Spatial Self-organization in Networks of Things

Kashif Zia and Alois Ferscha

Institut für Pervasive Computing
Johannes Kepler University Linz, Austria
{zia,ferscha}@soft.uni-linz.ac.at

Abstract. Miniaturized, wirelessly networked embedded systems combined with Peer-to-Peer computing principles have started to pervade into objects of everyday use, like tools, appliances or the environment, thus implementing ensembles of autonomous, interacting "networked things". With the development of the Peer-it framework, integrating a self-contained, miniaturized, universal and scalable embedded systems hardware platform, basically containing sensors, actuators, computing and wireless communication facilities, and a Peer-to-Peer (P2P) based software architecture, we have proposed a "stick-on" solution for the implementation of networks of things (NoTs). The Peer-it design and miniaturization ultimately aim to yield a "smart label", ready to be sticked on to literally every "thing" as a NoT enabler. The paper addresses the issue of spatial awareness of objects within NoTs, and proposes abstractions of space based on (i) topology, (ii) distance and (iii) orientation. Experiments are conducted to investigate on the ability of objects in a NoT to self-organize based on their spatial orientation.

Keywords: Autonomic Computing, Sensor/Actuator Systems, Context Awareness, Self-organization, Spatial Abstraction, Networks of Things.

1 Introduction

Pervasive computing has promoted a new era of computing, where the "computer" is no longer understood as a single device or a network of devices, but rather the entirety of services delivered through ad-hoc ensembles of electronic devices or information appliances. Such ensembles of objects are assumed to sense the physical world via a huge variety of cooperative and coordinated sensors, and acting via a plethora of (again) coordinated actuators. The nature and appearance of such objects appears to be hidden in the fabric of everyday life, like tools, appliances, machinery, furniture, clothing, etc., yet invisibly networked and omnipresent. Individually built with miniaturized embedded systems technology, usually for a specialized purpose, and engaging (short range) wireless technology for spontaneous communication, these objects raise the challenge of an operative, and semantically meaningful interplay among each other. "Meaningful" here referring to a variety of purposes for which a group of objects is spontaneously configured into a NoT service ensemble, be it to achieve a certain service goal, to improve service quality, dependability or performance, to increase fault-tolerance, etc.

K.A. Hummel and J.P.G. Sterbenz (Eds.): IWSOS 2008, LNCS 5343, pp. 333–339, 2008.

An acknowledged approach to address the challenge of “meaningful interplay” is by design, for example to design objects to be able to manage themselves in a more or less autonomous way, while at the same time designing ad-hoc ensembles of objects to self-organize. Self-management here stands for the ability of single object to describe itself, to select and use adequate sensors to capture information describing its situation, to find an interpretation for the situation (context), and to be ready to spontaneously interact with other objects in the vicinity, defined by the range of the involved wireless communication technology. Self-organizing (in the context of this paper) stands for the ability of a group of (possibly heterogeneous) objects to engage into a spontaneous ensemble based on interest, purpose or goal, to agree a common ensemble interest or negotiate a common ensemble goal, and to enforce individual objects to act so as to achieve the ensemble goal.

With previous work [1, 2] we have proposed to implement NoTs based on a miniaturized stick-on embedded computing systems, the Peer-it system, integrating sensor, actuator and wireless communication facilities, which connect these objects within limited vicinity. The Peer-it software stack implements features of self-management within a component based software architecture local to a peer, and features of self-organization in a totally distributed style. Interaction among peers (or NoT objects) at the application level is invoked based on the analysis of self-describing profile data exchanged across objects in vicinity. Peer-it based NoTs thus represent spontaneous ensembles of coordinated nodes in a wireless network, exhibiting features of autonomy like self-management at the node level, and self-organization at the network level.

Since NoTs operate in physical space, with nodes having locations in physical space, interesting questions arise with respect to the spatial distribution of nodes, particularly the impact of their distribution onto their ability to communicate, or even more challenging, to coordinate their activities. With this paper, therefore, we address issues on how the spatial distribution of nodes in NoTs, and their mutual spatial relationship impacts self-organization in NoTs. Specifically, we study the self-organization capability of NoTs with a certain number of objects at a certain density within a terrain of certain size. Section 2 motivates a scenario of investigation for NoTs within which objects are assumed to be able to sense their position and orientation. Section 3, based on simulation results, reports to which extent self-organization can be achieved among an ensemble of objects, if objects are only assumed to change their orientation, but not their position. We report evidence, that self-organization of “global” orientation in NoTs can be achieved with only local coordination (Section 4).

