An Energy-Efficient, Multi-Agent Sensor Network for Detecting Diffuse Events

Keywords: Sensor Networks, Multi-Agent Systems and Diffuse Events

Rónán Mac Ruairí [1,2] and Mark T. Keane [1]
[1]University College Dublin, Ireland, [2]Dundalk Institute of Technology, Ireland
ronan.macruairi@dkit.ie, mark.keane@ucd.ie

Abstract

Monitoring a diffuse event with a wireless sensor network differs from well studied applications such as target tracking and habitat monitoring and therefore we suggest that new approaches are needed. In this paper we propose a novel low power technique based on a multiple agent framework. We show how a set of simple rules can produce complex behavior that encompasses event characterization and data routing. We demonstrate the approach and examine its accuracy and scalability using a simulated gaseous plume monitoring scenario.

1 Introduction

Perhaps one of the major promises for sensor networks is the vision of a large-scale, environmental monitoring system that is deployed by a poorly-controlled dispersal and yet operates optimally for several years [Ramanathan et al. 2006; Werner-Allen et al., 2006; Estrin 2005; Szewczyk et al., 2004]. However, many challenges stand in the way of the vision, not least the issues surrounding efficient power usage to sustain the life of such a large, distributed sensing instrument. While, it is clear that hardware advances in mote and battery technology will play a key role in overcoming some of this power problem, we believe that significant advances in software will also prove important. Indeed, many environmental sensor networks will require significant software sagacity to achieve their goals in a power-efficient fashion.

Consider, any large-scale monitoring system that must amalgamate sensed data to accurately characterize an event, like a gas-plume moving across countryside. Such an instrument will need to route and collate significant amounts of data in order to identify and dynamically track a complex event as it moves through the environment. Not only will this sensor network have to carry out these tasks, but it will need to do them exhibiting an energy optimal and energy balanced strategy; for example, the overall network must take care not to use the same sensor nodes continuously. If they do nodes are lost as their batteries become exhausted. Conversely, using too many nodes to create a data route will ultimately use more energy and shorten the overall useful life of the network. Some current systems have considered some of these issues, though they have often concentrated on target tracking [Horling and Lesser, 2005; Lesser et al., 2003; Intanagonwiwat et al., 2000]. In this paper, we propose a novel multi-agent technique designed to handle the problems presented by large scale diffuse events.

1.1 Large-scale environmental monitoring

Energy management is perhaps the most critical constraint facing wireless sensor networks (WSNs) [Culler et al., 2004]. The routine response to this problem is to employ routing algorithms that limit communication overhead [Akyildiz et al., 2002]. Yet most of these techniques are designed for traditional target tracking applications - essentially scenarios with small numbers of data sources.

Large-scale environmental sensing of spatially-distributed events will routinely cause many nodes to generate readings at the same time. Furthermore, these multiple data points will need to be amalgamated efficiently. For example, consider a WSN tasked with reporting the limits (shape and location) and average concentration of gas plumes. While average concentration may be estimated accurately from a subset of data points, circumscribing the shape of a plume requires data points along the perimeter of the plume to be captured and fused.

Although achieving such high resolution and high fidelity might be viewed as the bad news, there is some good news for such applications. They can be reasonably delay tolerant. In particular the evaluation interval for environmental monitoring is likely to be of the order of months to years. In cases where the sensor network is tasked with an emergency response role the delay tolerance could be of the order of hours. Although quite different, both of these time intervals are delay tolerant relative to network latencies and routing delays. Further discussion of delay tolerant networking may be followed from Ho and Fall [2004].

We have suggested that algorithms developed for target tracking may not scale well in terms of efficiency as the size of a monitored phenomenon grows. Figure 1 provides some evidence for this intuition, as it shows the number of messages generated (as an indication of energy used) by a version of directed diffusion [Intanagonwiwat et al., 2000] running within the J-Sim network simulation package [Sobeih et al., 2005]. Although the algorithm ultimately scales in a manner close to linear it experiences an initial phase of explosive growth in the number messages generated. This behavior occurs as the size of the target exceeds a few percent of the overall monitored area. For us, large-scale phenomena are of interest and our aim is to achieve, at worst, linear growth in the number of messages generated as target size scales from below 1% to 30% of the monitored area.

