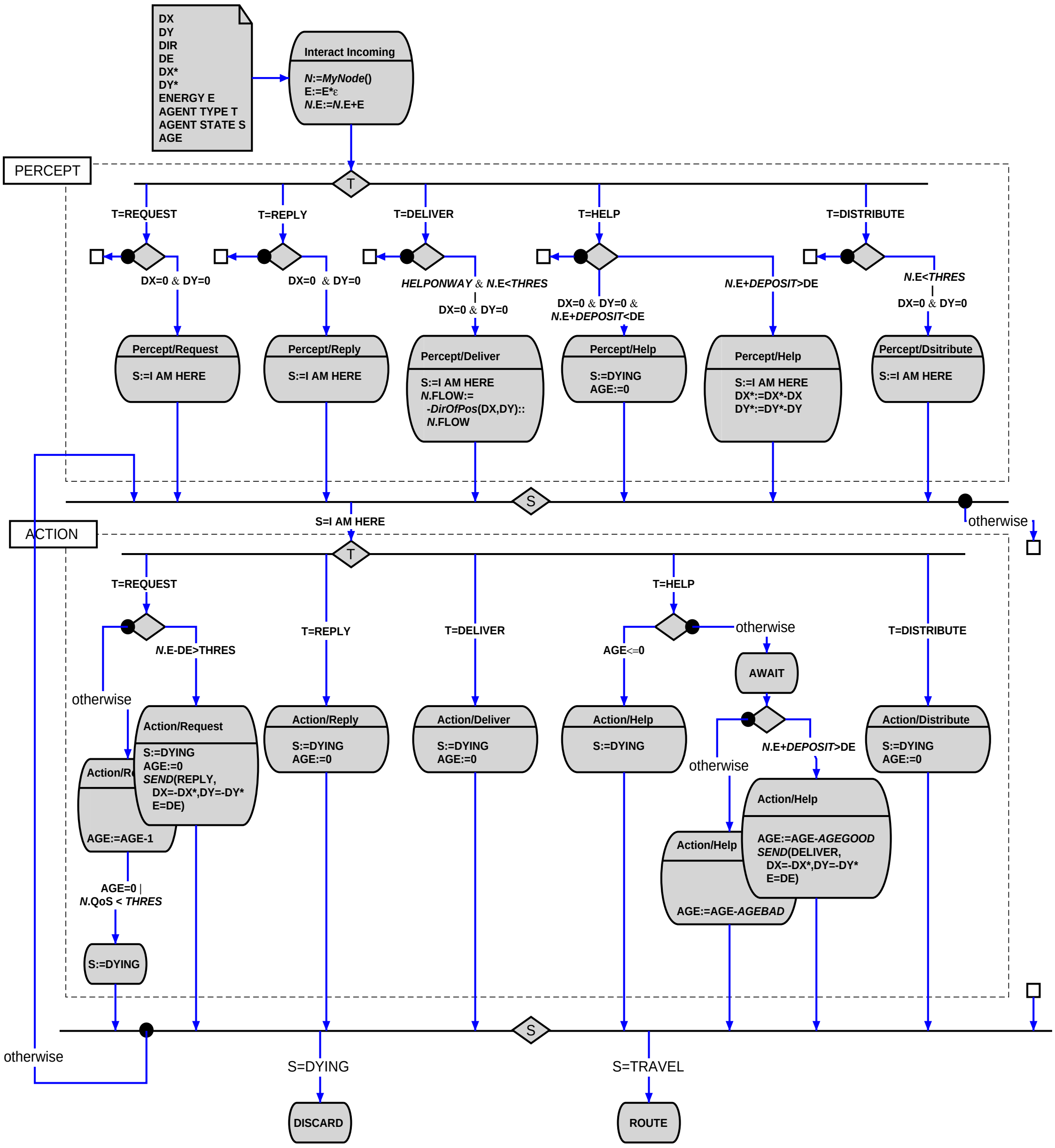


Figure **4** shows the activity diagram of the reasoning engine, implementing all types of agents, forming a super agent.

