Dependable Sensor Networks for Smart Cities
– Research Position Statement –

Oliver Theel

System Software and Distributed Systems
Department of Computer Science
University of Oldenburg, Germany

http://www.uol.de/svs

September 25, 2019

Outline

- Group's General Research Interests
- Smart Cities and Sensor Networks
- Sensor Networks and Distributed Systems
- Dependable Sensor Networks
- Replication of Data and Services
- Self-Stabilization
- Region Adherence

System Software and Distributed System Group

General Research Interests

- Distributed Systems
- Distributed Algorithms
- Dependability, Fault Tolerance
- Replication, Self-Stabilization, Region Adherence
- Dependability Measures, Performance, Energy Efficiency
- Scalability, Dynamics, Graceful Degradation, Consistency Notions
- Sensors for Environmental Phenomena

Smart Cities and Sensor Networks

Smart Cities Characteristics

Smart Cities use information technology to

1) make efficient use of resources to support strong & healthy economic/social/cultural development,
2) to foster innovative processes, collective intelligence, and citizen participation,
3) learn, adapt, and innovate, and thus are able to respond more efficiently and promptly to changing circumstances by improving the intelligence of the city.

(picture taken from smartcities.ieee.org)

Smart Cities and Sensor Networks

Smart Cities Characteristics

Smart Cities use information technology to

1) make efficient use of resources to support strong & healthy economic/social/cultural development,
2) to foster innovative processes, collective intelligence, and citizen participation,
3) learn, adapt, and innovate, and thus are able to respond more efficiently and promptly to changing circumstances by improving the intelligence of the city.

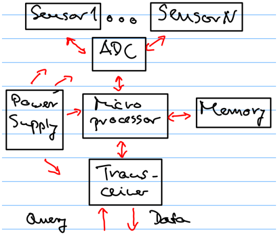
→ information technology plays an important role in the SC context

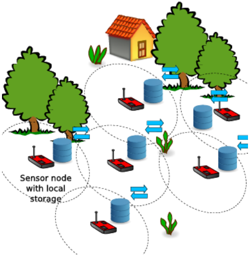
(picture taken from smartcities.ieee.org)

Sensor Network Characteristics

Sensor Networks

- consist of relatively cheap Sensor Nodes (SNs) being often matchbox-sized computers
- SNs are very resource-limited in terms of computing power, memory, and energy if no external power supply is attached
- SNs have mission-related sensors (and actors) attached
- communicate often via wireless communication standards like ZigBee with each other
- can be easily deployed
- may communicate through gateway nodes to LANs and WANs



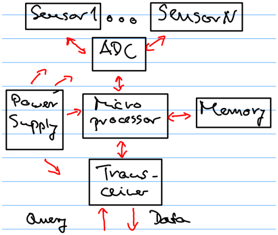


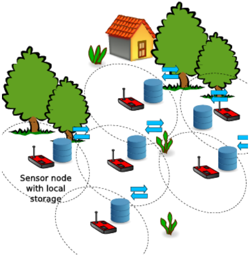
Sensor Network Characteristics

Sensor Networks

- consist of relatively cheap Sensor Nodes (SNs) being often matchbox-sized computers
- SNs are very resource-limited in terms of computing power, memory, and energy if no external power supply is attached
- SNs have mission-related sensors (and actors) attached
- communicate often via wireless communication standards like ZigBee with each other
- can be easily deployed
- may communicate through gateway nodes to LANs and WANs

→ in the following, Sensor Networks are referred to as Wireless Sensor Networks (WSNs)





Smart Cities and Sensor Networks

WSN as Part of Smart Cities' Information Technology

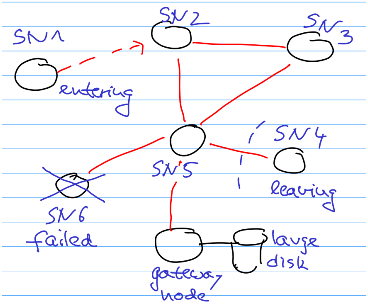
For example, in a Smart City, WSNs can be used to

1) monitor e.g.
   - air quality
   - fresh water and sewage quality
   - fresh water consumption and sewage production
   - river and sewage system levels etc.

   through connected, integrated SNs,
2) allowing citizens e.g.
   - inspect data collected by these SNs and
   - to set up and add their own sensors and SNs to the network,
3) thereby enabling e.g.
   - precise localization of and
   - timely response to

   critical situations.