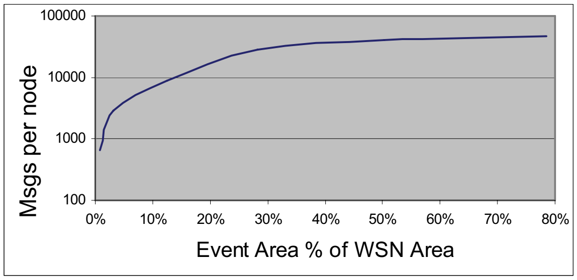

Figure 1. The average number of messages generated per node by directed diffusion for phenomena of varying scale (scale is shown as the percentage of the monitored area affected by the event).

2 The Multi-Agent Approach

Our application context is set in the near future, where low cost but powerful sensors can be deployed in very large numbers to monitor, with a focus on phenomena that require data from a large area to be integrated. The nodes are assumed to in place long before any event occurs, and conserve energy by using long sleep cycles – we are essentially suggesting that for long term monitoring most nodes must be powered down most of the time. Our approach also specifically trades latency almost entirely for energy conservation.

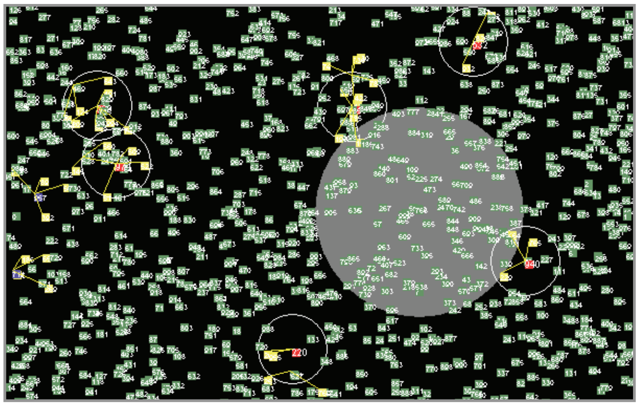

Figure 2. An agent is dispersed across several static nodes (square s), and the overall instrument consists of several agents at any one time traveling through the network. The large circle represents a contour of some notional event. The smaller circles represent the one hop (radio range) neighbourhood of leader nodes and the arcs depict the leader-member relationship that constitutes an agent.

We cause a number of agents to be generated spontaneously within the network and these agents gather and process the data. The approach is novel in that each agent is distributed across a number of nodes and the agents move by transferring part of their state to other WSN nodes. The WSN nodes themselves are static and may form part of many different agents during the lifetime of the instrument. We aim to show that a simple rule set can produce complex behavior allowing a small number of agents to satisfy accuracy and energy efficiency requirements of the application.

2.1 Distributed Agent Architecture

Our agents are composed of a set of roles distributed across multiple nodes. Each agent consists of a Leader role operating on one node, and multiple Member roles each operating on nodes adjacent to the Leader node. We do this for two reasons. Firstly it provides the agent with some spatially-distributed knowledge of the environment which can then be used for navigation. The agent uses this knowledge to navigate through the network to regions of different sensor readings. Secondly, the distribution of responsibilities achieved through nodes' adoption of roles provides some redundancy that gives the agent an ability to survive failure of individual nodes. This can occur because of energy depletion or damage - however this aspect is not examined in this work. The agent architecture consists of the agent state embodied in various roles along with a terse communication protocol for role deliberation and a rule set for navigation. Each node can assume only one role at a time (agents may overlap spatially but not in terms of membership).

The roles are Sleeper Leader, Member and Router. By default nodes adopt the Sleeper role. This role exists outside the scope of any individual agent and defines the default responsibilities – a sleep cycle punctuated by listening cycles in which the node is receptive to membership requests from nearby Leader nodes.

