A Survey on Sensor Networks from a Multiagent perspective

Meritxell Vinyals[1], Juan A. Rodriguez-Aguilar[1] and Jesus Cerquides[2]

[1] *IIIA, Artificial Intelligence Research Institute*
CSIC, Spanish National Research Council
[2] *WAI, Dep. Matemàtica Aplicada i Anàlisi*
Universitat de Barcelona
Email: {meritxell,jar}@iiia.csic.es,cerquide@maia.ub.es

Sensor networks have arisen as one of the most promising technologies for the next decades. The recent emergence of small and inexpensive sensors based upon microelectromechanical systems ease the development and proliferation of this kind of networks in a wide range of actual-world applications. Multiagent systems have been identified as one of the most suitable technologies to contribute to the deployment of sensor networks that exhibit flexibility, robustness, and autonomy. The purpose of this survey is twofold. On the one hand, we review the most relevant contributions of agent technologies to this emerging application domain. On the other hand, we identify the challenges that researchers must address to establish multiagent systems as the key enabling technology for sensor networks.

1. INTRODUCTION

Sensor networks (SN) have been identified as one of the most promising technologies for the future [1, 2, 3] due to: (1) the recent emergence of small and inexpensive sensors based upon microelectromechanical system (MEMS); (2) the set of advantages they offer in front of other monitoring technologies; and (3) the wide range of real-world applications that have been already identified for this technology. As this new technology emerges and applies to real-world domains, it poses a variety of new challenges to researchers leading to some new active areas of interest concerning hardware and software.

In this survey we focus on the software challenges SNs pose from the perspective of multiagent systems (MAS). SNs have been identified as an application domain with high potential for MAS because of their suitability for modelling autonomous, self-aware sensors in a natural, flexible way [4, 3]. Sensor networks fall into the category of complex, distributed, interconnected, and rapidly changing systems, identified in [5] as a hard and challenging domain for autonomic computing. Issues such as organisational structuring, coordination, collaboration, and distributed, real-time resource allocation are critical for their success. In SNs, sensor agents may go beyond reacting to their local situation; they may collaboratively determine what to do and with whom while ensuring that some collective, global properties are achieved. However, SNs may fairly vary from one to another depending on the features they exhibit. We realise that SNs with different features lead to different problems when considering its enactment and operation. Therefore, one of our contributions in this work is the definition of a taxonomy that classifies SNs in different families, each one leading to different problems of varying complexities when enabling them.

Next, we analyse what MAS can offer to SNs and how MAS technology help enact and deploy MAS. At this aim, we summarise the most relevant contributions to these topics while identifying the SN features in our taxonomy that each contribution focuses on. Moreover, we also identify some open issues that pose challenging problems that deserve further research, namely: (1) to widen the scalability scope of current algorithms; (2) to design algorithms that better abide by the physical distribution of a SN; (3) to design efficient adaptive approaches (the smaller the changes, the less the adaptation effort); (4) to endow organisational approaches with self-adaptive capabilities; (5) to design computationally-efficient, decentralised (economic) mechanisms for SNs with multiple owners; (6) agent-based middlewares that support self-* capabilities.

The rest of this paper is structured as follows. In section 2 we introduce SNs and analyse their properties as a system and their suitability as an application domain for MAS. In section 3 we propose a taxonomy for SNs. Next, in section 4 we identify the main research topics for sensor networks, describing the most salient MAS contributions and results. Section 5 reviews the contributions of MAS research to the development of SNs. Finally, in section 6 we identify some promising research directions for the MAS community.

2. SENSOR NETWORKS AND MULTIAGENT SYSTEMS

In this section we briefly discuss the advantages of SNs as a solution to the problem of performing continuous wide-area monitoring. Moreover, we discuss what MAS can offer to SNs.

