Model-driven Development of Self-organizing Control Applications (MODOC)

H.-U. Heiß, J. Richling, H. Parzyjegla, D. Graff
Berlin Institute of Technology

T. Weis, A. Wacker, S. Schuster, S. Holzapfel
University of Duisburg-Essen

G. Mühl, N. Jeremic
University of Rostock

Scenario

Ubiquitous computing environment
- Embedded and intelligent end-user devices (e.g., smartphones, home automation)

Actuator/Sensor Network
- Wireless communication, gathering and processing of sensor data within network

Distributed Control Applications
- Cooperative services based on sensors, processing nodes, and actuators within vicinity

Model-driven Approach

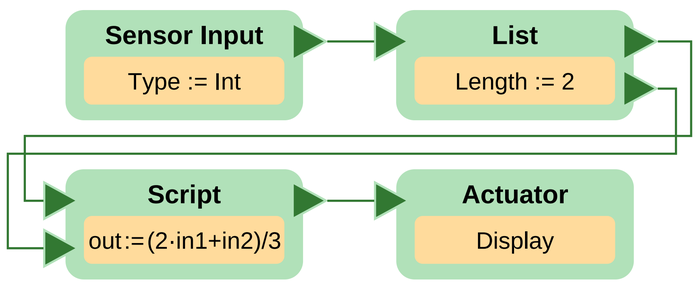
Data-flow oriented modeling language
- Modeling of application logic
- Specified by application developer

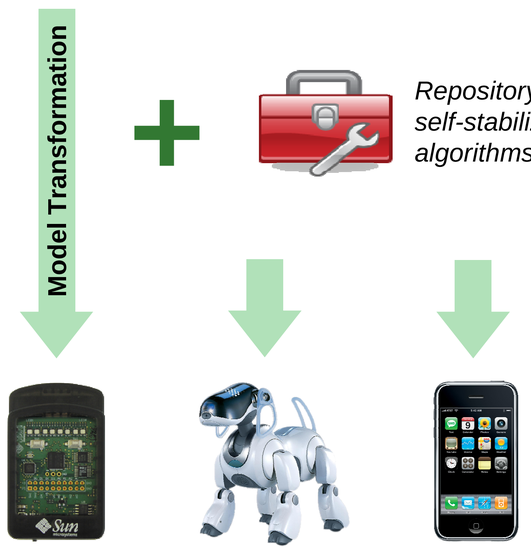
Repository of self-stabilizing algorithms
- Addressing non-functional properties (e.g., self-organization, adaptivity)
- Provided by domain experts

Model transformation
- Generation of executables
- Automatic integration of self-organization and self-stabilization properties
- Support for multiple target platforms

Benefits for application developers
- Self-x properties for free
- Automatic heterogeneity support

Application model.



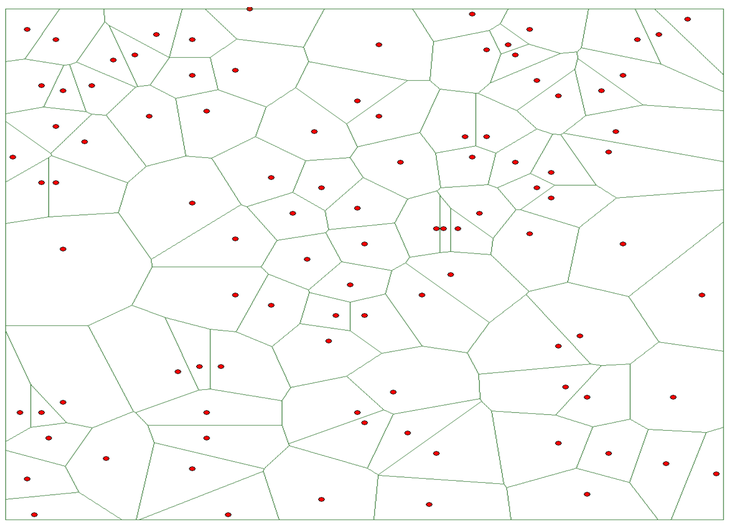


Repository of self-stabilizing algorithms.

Executables for target platforms.

Self-organizing P2P-Overlay

System Model
- Peer-to-peer overlay with location-related data
- Data has n-dimensional location coordinates
- Peers have coordinates in the same space
- Example applications: Location-based services, pervasive computing scenarios, massively multiplayer online games



Voronoi decomposition.