The overall WSN operates by seeding itself with a number of Agents. This is achieved by Sleeper nodes having a small probability of converting to the Leader role at each wake cycle if its sensor reports a data reading. A Leader recruits a number of nearby nodes that become Members. Once formed a data agent consisting of a Leader and a set of Members then uses its combined spatial knowledge to navigate according to a rule set. The final role – Router – is a specialized Leader role originating at the data sink that is used to harvest data from Leader nodes.

2.2 The Roles

2.2.1 The Sleeper Role

By default all nodes assume the Sleeper role, in which they operate a sleep cycle to conserve energy. They wake periodically and listen for membership requests from nodes in their immediate environment (i.e., one radio hop away).

Rule	Sleeper Role
S1	If a Leader's member request message is received adopt the Member role.

S2	If a member acknowledgement message (M1) is received from any neighbor role broadcast sensor reading to all neighbors.
S3	Check probability of becoming a Leader or Router Role (Leaders are only formed if a local data reading > 0, and Routers if the data sink is a neighbor).
S4	Calculate local hop count (based on messages received and local sensor reading) to event and forward to neighbors.

Table 1. The rules used to define the Sleeper role.

If multiple requests are received the Sleeper role causes its node to select one randomly and join that agent thus assuming the membership role. If no requests are received within some period the node powers down and the sleep cycle recommences.

2.2.2 The Leader Role

Leader roles form the basis of data gathering agents (data agents) and are created randomly within the network (via rule S3). Their first duty is to grow their membership. In our demonstration the Leader seeks to enlist a minimum of three neighboring nodes with the aim of providing good spatial coverage. The Leader then uses the sensor readings from these nodes, or their hop count to the nearest data reading, to calculate a gradient across its sensed area. The Leader selects one of its members, either upstream or downstream depending on the mapping algorithm in operation and declares it the new Leader. The old Leader forwards current mapping information – a series of tuples {reading, location, time} - to all neighbors and then all members retire to Sleeper role and the process commences again. Section 2.3 describes the Leader rule sets used to implement contour mapping.

2.2.3 The Member Role

Once a Leader requests a node's membership the Member senses its local data level and transmits this to the Leader along with a membership acknowledgement.

Rule	Member Role
M1	Acknowledge membership on receipt of message from Leader.
M2	Adopt Leadership if transferred (Leader Nomination) from Leader.
M3	Return to Sleeper role after some period if Leader Nomination message or keep awake message is not received from current Leader

Table 2. The rules used to define the Member role.

2.2.4 The Router Role

This role is randomly created by nodes neighboring data sinks (base stations). Initially Routers use a navigation rule {R1} similar to that of Leaders' {L1, L2} until an event contour is located. On the outward journey the Router navigation differs from Leaders in that a number of random moves are used to balance energy usage against burn out of nodes close to the sink. Once data levels close to the desired contour has been located Routers wait to encounter other agents. If they encounter an agent carrying a significant amount of data then this is harvested and the router attempts to return to the data sink. Each node in a Router's path records the identity of the previous node to allow Routers to retrace its steps for return journeys.

Rule	Router Role
R1	Move Router leadership to member with reading closest to desired contour level or – periodically – select a member at random.
R2	If reading is within some margin of the desired contour then stop transferring leadership, shed membership and increase sleep durations.
R3	If a data agent is encountered request its data, if more than some number of data points are available return to base.
R4	If returning to base, move Leadership by retracing outward path.
R5	If no move is available transfer Leadership to a randomly selected neighboring node.

Table 3. The rules used to define the Router role

2.3 The contour-mapping algorithm

Each agent moves through the static WSN node field by transferring the Leader role from one node to another. To map the contours of the event, the agents need to move through the WSN field until an event is detected (e.g., a gas plume). Once the event is encountered the agent should locate a sensor reading close to the required level and attempt to circumnavigate the contour. This can be achieved using the following rules:

Rule	Leader Role
L1	Transfer Leadership to the member node with a sensor reading closest to the desired contour.
L2	Do not transfer leadership to member nodes that recently held a Leader role.