SNs consist of sensing devices distributed in a wide area that coordinate to produce meaningful information from individual, raw, local data. They have emerged as an alternative to other, already-existing monitoring technologies, typically composed of a single or a few calibrated *static* sensors that acquire homogeneous streams of data in highly-controlled environments. Sensor networks, however, are usually wireless, *dynamic* networks composed of a large number of (possibly heterogeneous) sensors. These sensors distributedly acquire pieces of information that may substantially differ in content, resolution, and accuracy. Table 1 outlines a comparison between SNs and alternative monitoring technologies. Notice that SNs are increasingly regarded as a better solution than alternative monitoring technologies because they:

- are non-invasive (they can remotely sense without being embedded in the monitoring target);
- can cover wide-range areas by using a large number of inexpensive and small sensors;
- are inherently fault-tolerant and robust to node failures (because of their distributed structure), and hence suitable to monitor remote and/or hostile environments; and
- allow to fuse heterogeneous data stemming from different interest areas and pieces of information that substantially differ in content, resolution, and accuracy.

Moreover, SNs have demonstrated their applicability through a large number of applications in domains of varying characteristics such as, for instance, habitat monitoring, biomedical applications, smart spaces or distributed robotics. We refer the interested reader to [6] and [7], which survey a wealth of applications of SNs.

Consider now the generic challenges faced to make SNs operate. Following [8], sensors in a SN must typically coordinate their actions to achieve system-wide goals. Such challenge has been in the heart of MAS research from its inception. Therefore, MAS appears as a promising enabling technology for SNs. Nonetheless, there are further features required by a SN that make the application of agent technology particularly challenging:

Physical distribution. Sensors are typically deployed over some wide area. Thus, the operation of a SN involves computation and information resources that are physically distributed.

Resource boundedness. There are constraints regarding computational, energy, and communication resources. Nonetheless, sensors must be able to operate though they are low-powered, their computation capabilities are low, and communication is slow and intermittent. Moreover, SNs are typically located in rapidly changing environments, hence requiring that they exhibit anytime capabilities to cope with time constraints.

Information uncertainty. Eventually, there is incomplete information about the states of sensors, the network, and the environment where the SN is situated, as well as about the effects of sensors' actions.

Large scale. They are usually composed of a large number of sensors, making unfeasible computational approaches that do not scale with the number of sensors.

Decentralised control. Sensors may need to coordinate to achieve tasks that cannot be achieved by the operation of a single sensor. Importantly, such coordinations must occur without centralised control.

Adaptiveness. The operation of a SN is expected to be adaptive, namely the network must autonomously and dynamically adapt to either (external) changes in the environment wherein it is situated or structural (inner) changes in the network (be it because of the unreliability or failure of some sensors or the addition of new sensors) so that its system-wide goals are still achieved.

Notice that first four features above refer to *structural* aspects of a SN, whereas the last two features refer to *functional* aspects.

At this point, it is quite natural to think of agents *controlling* some sensor(s) to achieve a SN system-wide goals. Then, MAS research provides a wealth of formalisms, algorithms, and methodologies to help enact SNs that satisfy the requirements above. Nonetheless, despite their value, we agree with [8] that agent technologies cannot be applied directly because they usually fail to address: (i) severe constraints on (computational, energy, and communication) resources; (ii) malfunctioning and failures; (iii) situatedness in highly dynamic environments; and (iv) large scale settings. As a positive sign, current contributions by MAS research (e.g. [9, 10]) demonstrate that it is possible to exploit existing agent technologies.