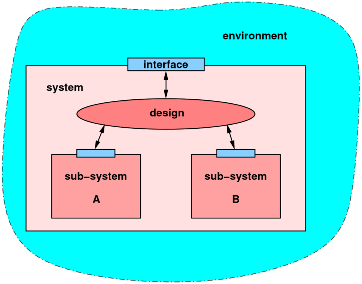
Sensor Networks and Distributed Systems



- conceptually, WSNs are distributed systems
- in the SC context, often, they
  - are dynamic: SNs enter or leave the WSN over time
  - must scale: # of SNs of WSN may strongly increase over time
  - must be dependable: WSN must achieve its mission despite some failed sub-systems

Dependable Sensor Networks

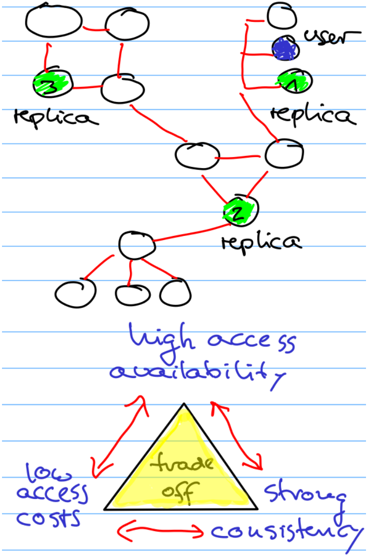
- dependable distributed system can be realized through fault tolerance
- **Fault Tolerance**
  - construction of highly dependable systems based on failure-prone sub-systems
  - some sub-systems may fail but system is not rendered useless or incorrect → "tolerates failures to some extend"
- **Fault Tolerance Classes**
  - fail-stop fault tolerance
  - masking fault tolerance
  - non-masking fault tolerance
- **Fault Tolerance Concepts**
  - replication
  - self-stabilization
  - region adherence



	live	not live
safe	Masking Fault–Tolerant	Fail–Stop Fault–Tolerant
not safe	Non–Masking Fault–Tolerant	Not Fault–Tolerant

Replication of Data and Services

- create copies of data → replicas
- access operations are performed on a subset of replicas thereby honoring a given consistency notion → e.g. 1SR
- a replication strategy (RS) defines access operations, consistency notion, availabilities and costs



General Aim

- guarantee high availability and at the same time low costs of data access operations

Problems

- there is no single best RS
- subtle relation availability ↔ costs
- fault model and workload key to good solutions

Replication of Data and Services

Aim in WSN context

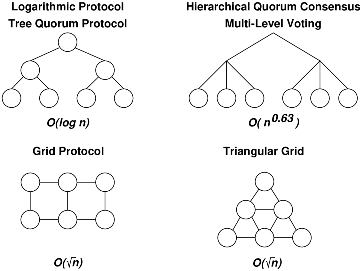
- store and provide sensed data

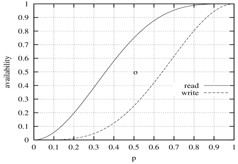
Problems in WSN context

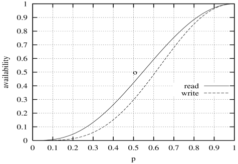
- access costs must be low since energy is restricted
- at the same time: data must be highly available since data might be crucial
- WSN: dynamic, large, subject to faults

Our Contributions

- easy replacement of obsolete replication strategies if WSN has changed substantially $\rightarrow$ replication framework
- synthesis of mission-optimized replication strategies
  $\rightarrow$ by genetic algorithm (GA)

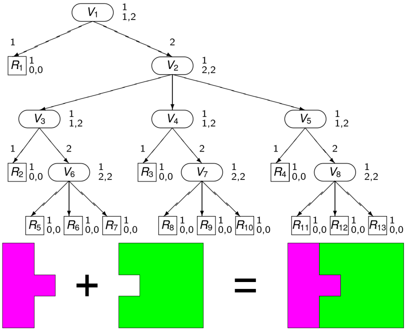


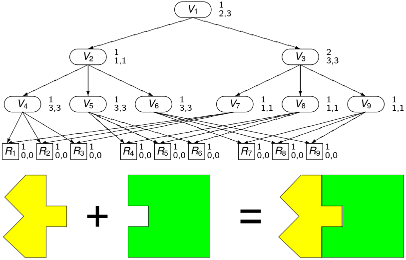




Replication Framework Approach

- replication strategy is modeled as a voting structure (VS)
- general mechanism interprets VS at run-time leading to the RS's behavior in terms of access availabilities and costs
- RS A can be replaced by RS B simply by changing the VS
- no coding, only reconfiguration, possible even at run-time