Table 4. The rules used to define the Leader role

These rules although simple, are effective in enabling agents to follow the event contour. The second rule, L2, has a dual purpose – it causes an agent to move away from areas recently explored and it also contributes to overall efficiency as it causes other agents to avoid the same areas. Depending on the direction of an agent's initial approach to the contour some agents move in a clockwise direction while others travel in a counter clockwise direction.

3 Experiments and Results

To demonstrate the proposed multi-agent algorithm, we have developed a sensor network simulation incorporating an abstract event - essentially a disk shaped phenomenon. To gauge the effectiveness we have created two experiments, the first examines the accuracy of our approach by recording the positions reached by our agents as they seek to map the event contour. The second explores the effect of increasingly diffuse targets on the efficiency of our algorithm. All experiments in this section simulated 1000 nodes in a two dimensional 600x400 grid (see figure 2) and each trial ran for 2500 seconds. The target event and sensor nodes are stationary.

3.1 Parameter discovery and selection

A number of parameters dictate the operation of our approach. The following parameter choices (table 5) have been developed during our experimentation however we have not explored the parameter space in a systematic manner and we expect that more optimal parameter sets exist.

Parameter	Value
Number of members in a Data Agent	3
Sleeper sleep duration	3 sec
Sleeper wake duration	3 sec
No move period	3 sec
sensor minimum threshold	0.01
margin for rules R2 , L2	0.05
Number of data points (rule R3)	10
Number of moves (rule R4)	5
Member stay awake period (M3)	6 sec

Table 5. The parameters used in this work

3.1 Results

To test out hypothesis that the Leader navigation rules {L1,L2} will cause agents to circumnavigate the contours of an event we have recorded and superimposed the locations reached at each time step by agents over several different experimental runs (Figure 3).

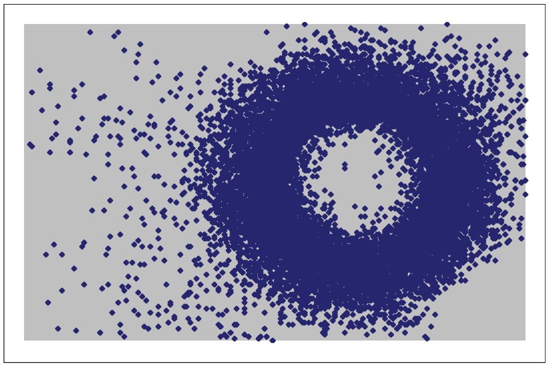

Figure 3. The superimposition of 100 runs, each showing the node location at each second of agent's Leader roles. Each run consisted of one agent navigating for 1000 seconds with a target event similar to that displayed in figure 2.

Though simple the rules are very effective in identifying and tracking the presented contour. In a real scenario we are likely to be interested only in the actual contour data returned to the sink node (as opposed to an accumulation of locations reached as shown in Figure 3). The following (figure 4) provides an example of the useful information generated and routed to the data sink during a single simulated task.

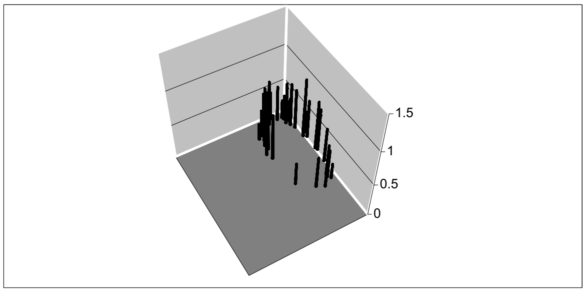

Figure 4. An example of the contour data (data target is 1.0 plus or minus 15%) returned during one experimental run using 6 Leaders and 4 Routers (the x axis is cropped and compressed compared to figures 2 and 3).

The second experiment examines the relationship between energy usage and the physical size of the target event. As discussed in this papers introduction this relationship is important as many extant algorithms have been developed for target tracking applications (figure 1) where the target is likely to account for less than 1% of the sensed area. In figures 4 and 5 the probabilities used for rule S3 (table 1) result in an average of 6 data agents and 4 router agents being used for the duration of each experimental run.