Besides control tasks, *information* agents can be also employed in SNs for information-processing tasks [11]. *Information* agents are in charge of autonomously acquiring real-time sensor data to process it and obtain the relevant information required for operational decision making. At this aim, these agents must be able to cope with data uncertainty (be it because of missing, delayed, or noisy data), sensor failures, and resource constraints that limit readings. Besides supporting decision making, information agents are key

Feature	Sensor Networks	Alternative technologies
Sensor	Low-cost, low-power	Expensive, power-consuming
Coverage	Wide-area	Small-size areas
Monitoring environment	Remote and/or hostile environments	Highly-controlled
Robustness	Fault-tolerant and robust to node failures	Non-robust
Invasiveness	Non-invasive	Invasive
Data acquisition	Irregularly sampled datasets	Regularly sampled datasets
Architecture	Distributed	Centralistic
Connectivity	Low-bandwidth	High-bandwidth
Power	Battery	Electric

TABLE 1. Differences between sensor networks and other monitoring technologies.

to realise the sensor web [12], bridging the gap between the physical and on-line world. Thus, information agents can permit the sensing of events in the physical world, from the traffic congestion in our roads to the pollution in our rivers, to subsequently deliver information services on the Internet.

Finally, it is reasonable to wonder whether agent technology can help us build SNs. The paradigm of agent-oriented software engineering (AOSE) is argued to be well suited for developing complex software systems in distributed and dynamic environments [13]. Therefore, AOSE appears as a promising approach to support the intricate task of developing SNs. Different middleware architectures have been proposed [14, 15] to support the development, deployment, execution, and maintenance of sensor-based applications [16]. Therefore, SN programmers can implement their applications on top of some out-of-the-box middleware for SNs. Alternatively, agent technology offers several approaches to develop agent-based sensor applications, namely: (i) the development of a multi-agent application on top of some middleware for SNs (e.g. [17]); (ii) the development of agent-based middlewares that support agent-based programming (e.g. [18, 19, 20, 21, 22]); and (iii) the extension of an existing multi-agent platform with middleware capabilities for sensors (e.g. [23] extends the existing agent framework EMAA [24], whereas [25] extends Agentscape [26]). We put off a more thorough discussion on this topic to section 5.

3. A TAXONOMY FOR SENSOR NETWORKS

As discussed in section 2, MAS have a high potential to help build SNs that autonomously coordinate their actions to achieve their system-wide goals. However, the complexity of this endeavour may fairly vary because SNs may also largely differ depending on the features of: (i) sensors; (ii) the network putting together sensors; (iii) the the environment wherein the SN is situated; (iv) their tasks. For instance, consider a SN with electric-powered sensors (no energy constraints) situated in a fully-observable environment where events occur at a very low rate and goals do not require collective actions. Consider now a SN with battery-powered sensors situated in a partially-observable environment where events occur at a very high rate and goals require the coordinated action of sensors. Obviously both SNs pose very different problems if we intend to endow them with autonomous behaviour. Therefore, the features of a SN and its environment, along with the goals it pursue, must be carefully considered by any MAS-based approach because different features lead to different problems. In what follows, we thorough analyse such features by defining a taxonomy that founds a characterisation of the problems posed by each type of sensor network. Moreover, further on we will employ our taxonomy to locate where MAS research has mostly contributed.