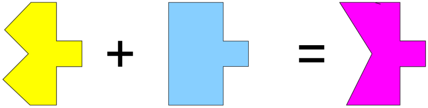


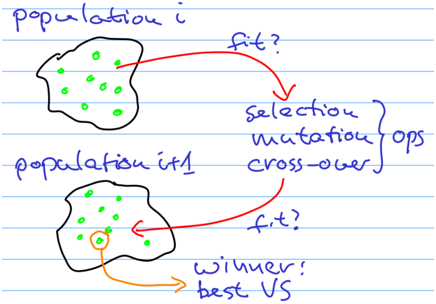


Genetic Algorithm Approach

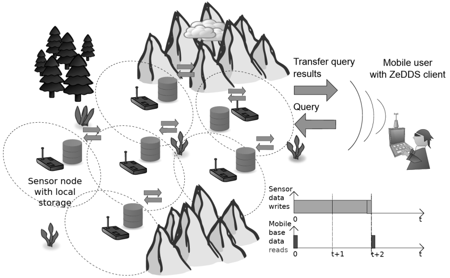
- specify desired properties of RS via constraint system
  $\rightarrow$ fitness function, fitness value
- known replication strategies are represented as VSs
- VSs are the individuals of a GA population

1) select initial population
2) apply genetic operators (selection, mutation, cross-overs) and fitness check in order to create new generation
3) repeat Step 2) for some time or until desired fitness is met
4) use VS (= RS) with highest fitness obtained





Example of a Sensor Network with Replicated Data [1]



- WSN setting
  - sensors collect data
  - data sink is mobile and often beyond communication range
  - measured data should not get lost in the meantime , e.g., due to failed SNs
  - data is stored within the WSN itself
  - energy within WSN is limited
- ZeDDS approach
  - stands for "dependable and energy-efficient data management in WSNs"
  - data is replicated according to some RSs → trade-off availability vs. energy costs
- ZeDDS approach (Cont.)
  - RSs are represented by VS
  - RSs can be changed at run-time
  - sink can harvest data from any SN when in communication range
  - harvested data can subsequently be deleted

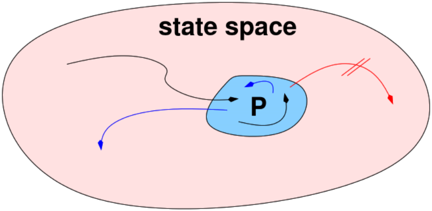
Self-Stabilization

A system A is self-stabilizing wrt. a set of states P if

- (Convergence): regardless of its initial state, A reaches a state in P in finite steps and
- (Closure): once in state in P, A does not subsequently leave P.

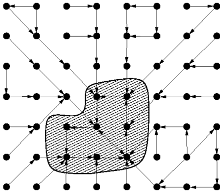
General Aim

- guarantee autonomous, uninitialized functioning of a distributed system despite any transient faults



General Problems

- self-stabilizing systems (SSS) are complicated to design
- SSS properties are likely to be destroyed under composition
- SSS are complicated to formally prove correct



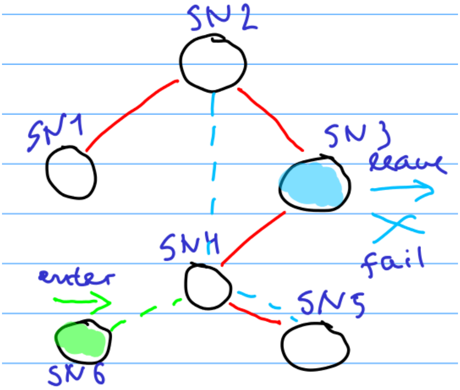
Self-Stabilization

Aim in WSN context

- provide high availability of a WSN's communication backbone when SNs fail or leave or enter the system

Problems

- SSS are normally only analyzed in terms of convergence after the last fault
  → convergence yes?/no?
- but here, due to network dynamics, SSS must be analyzed under ongoing fault assumptions
- which SS spanning tree algorithm provides high connectivity under these assumptions?