2 The Self-organization of “Things” wrt. Orientation

The spatial orientation of objects (or “things”) in physical space can be basically abstracted wrt. extrinsic or intrinsic frames of reference. In an extrinsic frame of reference, the direction of an object is defined in relation to fixed bearings like the cardinal directions north, south, east, west, and any arbitrary resolution of accuracy, or gravity (like high or low). The technological sensors in Peer-its to capture orientation of an object according to the extrinsic scheme are compasses and gyroscopes. Within an intrinsic frame of reference, the orientation of an object is defined in relation to part of itself or part of another object (e.g. front, back, left, rigt, etc.). The technological sensors in Peer-its to capture orientation according to the intrinsic scheme are ultrasound sensors or laser scanners. For the rest of the paper we consider orientation wrt. to an extrinsic frame of reference, and study cases of self-organization where the orientation information of objects (randomly placed in physical space) is the only control parameter for ensemble self-organization. Further, we concentrated on the re-adjustment capabilities of the whole population with respect to orientation.

Our research hypothesis is that the self-organization of orientation of things in NoTs can be achieved at global level with the knowledge of local information about neighborhood and corresponding orientations only. In addition, we study the effects onto self-organization when varying the node density, the number of nodes eligible for “re-organization” (i.e. with the ability of an object to autonomously change its orientation, e.g. with a built-in actuator like a motor), and the shape, size or dimension (2D, 3D) of the NoT terrain (here, only results for 2D, quadratic terrains are reported).

In a wider context, spatial self-organization has been a subject of interest in agent technology [5] and robotics [6]. Research in AI and cognitive science has also focussed on flocking behavior [7] and swarm intelligence [8]. Our research is not focussed on any of these areas which are specific application domains. Instead, we formulated dependencies between driving forces (node density and number of re-adjustments required) to gauge expectations. In an environment of varying settings, the known expectation can help understand the resultant behavior. It can also motivate the user to change the settings to get the expected result within acceptable range.

3 Experiment Settings and Results

In a scenario having randomly placed objects, we assume objects having actuators to re-adjust the orientation if required. Objects can sense and actuate a reaction within a constraint. We take this constraint to be 30°. It means that a node can sense orientation from 0 to 330 degrees, and an actuator can invoke a re-adjustment in orientation by steps of +30° or -30°. In this research, we consider a static network, in which nodes are not moving (change of orientation is not considered as a motion). In an environment of nodes having only orientation sensors, it would be insignificant to consider moving nodes, not knowing measures of their motion. But motion, alone can have dynamics (a constantly volatile neighborhood) which can impact the self-organization capabilities of network. We have retained this aspect for future research.

The investigation terrain, or Global Space (GS), is defined by the size, shape and dimensions of the space. For simplicity, we have considered 2D terrain space of 500 * 500 units of distance. Further we define the Zone of Influence (ZoI) of an object to be the number of other objects within interaction range. This zone can be of any shape. For simplicity we consider a 2D circular region as the shape being considered. As the ZoI expands, the number of objects an object can interact with increases. We conducted experiments for a varying number of objects (10, 25, 50, 100, 200, and 500), and increased the object coverage of ZoIs periodically for each case (percentages of covered objects to be 0.5%, 3%, 10%, 33%, and 100%). For each such subset, we increased the percentage of objects which needed re-adjustment from 1 to 50 % (1%, 2%, 5%, 10%, 20%, 30%, 40%, and 50%). At this lowest level each such experiment was performed for a 1000 times to meet statistical significance. The pseudo code described below sketches the simulation algorithm:

```
for each node                  // we refer to objects as nodes
     switch_destination = get_max_orient() // counts median of
            neighbors' orientation
     // if neighbors' count > 1, median is unique, and different from
            node's own orientation, do following
     switch_factor = get_min_factor() // returns either +30 or -30
     is_switching = TRUE
     for each node
            while (is_switching)
                   orientation=orientation + switch_factor
                   if orientation >= 360 orientation = 0
                   if orientation = switch_destination
                          is_switching = FALSE
                          has_switched = TRUE
get_min_factor()
     if (orientation!=switch_destination)
            if (Math.Abs(switch_destination-orientation) < 180)

                   return 30
            else
                   return -30
```

We refer to nodes which require re-adjustments to be the 'wrong' (‘wrongly directed’) nodes, as opposed to nodes which do not require re-adjustments ('right' nodes). After injecting a certain amount of wrong nodes into the terrain, each node in parallel calculates the median of neighbors' orientations within its ZoI. If the median is greater than 1 and different from the node's orientation, a re-adjustment is performed, referred to as a node 'switch'.