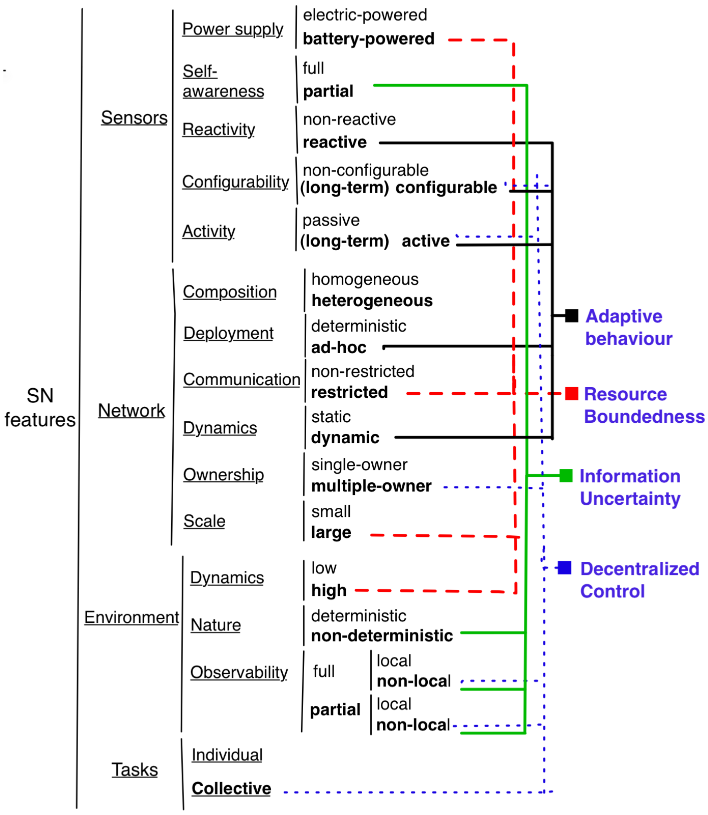

FIGURE 1. A taxonomy for sensor networks.

Figure 1 shows our taxonomy where challenging SN features are boldfaced. Features are linked with the operational challenges that they pose. Next, we detail our taxonomy and discuss the relationship between the SN features in the taxonomy and the challenges for agent-technology listed in section 2.

3.1. Sensors

Sensors are the building blocks that compose sensor networks. Each sensor contains some microprocessor and communication device in addition to its sensing device. Moreover, they may also include actuators to act over the environment. Out of all sensor features, we distinguish the following as the most salient ones:

Power supply. Sensors' power can stem from either batteries (finite power source) or electricity (power source is infinite). Battery-powered sensors pose the problem of acting under energy constraints.

Self-awareness. A sensor can be either *partially self-aware*, in case some of its features are not directly accessible; or otherwise *fully self-aware* when it can access a complete representation of all its features. For instance, sensors unaware of their own location or their battery levels are partially self-aware. Partially self-aware sensors must deal with uncertainty about their own states.

Reactivity. A sensor can be either *reactive*, if external environmental changes may affect it or *non-reactive.* Examples of reactive sensors are: sensor that move or whose batteries are recharged as a consequence of interactions in the environment, or sensors that may either disappear or be destroyed because of environmental events. Reactivity sensors must be adaptive to adapt to the changes of their states caused by the environment.

Configurability. *Configurable* sensors can have (some of) their sensing, communication, and processing capabilities configured. They are said to be *non-configurable* otherwise. Examples of configurable sensors are: mobile sensors, sensors with different transmission powers, or sensors that can switch between different sensing qualities. Furthermore, configurations can be either *short-term* or *long-term* effects. Long-term configuration may restrict a sensor future operations. For instance, a battery-powered sensor that switches to a configuration that allows consuming energy when limited compromises energy availability for future (eventually highly energy-consuming) actions. Configurable sensors enable (adaptive) control because they can switch between configurations to achieve a SN goals. Notice that long-term configurations require to consider the configuration's effects in the future, leading to challenging, non-myopic control approaches.

Activity. *Active* sensors count on actuators to act over the environment, whereas *passive* sensors limit to perceiving, processing information, and communicating. For instance, active sensors can control illumination or heating sources, or apply control forces over some material. Moreover, similarly to configurations, sensor actions over the environment are classified into *short-term* and *long-term* depending on their effects over the environment. Examples of actions with long-term effects are: those moving objects in the environment or changing some environmental conditions (e.g. the heating or the cooling of a room). Likewise configurable sensors, active sensors enable (adaptive) control because sensors can operate on the environment to achieve a SN goals. Moreover, sensors' actions may have long-term effects that forces sensors to look-ahead in the future to plan their operations.

3.2. The network

The network in a SN is composed of all deployed sensors along with their communications links. We propose to characterise a network over five dimensions:

Heterogeneity. A network can be either *homogeneous*, if composed of sensors of the very same type (e.g. same processing, sensing, or communication capabilities), or *heterogeneous* if composed with sensors with different capabilities.