The delta-distance or region of interest vector (DX,DY) is modified on each message routing, the (DX*,DY*) vector holds an unmodified copy, which is used for replies, the DE entry specifies the requested energy, and the ENERGY entry reflects the actual energy carried with the message (without the contribution of the data part itself). This entry is altered each time a message is routed, respecting the transmission effciency ε.

4. Analysis, Experimental Results, and Conclusions

A first proof of concept and experimental results were achieved by using a multi-agent simulation framework (SeSAM) **[2]**. The simulation test-bed consists of a network with N=100 nodes arranged in a 10 by 10 matrix. Each node can communicate with up to four neighbours in the directions `DIR`= `{North,South,West,East}`.

Each node N_i periodically collects (store) energy from a local source having a stochastically distributed energy spectrum in the range $[0,E_{i,max}]$. Monte Carlo simulation is used to specify each $E_{i,max}$ before a simulation run. Data processing, interaction with the environment (e.g. sensor acquisition), and agents consume energy, which reduces the energy deposit E_i.

Each sensor node is modelled with an agent, too. Energy management agents and sensor node agents negotiate energy demands and communicate by using globally shared variables. If the energy deposit of a node is below a threshold $E_i<E_{low}$ (called bad node), help agents are sent out, if $E_i>E_{high}>E_{good}$ then distribute agents are used to distribute energy to surrounding bad nodes. If $E_i>E_{good}>E_{low}$ then the node is fully functional (called good node).

Assuming a specific stochastic spatial configuration $\{E_{i,max}\}$, simulation results in figures **5** and **6** show the benefit of energy management using help and distribute agents. Without energy management, there are about 50% bad nodes (blue rectangles) never reaching an energy level above a critical threshold. With agents, the spatial energy distribution is more regular, and the fraction of bad nodes is below 10% all the time. Figure **8** compares the number of bad nodes resulting from different combinations of agents and agent behaviours. Using additional distribute agents results in a decrease of 30% relative to the case using only help agents, but absolutely the benefit is below 5 % and is therefore negligible. Moreover, the fraction of all nodes with mean up-time below a critical threshold (10%) is always below 5%, shown in figure **9**. Figure **7** shows the temporal progress of total system energy in dependency on different energy management agents. Due to the high loss of energy transfer between nodes (here 20%), the total energy efficiency is dramatically decreased compared with the case without management, and distribute agents reduce the total system efficiency again about

50%.

Fig. 5. Simulation results for a network of sensor nodes arranged in 10 by 10 matrix topology without any distributed energy management. Each node collects energy from a local source (the maximal energy gain is random and derived from Monte Carlo simulation). Shown are the spatial energy distribution after 10000 time units (right) and the temporal population for bad and good nodes (left).