The main purpose of this experiment is to analyze the inter-relation between the number of nodes, the number of neighbors and the number of wrong nodes in a setting to question the possibility of reaching to a stabilized condition (no more switching required), and to inquire about relative improvement and limitations, from one iteration to the other. To analyze the inter-relationship among these factors, we concentrate on following indicators:

- **Wrong-Wrong** (WW) percentage.
  percentage of nodes which were wrong before simulation and still are wrong after simulation.
- **Wrong-Right** (WR) percentage.
  percentage of nodes which were wrong before simulation but are right after simulation.
- **Right-Right** (RR) percentage.
  percentage of nodes which were right before simulation and still are right after simulation.
- **Right-Wrong** (RW) percentage.
  percentage of nodes which were right before simulation but are wrong after simulation.

To analyze the relative improvement, we performed one iteration after the other and compared the results (up to 3 iterations).

Fig 1. shows a graphical view of a sample node distribution for 25 nodes, ZoI adjusted to include 10% of the neighbor nodes and 10% wrong nodes. Fig 2. shows a graphical view of a sample node distribution for 50 nodes, ZoI adjusted to include 3% of the neighbor nodes and 30% wrong nodes (nodes with orientation 0° are referred to as right).

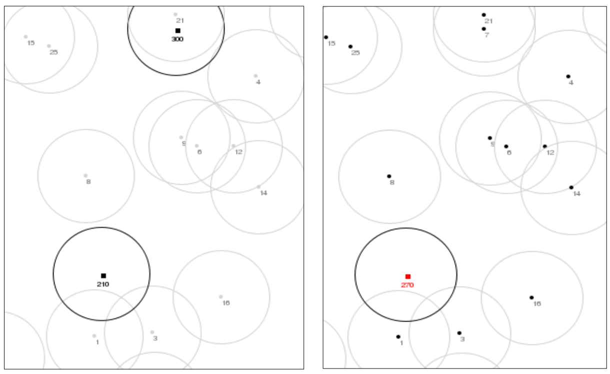

Fig. 1. Part of the scenario with 25 nodes. Left: Node 7 (top) and 19 (bottom) are wrong nodes represented by darker ZoI circles, having orientation equal to 300 and 210 respectively. Right: After two switches, node 7 has settled to 0; node 19 would also be settled to 0 after 3 more switches.

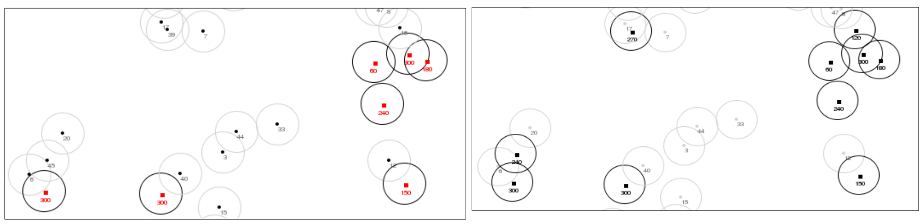

Fig. 2. Part of the scenario with 50 nodes. Node 39 (top-middle) shows expected behavior and settled to 0 after 4 switches. Node 45 (bottom-left) shows expected behavior but restricts other node with orientation 300 to perform switching. The same is the case with clusters of nodes (top-right). Only one node (top most) is to perform the switching whereas the other remains the same due to variety of neighbors' orientations (with no clear winner).

Analysis was done for each of 10, 25, 50, 100, 200, and 500 nodes. For limited space reasons, in Fig. 3 we only report on the results for the case with 25 nodes. The x-axis of graph in Fig. 3 is divided into five basic chunks representing cases where the ZoI is adjusted to get 0.5%, 3%, 10%, 33% or 100% neighbors. Each of these chunks is subdivided into levels of 1%, 2%, 5%, 10%, 20%, 30%, 40% or 50% of "wrong" nodes. The y-axis in Fig. 3 represents the possible (%-age) levels of WW, WR, RW and RR. Irrespective of actual quantities, we have stacked these values according to contributing percentages. It is important to note that the statistics of WW, WR, RW and RR are percentages stacked only for the 1st iteration, whereas improvement lines represent improvement after the 1st (thick gray dotted line), 2nd (gray line) and 3rd (black line) iteration. As for example, in the case of 50% wrong nodes inserted, and 10% neighborhood conditions (12 nodes out of 25 are wrong), lets say in the 1st iteration, we have the following results: WW = 28% (7 nodes), WR = 20% (5 nodes), RW = 4% (1 node), RR = 48% (12 nodes). This result is expressed in the 50% column of the 10% chunk on the x-axis. The percentage of improvement (1st iteration) line shows approximately 35% improvement (which is understandable when we see that after the 1st iteration, the number of wrong nodes decreased from 12 to 8). In the next iterations, the aggregated value of improvement has marginally increased.