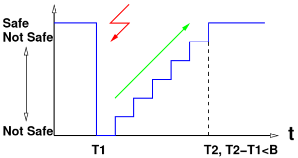


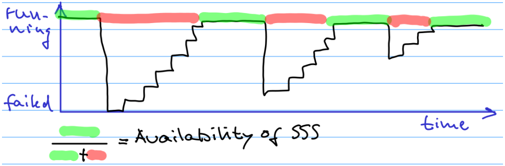
Our Contributions

- notion of availability of SSS
- availability optimization of a SSS

Availability of SSS

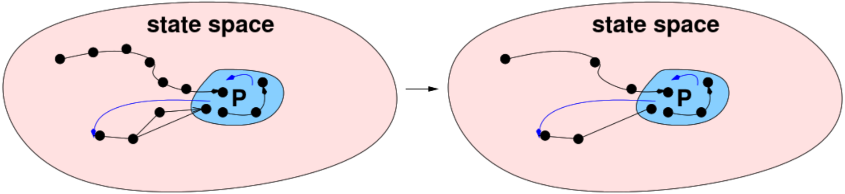
- when in P, a SSS does useful work $\rightarrow$ system is available
- when not in P, some illegal behavior may be observed $\rightarrow$ system has failed
- (limiting) availability of a SSS is the probability that SSS is in P at an arbitrary point in time
- when two SSSs solve the same problem, then the system with the higher availability might be the better choice





Availability Optimization Approach

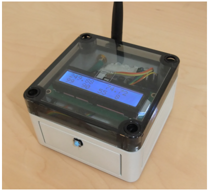
- analyze SSS using probabilistic model checking
- analyze resulting "fault tree"
- modify SSS: shorten existing paths
- modify SSS: introduce short-cut paths
- modify SSS: eliminate too long paths

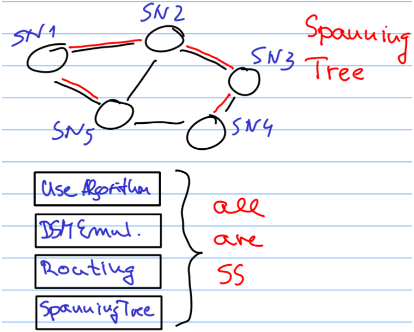


- less "time spent outside P" $\rightarrow$ availability is increased

Example of a Self-Stabilizing Sensor Network [2]

- SNs equipped with
  - brightness sensor
  - push button
  - LC Display
- each SN
  - measures brightness when button pressed
  - calculates average brightness of all current brightness values of SNs and
  - displays it
- SNs may fail, join, or leave the system autonomously
- WSN self-stabilizes to states where all SNs display the average brightness





Region Adherence

A region-adherent system (RAS)

- gracefully degrades the service quality provided by the system per fault up to some maximal number of faults and
- degradation is upper-bounded per fault

Region Adherence

A region-adherent system (RAS)

- gracefully degrades the service quality provided by the system per fault up to some maximal number of faults and
- degradation is upper-bounded per fault

Formal Definition

We assume a system with configurations C, initial configurations C_0 and algorithm $\mathcal{A}$ under fault model $\mathcal{F}$. Let $g : C \mapsto [0, 1]$ be a function stating the service quality of the system and let f be a natural number.
$r : \{0, \ldots, f\} \mapsto [0, 1)$ is a non-decreasing function with $r(0) = 0$ and $r(f) < 1$. Algorithm $\mathcal{A}$ is called *f-region-adherent wrt. g, r, and $\mathcal{F}$*, if and only if for all reachable configurations $c \in C$, all initial configurations $c_0 \in C_0$, and all executions $\gamma = c_0 \cdots c$ ending in c the following holds:

$$g(c) \geq 1 - r\left(\#_{\mathcal{F}\setminus\mathcal{A}}(\gamma)\right), \qquad (1)$$

where $\#_{\mathcal{F}\setminus\mathcal{A}}(\gamma)$ represents the number of fault steps of execution γ. A system executing an f-region-adherent algorithm is also called f-region-adherent.