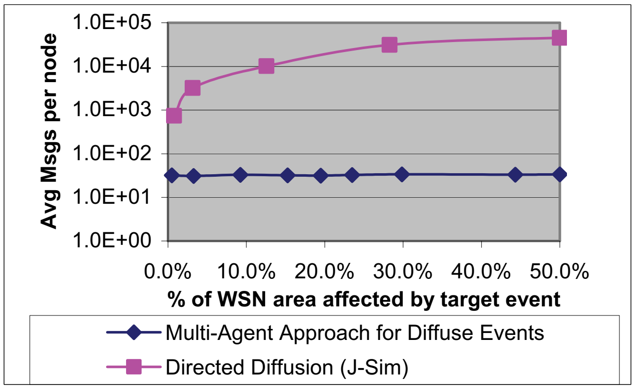

Figure 5. The number of messages generated per node (each point is an average of 20 trials) by our approach for phenomena of varying scale (shown as the percentage of the monitored area affected by the phenomenon) compared to the implementation of Directed Diffusion [Intanagonwiwat et al., 2000] packaged with J-Sim [Sobeih et al., 2005].

The amount of energy used, as indicated by the number of messages generated, is several orders of magnitude smaller than that of directed diffusion. The number of messages per node shown in figure 5 is on average only 0.15% of those for similar event sizes. While these numbers are not directly comparable due to the different aims of the algorithms it is clear that our approach is significantly more efficient for this application. The results also show (figure 5) that our approach is almost free of dependence on phenomenon size and therefore meets our design aims for use with large diffuse events. However as you might expect the number of agents used, as dictated by the probabilities used for Rule S3, has an effect on both the system accuracy and the energy used. In figure 7, we define accuracy as the total number of unique contour values returned to the data sink.

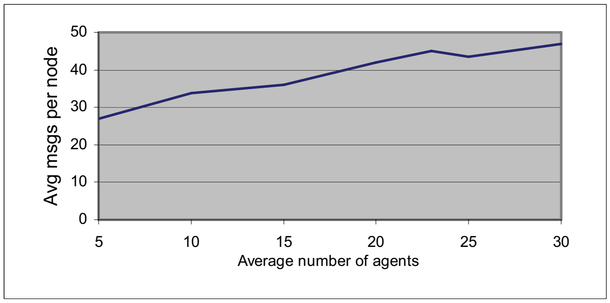

Figure 6. The effect of varying the number of agents deployed on the average energy usage per node for an event covering approximately 20% of the sensed area.

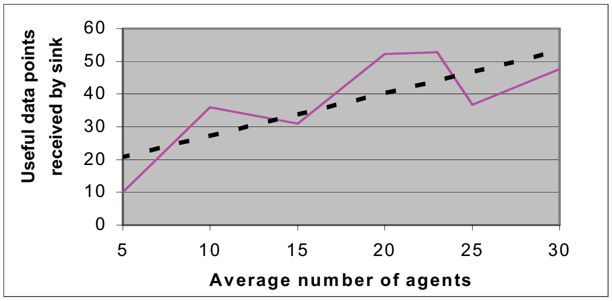

Figure 7. The effect of varying the number of agents deployed on the overall accuracy for an event covering approximately 20% of the sensed area.

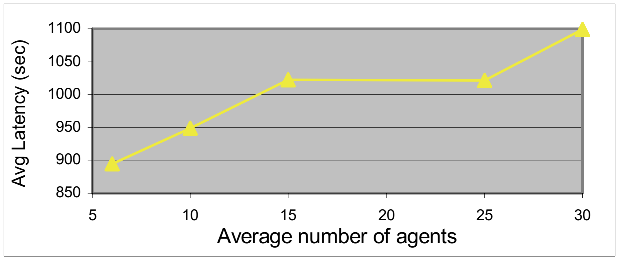

Figure 8. Average elapsed time before the event data was delivered to the data sink compared to the number of agents deployed.