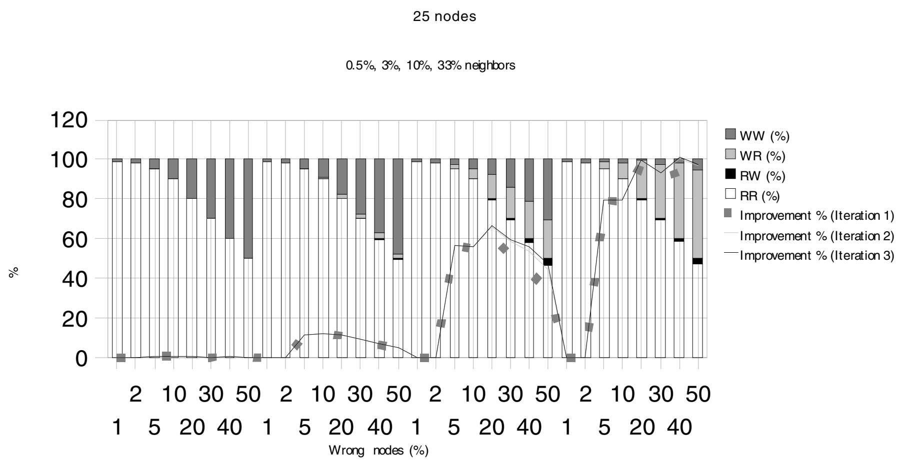

Fig. 3. Case 2: Number of nodes = 25; Percentage Stacked values (WW, WR, RW, RR) against 0.5%, 3%, 10% and 33% neighbors. Each neighborhood is further sub-divided into 1%, 2%, 5%, 10%, 20%, 30%, 40% and 50% wrong nodes.

In summary, from the experiments involving all the scenarios (10, 25, 50, 100, 200, and 500 nodes, not only the ones reported in Fig. 3), the following conclusions could be drawn:

– Finding 1: With an increasing number of wrong nodes, the improvement decreases for sufficiently large value of wrong nodes (more than 5%) and RW increases (thus decreasing RR).
– Finding 2: With an increasing number of neighbors, the improvement increases and RW increases (thus decreasing RR) followed by a decrease.
– Finding 3: With an increasing number of nodes, the improvement increases and RW increases (thus decreasing RR) followed by a decrease.

4 Conclusion

Self-organization of orientation in NoTs can be achieved at global level with a local level coordination mechanism only. In addition, the simulation results show that self-organization depends on number of factors including node density, percentage of nodes eligible for self-organization and global space specifications. Irrespective of the application domain, studying the behavioral inter-play between these factors is important to forecast effectiveness of self-organization. Alternatively, if applicable, application expectations can be tailored by re-adjustment of one or more of these factors.

Acknowledgements

We wish to thank the reviewers for their comments on the longer version of this work, and our shepherd for valuable suggestions on how to compact its contribution into less pages. She was very guiding wrt. to finding the right focus, and thoroughly and patiently helping to even get the paper error and typo free.

References

1. Ferscha, A., et al.: Peer-it: Stick-on solutions for networks of things. Das, S.K., Contri, M., Shirazi, B. (eds.) Pervasive and Mobile Computing Journal 4(3), 448–479 (2008)
2. Ferscha, A., et al.: Building Flexible Manufacturing Systems Based on Peer-its. Embedded Systems Design in Intelligent Industrial Automation, EURASIP Journal on Embedded Systems (October 2007) (special issue)
3. Gervasi, V., Prencipe, G.: Flocking by a set of autonomous mobile robots. Technical report TR-01-21, University of Pisa (2001)
4. Mataric, M., Zordan, V., Williamson, M.: Making complex articulated agents dance: an analysis of control methods drawn from robotics, animation and biology. Autonomous Agents and Multi-Agent Systems 2(1), 23–44 (1999)
5. Reynolds, C.W.: Flocks, herds and schools: a distributed behavioral model. Computer Graphics 21(4), 25–34 (1987)
6. Mondada, F., Guignard, A., Bonani, M., Floreano, D., Bar, D., Lauria, M.: Swarm-bot: from concept to implementation. In: Lee, G., Yuj, J. (eds.) Proc. of the IEEE/RSJ International Conference on Intelligent Robot and Systems, pp. 1626–1631 (2003)