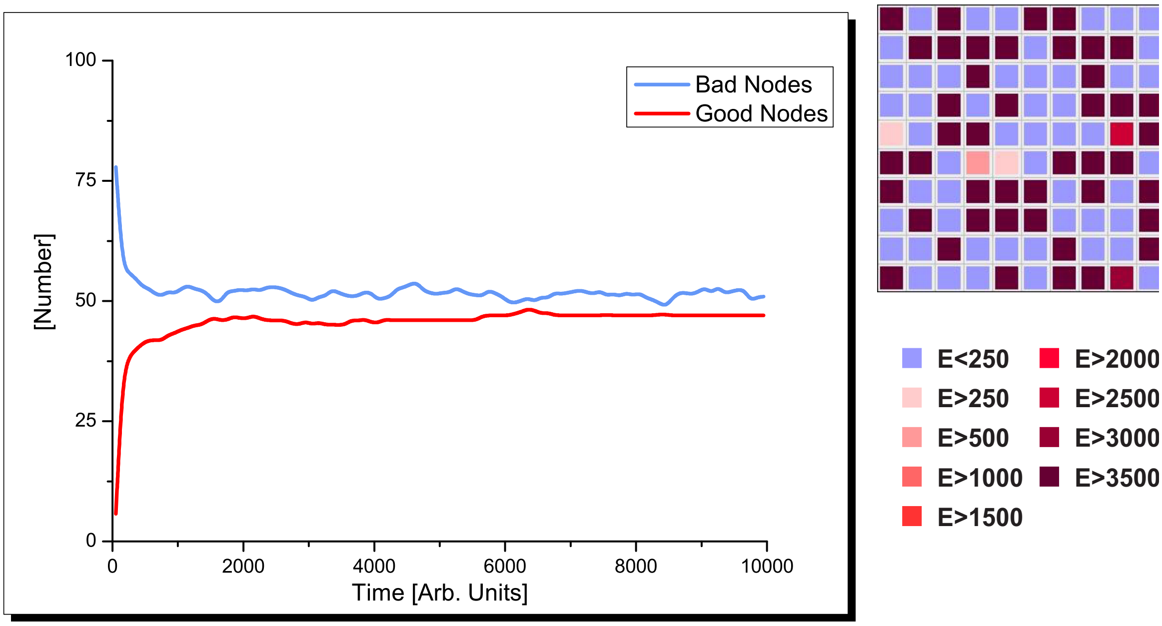


Fig. 6. Simulation results with energy management. Help, deliver, and distribute agents are used to compensate low-energy nodes. Shown are the spatial energy distribution after 10000 time units (right, with some agents) and the temporal population for bad and good nodes and agents (left).

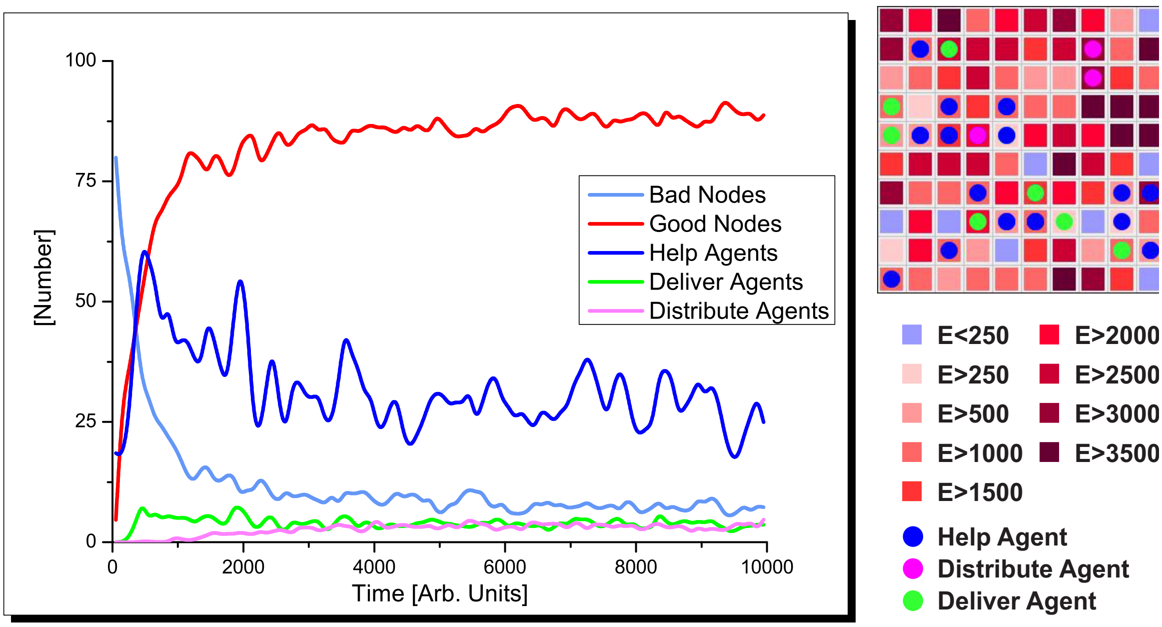


Fig. 7. **Comparison of total sensor network energy for different energy management agent systems.**