The results shown in figures 6 and 7 show an increase in both accuracy and energy used as additional agents are used. This is to be expected as more agents should generate more messages, as well as the potential to deliver more useful data. The increase in energy expenditure does achieve a considerable increase in the amount of data delivered to the data sink. The figures (6 and 7) show that the trends are linear and suggests that the ratio of accuracy gain to energy expenditure is likely to be favorable, offering the potential that designers could tune the approach to their own accuracy or resolution requirements. However as noted in section 2, our approach eschews the elapsed time to return results to save energy and this is measured and shown in figure 8.

4 Related Work

Multi agent systems (MAS) are increasingly being applied to wireless sensor networks. O'Hare et al. [2005] argues that strong agency in the form of BDI agents [Rao and Georgeff 1995], might have significant potential for decision making within WSNs.

In the context of traditional target tracking [Lesser et al., 2003] discuss the application of various agent techniques to WSNs. More recently [Tynan et al., 2005] have shown how multi agent systems can be used for the rapid design and verification of sensor network algorithms and topologies. This is interesting as they have developed tools which automate the translation and deployment of agent architectures to nodes, specifically from Agent Factory [O'Hare et al., 1998] to Mica2 motes [Crossbow Technologies, 2006].

In this paper, we have advanced a multi-agent algorithm for sensor networks that is capable of both routing and processing data for diffuse events. The idea of a persistent agent composed of several roles has been informed and inspired by the role-based techniques described in [Mac Ruairí and Keane 2004] and by Beal's description of persistent nodes [Beal 2005]. In Beal's work nodes are symmetrical clusters of hundreds or thousands of nodes in an amorphous computing context that can move and preserve state using a simple rule set. Although our motivation in terms of resilience to node failure is similar; our agents are closer in context and scale to those in Rumor Routing [Braginsky and Estrin, 2002]. However our notion of an agent comprising of roles dispersed across different nodes, driven by a need for spatially-distributed information, is novel. Our approach also differs in that we embed features such as contour mapping specific to monitoring large diffuse targets.

5 Conclusions and future work

We have described and implemented a sensor network algorithm capable of mapping contours during large diffuse events such as gas plumes. Using a multi-agent paradigm where each agent is composed of roles distributed across multiple nodes, this algorithm achieves efficient routing and data processing within the sensor network. Our simulations have shown that this approach can accurately identify an event's contour and return the information to the data sink. A primary motivation of this work was to achieve such event detection in an energy efficient manner, given the finding that some extant algorithms degrade as the target size increases. Our results suggest that our approach scales extremely well in this regard and furthermore generates significantly less network traffic than a parallel implementation using directed diffusion.

In this study, our event model is quite abstract and so in the near future we will examine more complex phenomena. One interesting set of future issues is to examine the effect that different network densities has on the approach. By varying the contour-mapping rule-set we hope to allow the algorithm to adapt to network density and to provide additional application functionality beyond contour mapping.

We have offered a novel multi-agent approach to data processing within Sensor Networks. This is aimed at monitoring large scale diffuse events – a sensor network task understudied relative to target tracking. We believe this contribution to be important not only in advancing current practice but also in highlighting this progressively more vital sensor network application domain.

Acknowledgments

This work was funded in part by grants from Dundalk Institute of Technology in Ireland to the first author and Science Foundation Ireland under Grant No.03/IN.3/I361 to the second author.

References

[Akyildiz et al., 2002] Akyildiz, W. Su, Y. Sankarasubramaniam, And E. Cayirci.. A survey on sensor networks. IEEE Communications Magazine, Vol: 40 Issue: 8, 102-114, 2002.

[Beal 2005] Beal J., Amorphous Medium Language. Large-Scale Multi-Agent Systems Workshop at AAMAS 2005.

[Braginsky and Estrin, 2002] D. Braginsky, and D. Estrin. Rumor routing algorithm for sensor networks. In Proceedings of First ACM Workshop on Sensor Networks and Applications, Atlanta, GA, USA 22–31. 2002

[Crossbow Technologies, 2006] Crossbow Technologies Website, http://www.xbow.com/, products section, accessed May 2006.