Goals
- Self-organizing peer-to-peer Voronoi overlay
- Self-stabilizing overlay despite of churning

Voronoi Decomposition
- Decomposition of the n-dimensional space determined by the distance to a peer
- Each peer is the center of a Voronoi cell
- Each peer is responsible for the data elements closest to it

Challenges

No global knowledge
- Peers must determine their Voronoi cell with local knowledge only
- Algorithms for constructing Voronoi diagrams are well known, but rely on global knowledge
- Self-organization is required, i.e., achieving a global property (Voronoi diagram) based on local interactions only

No central solution
- All peers are equal
- No administrator, maintainer or any other centralized authority
- Cooperative algorithm required

Churning of peers
- Peers can enter and leave the overlay anytime
- Many peers leave without closing their ‘gap’
- Overlay repair required? How to repair? How to detect that repair is necessary?

Self-stabilization
- No need to detect that repair is required
- Recover from any distortion of the overlay
- Only limitation: Overlay must be connected (path between any two peers exists)

Self-stabilizing Algorithm

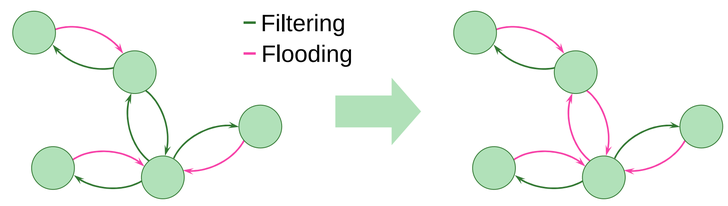
Algorithm basics
- Each peer learns its surrounding → constructs its own Voronoi cell
- Each peer keeps connections to all peers of neighboring Voronoi cells
- Non-neighboring peers are disconnected and reconnected with a close neighbor → assures that the overlay is not partitioned

Self-stabilization property
- Regularly gossip list of current neighbors to neighboring cells
- If the Voronoi cell changes, start gossiping to all connected peers

Adaptive Event Routing

Flexible publish/subscribe communication
- Homogeneous publish/subscribe routing: One algorithm for all edges
- Hybrid publish/subscribe routing: Different algorithms utilized simultaneously to combine their inherent benefits

Adaptivity by self-optimization
- Adapt routing strategy to changing conditions
- Fine-grained switching of routing algorithms per edge and direction

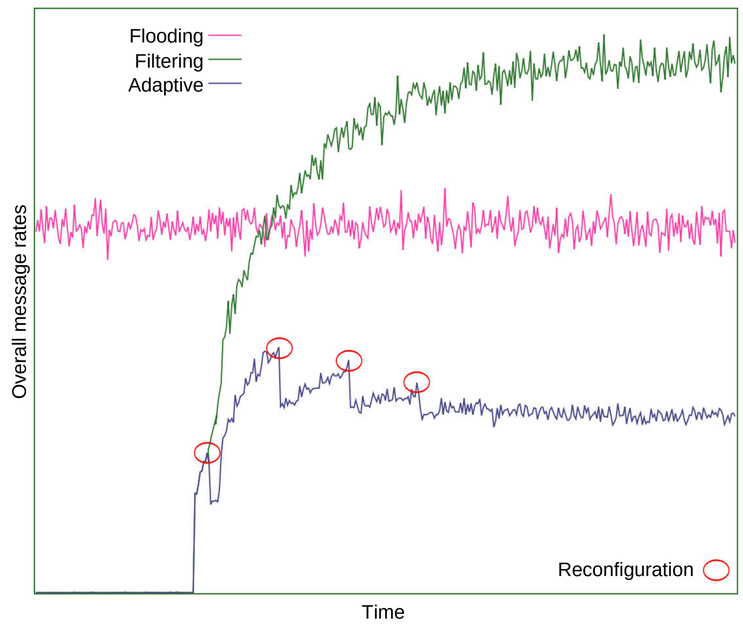


Edge-wise switching from notification filtering to flooding.

Optimization goals and criteria
- Minimize notification traffic
- Reduce subscription traffic
- Minimize computation costs

Properties
- Requires no global, but only local knowledge
- Completely distributed
- Never worse than homogeneous routing (e.g., flooding, filtering)

Discrete event simulation
- 127 nodes with increasing subscription rate



Simulation of flooding, filtering, and adaptive routing.

Summary and Outlook

Separation of concerns
- Business logic by application developer
- Non-functional properties by domain experts
- Composition by model transformation

Algorithm repository
- Self-stabilizing peer-to-peer overlay network
- Adaptive routing based on publish/subscribe

Future Work
- Detection of complex events and conditions
- Enforcement of safety constraints
- Round trip modeling to provide network feedback to simplify debugging and profiling

Web: http://www.kbs.tu-berlin.de/modoc E-mail: modoc@kbs.tu-berlin.de