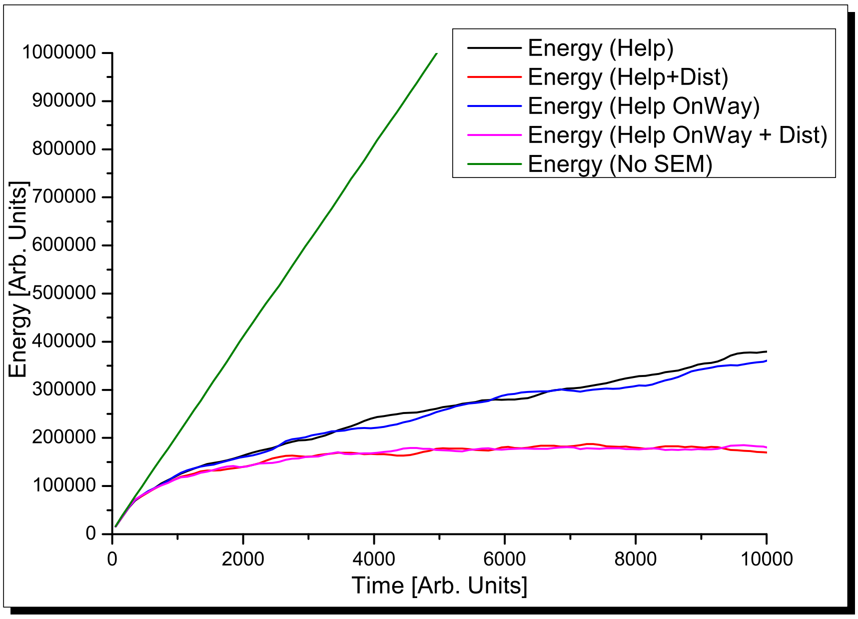
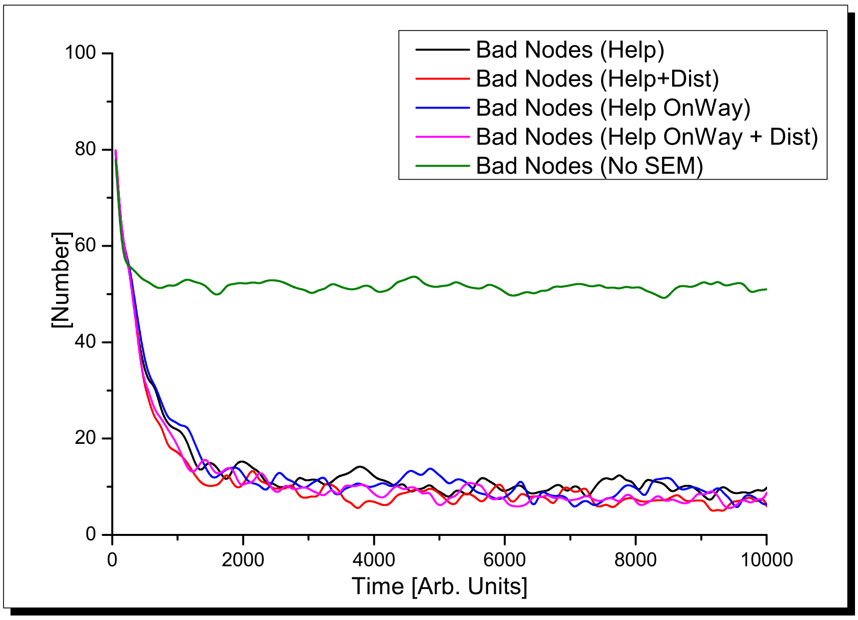
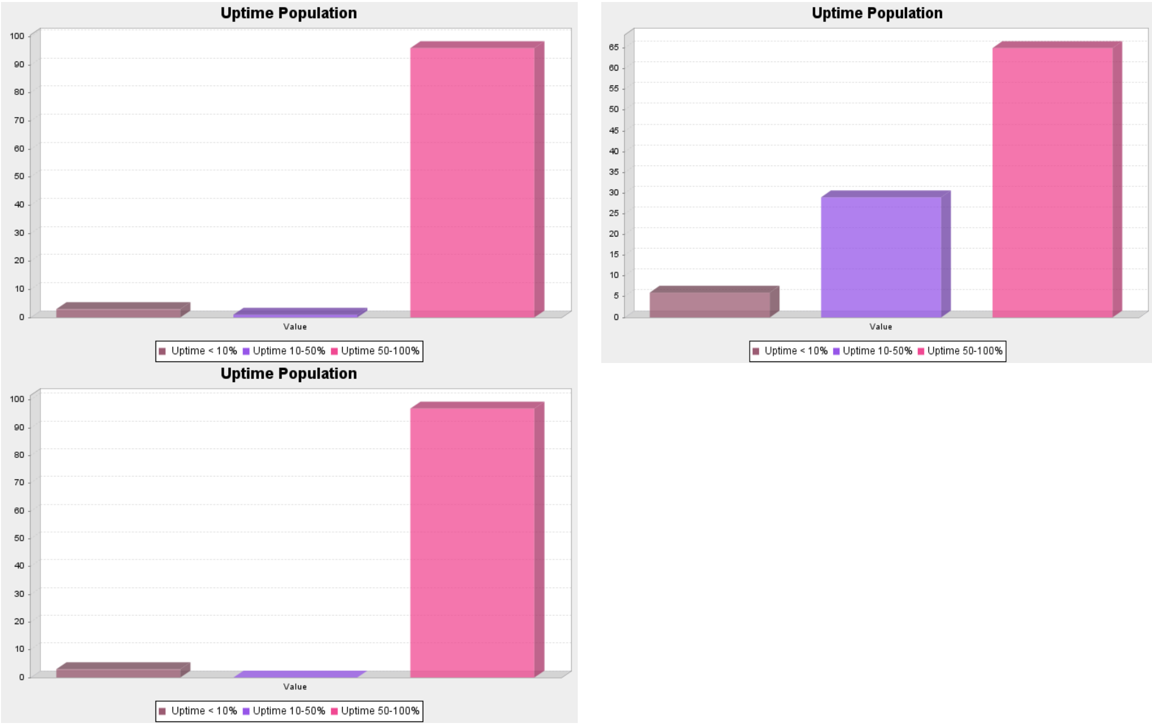