[Culler et al., 2004] D. Culler, D. Estrin, M. Srivastava. An Overview of Sensor Networks. In IEEE Computer , vol:37 , no:8 , 41-49. 2004.

[Estrin 2005] Deborah Estrin. Reliability and Storage in Sensor Networks. in CENS Technical Report #59, 2005.

[Ho and Fall, 2004] Melissa Ho and Kevin Fall. Delay Tolerant Networking for Sensor Networks. In the First IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004). October 2004

[Horling and Lesser, 2005] B. Horling and V. Lesser. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, Vol. 19, Num 4, Cambridge University Press, 281-316. 2005.

[Intanagonwiwat et al., 2000] Intanagonwiwat, C., Govindan, R. and Estrin, D. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In proc. of the Sixth Annual ACM/IEEE Int. Conference on Mobile Computing and Networking (Mobicom), Boston, Massachusetts, 56-67, 2000.

[Kochhal et al., 2003] Kochhal, M., Schwiebert, L. and Gupta, S. K. S..Role-based hierarchical self organization for ad hoc wireless sensor networks. In WAF , San Diego, USA, 98–107. 2003.

[Lesser et al., 2003] Lesser V., Ortiz, C.L. Jr., Tambe , M. (eds). Distributed Sensor Networks : A Multiagent Perspective. Kluwer Academic Publishers, 2003.

[Mac Ruairí and Keane 2004] Mac Ruairí, R. and Keane, M. T. 2004. Role Swapping in Multi Agent based Sensor Webs. In Proceedings European Conference on Artificial Intelligence, (ECAI'04),Valencia, Spain.

[O'Hare et al., 1998] O'Hare, G.M.P., Collier, R., Conlon, J. and Abbas, S., (1998) Agent Factory: An Environment for Constructing and Visualising Agent Communities, 9th AICS Conference, Irish Artificial Intelligence and Cognitive Science Conference, UCD, Dublin, 1998.

[O'Hare et al., 2005] O'Hare G.M.P., Marsh D., Ruzzelli A. and R. Tynan, Agents for Wireless Sensor Network Power Management. Proceedings of International Workshop on Wireless and Sensor Networks (WSNET-05), Oslo, Norway IEEE Press, 2005.

[Ramanathan et al. 2006] N. Ramanathan, L. Balzano, M. Burt, D. Estrin, T. Harmon, C. Harvey, J. Jay, E. Kohler, S. Rothenberg, and M.Srivastava. Monitoring a Toxin in a Rural Rice Field with a Wireless Sensor Network in CENS Technical Report #62 , 2006.

[Rao and Georgeff 1995] A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In Proceedings of International Conference on Multi Agent Systems, pages 312–319. AAAI Press / MIT Press, 1995.

[Sobeih et al., 2005] Sobeih, A., Chen, W., Hou, J.C., Kung, L., Li, N., Lim, H., Tyan, H. and Zhang, H.. J-Sim: a simulation and emulation environment for wireless sensor networks. In Proceedings 38th Annual Simulation Symposium, San Diego, CA, USA, 175 – 187, 2005.

[Szewczyk et al., 2004] R. Szewczyk, E. Osterweil, J. Polastre, M Hamilton, A Mainwaring, D. Estrin. Habitat Monitoring with Sensor Networks, Communications of the ACM, Vol. 47, No. 6, pp. 34-40, June 2004.

[Tynan et al., 2005] Richard Tynan, Antonio G. Ruzzelli, G.M.P. O'Hare A Methodology for the Development of Multi-Agent Systems on Wireless Sensor Networks. SEKE'05, the 17th International Conference on Software Engineering and Knowledge Engineering, Taiwan, China, IJSEKE press, 2005.

[Werner-Allen et al., 2006] Geoff Werner-Allen, Konrad Lorincz, Mario Ruiz, Omar Marcillo, Jeff Johnson, Jonathan Lees, and Matt Welsh. Deploying a Wireless Sensor Network on an Active Volcano, In *IEEE Internet Computing*, Special Issue on Data-Driven Applications in Sensor Networks, March/April 2006.