Region Adherence

A region-adherent system (RAS)

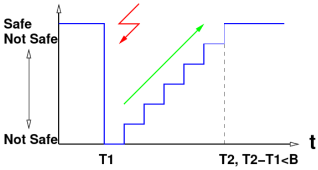
- gracefully degrades the service quality provided by the system per fault up to some maximal number of faults and
- degradation is upper-bounded per fault

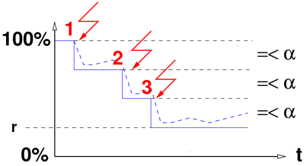
General Aim

- guarantee a priori-known minimal service quality even after some number f of faults
- f-region-adherent

General Problems

- region adherent systems are complicated to design
- RAS properties are likely to be destroyed under composition
- RAS are complicated to formally prove correct





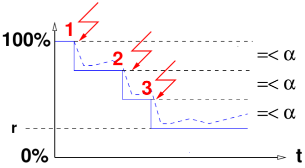
Region Adherence

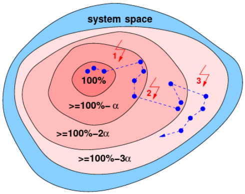
A region-adherent system (RAS)

- gracefully degrades the service quality provided by the system per fault up to some maximal number of faults and
- degradation is upper-bounded per fault

General Aim

- guarantee a priori-known minimal service quality even after some number f of faults
- f-region-adherent





General Problems

- region adherent systems are complicated to design
- RAS properties are likely to be destroyed under composition
- RAS are complicated to formally prove correct

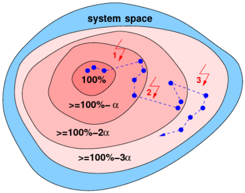
Region Adherence

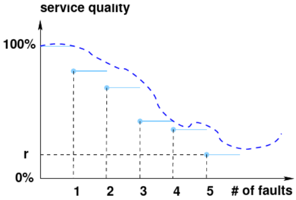
A region-adherent system (RAS)

- gracefully degrades the service quality provided by the system per fault up to some maximal number of faults and
- degradation is upper-bounded per fault

General Aim

- guarantee a priori-known minimal service quality even after some number f of faults
- f-region-adherent





General Problems

- region adherent systems are complicated to design
- RAS properties are likely to be destroyed under composition
- RAS are complicated to formally prove correct

Region Adherence

Aim in WSN context

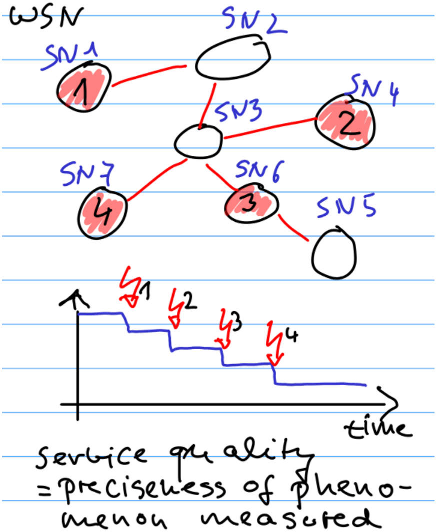
- sense distributed phenomenon with gracefully degrading quality if some sensors fail

Problems

- how to find RAS that provably reduce service quality very slowly?
- how to find RAs that withstand a large number of faults prior to delivering 0 service quality?

Our Contributions

- notion of maximizing extension of a RAS
- hardening transformation of RAS

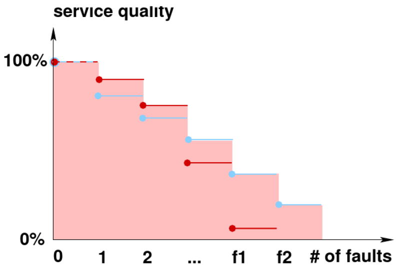


Maximizing Extension Approach

- given two RAS A and B using the same algorithm S and offering the same notion of service quality g
- RAS A is known to be a f_1-region-adherent with quality reduction function $r = \langle r_0, r_1, \ldots, r_{f_1} \rangle$
- RAS B is known to be a f_2-region-adherent with quality reduction function $r' = \langle r'_0, r'_1, \ldots, r'_{f_2} \rangle$
- and w.l.o.g. $f_1 \leq f_2$.
- Then, we know that there is a RAS C using algorithm S offering service quality notion g that is f_2-region-adherent with quality reduction function $r'' = \langle \min(r_0, r'_0), \ldots, \min(r_{f_1}, r'_{f_2}), r'_{f_1+1}, \ldots, r'_{f_2} \rangle$
- Thus, there exists a RAS C (and we know it) that potentially withstands more faults and/or reduces service quality not that much per fault

Maximizing Extension Approach (Cont.)