Fig. 8. **Comparison of bad node population for different energy management agent systems.**



To summarize, help agents with simple exploration and exploitation behaviours are suitable to meet the goal of a regular energy distribution and a significant reduction of bad nodes not able to contribute sensor information, but additional distribute agents create no significant benefit. The multi-agent implementation offers a distributed management service rather than a centralized approach commonly used. The simple agent behaviours can be easily implemented in digital logic hardware.

Fig. 9. Comparison of up-time distribution with (left) and without smart energy management (right). Each bar shows a up-time range relative to total running time, and the number of nodes (pay attention to different scales). The upper left chart shows the simulation using help-on-way and distribute agents, and for the lower left chart only help-on-way agents are used.



5. References

[1] M. Wooldridge, *An Introduction to MultiAgent Systems*, Wiley

[2] F. Klügel, *The Multi-Agent Simulation Environment SeSAm*, In: H. Kleine Büning (Ed.): Proceedings of Workshop "Simulation in Knowledge-based Systems", Paderborn, April 1998

[3] S. Bosse, D. Lehmhus, *Smart Communication in a Wired Sensor- and Actuator-Network of a Modular Robot Actuator System using a Hop-Protocol with Delta-Routing*, Proceedings of Smart Systems Integration conference, Como, Italy, 23-24.3.2010 (2010)

[4] D. Kedar, S. Arnon, *Optical wireless communication in distributed sensor networks*, SPIE Newsroom, 2006

[5] J. Lagorse, D. Paire, A. Miraoui, A multi-agent system for energy management of distributed power sources, J. of Renewable Energy, Vol. 35, Issue 1