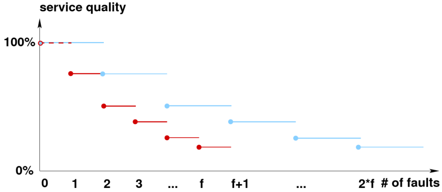
- RAS A has been proven to
  - have quality reduction function given in red
  - be $f_1 = 4$-region-adherent
- RAS B has been proven to
  - have quality reduction function given in blue
  - be $f_2 = 5$-region-adherent
- RAS C is guaranteed to
  - be $f_2 = 5$-region-adherent
  - having the "red area" quality reduction function



- a RAS implementing algorithm S is actually better in terms of region-adherence then it was previously known
- due to its RA properties, algorithm S may now be eligible for actual use in a WSN context

Hardening Transformation Approach

- Idea: transform f-region-adherent system into $(n+1)f$-region-adherent system by hardening the system against faults
- up to now: system switched to new region after single new fault
- from now on: system
  - remains for 1 to n faults in current region
  - switches to new region after $n+1$st new fault
- new system is obviously much more RA



Transformation

- all variables are replicated $2n+1$ times

Transformation (Cont.)

- reading a variable: replaced by a function that delivers the majority value of the $2n+1$ replicas[a]
- writing a variable: replaced by a function that writes the new value to all $2n+1$ replicas

[a] and writes this value back to all $2n+1$ replicas

Example of a Region-Adherent Sensor Network [3]

- WSN measures air humidity in a region
- SNs equipped with humidity sensors each
- SNs send measured value to a data sink (gateway node)
- SN may fail; in any case, they send a (potentially incorrect) humidity value

$$v_i \in \begin{cases} [l, u] & \text{if failed} \\ [v - \epsilon, v + \epsilon] \cap [l, u] & \text{else} \end{cases}$$

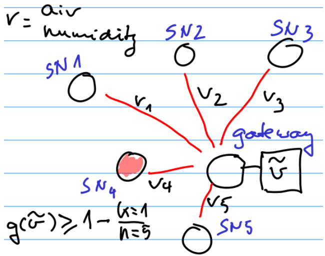
  v is the true air humidity value physically present
- gateway node calculates a mean value

$$\tilde{v} := \frac{1}{n} \sum_{i=0}^{n-1} v_i$$

- with service quality function

$$g(\tilde{v}) := \min \left\{ 1 - \frac{|\tilde{v} - v| - \varepsilon}{u - \ell},\ 1 \right\},$$

  it holds that $g(\tilde{v}) \geq 1 - k/n$ with $0 \leq k < n$ being the # of faults occured
- WSN is $(n-1)$-region-adherent



Conclusion

- WSNs for SCs must be
  - scalable
  - fault-tolerant
  - able to cope with dynamics
- realizable through FT concepts
  - replication
  - self-stabilization
  - region adherence
  - and combinations thereof
- gave examples of WSNs using these concepts

System Software and Distributed Systems
Department of Computer Science
University of Oldenburg, Germany

www.uol.de/svs

oliver.theel@uol.de

Conclusion

- WSNs for SCs must be
  - scalable
  - fault-tolerant
  - able to cope with dynamics
- realizable through FT concepts
  - replication
  - self-stabilization
  - region adherence
  - and combinations thereof
- gave examples of WSNs using these concepts
- looking forward to cooperations with you on these topics

System Software and Distributed Systems
Department of Computer Science
University of Oldenburg, Germany

www.uol.de/svs

oliver.theel@uol.de

Thank you for attending :)

- J. Kamenik, C. Peuser, V. Gollücke, D. Lorenz, R. Piechocki, M. Wasmann, and O. Theel, "ZeDDS – Fault-Tolerant Data Management in Wireless Sensor Networks with Mobile Users," in *Proc. of the 5th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '11)*. Lisbon, Portugal: IARIA, Nov. 2011, pp. 11–16, iSSN: 2308-4278.
- M. Hacker, "Design and Implementation of a Self-Stabilizing Sensor Network (in German)," Bachelor Thesis in Computer Science, University of Oldenburg, Department of Computer Science, System Software and Distributed Systems Group, Oldenburg, Germany, 2013.
- J. S. Becker, D. Rahmatov, and O. Theel, "Dependable Systems through Region-Adherent Distributed Algorithms," in *Proc. of the International Conference in Central Asia on Internet (ICI '13)*. Tashkent, Uzbekistan: IEEE, Oct. 2013.