Deployment. The deployment of a network can be either *deterministic*, when sensors' locations are set at design time, or *ad-hoc.* Deterministic deployments are common in friendly and accessible environments, whereas ad-hoc deployments are typically undertaken over open or remote areas. For instance, forest monitoring SNs have their sensors deployed ad-hoc by throwing them out from a helicopter. An ad-hoc deployment forces sensors to adapt their operation to start from an uncontrolled, random topology.

Communication. Communication in a network can be *restricted* (due to low bandwidth or costs or unreliability), or *non-restricted.* For instance, networks with radio frequency transmission are a good example of restricted communication. Restricted communication poses the challenging problem of operating under resource boundedness, in this case, communication boundedness.

Dynamics. A network can be either *dynamic*, when composed of *reactive* sensors or when its communication links vary as a consequence of environmental events and phenomena (e.g weather conditions, animal attacks, ...); or *static.* For instance, a network whose sensor nodes appear/disappear in an unpredictable way is a good example of a dynamic network. In a dynamic network, a SN must exhibit adaptive behaviour to cope with unexpected topological changes.

Ownership. A network may have either a *single owner*, when all sensors belong to the very same stakeholder or company, or *multiple owners.* In a multiple-owner network, although different sensors may have different goals, they may all benefit from collaborating by exchanging information and tasks. At this aim, a SN must deploy decentralised control mechanisms that allow selfish sensors to cooperate.

Scale. A network may be *large*, if composed of thousands of nodes, or *small* otherwise. Large-scale networks constrain sensors to act in its vinicity and operate using local information in a decentralised way. Any centralised or global control that requires global information is not affordable.

3.3. Environment

We regard the environment of a SN as all external events of interest to the SN. We propose to characterise a SN environment along three dimensions:

Dynamics. An environment exhibits either *high* dynamics, when events and phenomena occur very frequently (the environment changes very rapidly), or *low* dynamics [3] Highly dynamic environments constrain SNs operation with severe time restrictions that enforces sensors to anticipate environment changes.

Nature. An environment can be either *deterministic*, whenever every action over the environment has a single, guaranteed effect (there is no uncertainty about the environment state that will result after performing an action), or *non-deterministic*. In a non-deterministic environment, a SN must operate under uncertainty about the outcome of its actions, and hence its control becomes more complex.

Observability. An environment can be *fully-observable*, when sensors' observations can find out the environment state without uncertainty, or *partially-observable*. Furthermore, observability may be either *local* to sensors, when sensors only need their own observations to determine their local environment, or *non-local*, if they need further observations from other sensors to complete their local view. In partially-observable environments, a SN must operate with incomplete information, and therefore its control mechanisms must deal with uncertainty. Moreover, in any non-local observable environment, a SN requires some coordination mechanism that allows sensors to share their observations.

3.4. Tasks

The last element of our taxonomy are the tasks specified by a SN designer. Tasks deeply affect the challenges that SNs pose during their operation. Tasks can be classified in : *individual* tasks, when sensors' actions are separately undertaken, or *collective* tasks, when sensors' actions coordinate to achieve it. For instance, a SN whose goal is to sample an area so that each sensor individually adapts its sampling depending on environmental local measures can be considered as a goal with individual actions. However, if the sensors collaboratively sample the area to obtain some global measure, this goal would involve collective actions. Collective tasks require that a SN counts on a decentralised control mechanism that allows sensors to coordinate in order to achieve the SN goals.

4. MAS RESEARCH CONTRIBUTIONS

In this section we review the main contributions of MAS research to the field of SNs. We classify the main tasks that a sensor network performs in four categories: *localization*, *routing*, *information processing*, and *active sensing*. Therefore, we review the contributions made by MAS research for these tasks. Moreover, we highlight more rellevant MAS techniques for SN as summarised in figure 2. Henceforth, in this section we mainly consider approaches where agents control sensors.

4.1. Localization

Each node in a SN can be aware of its own location as well as the identity and location of its neighbours. Typically, when sensors are deployed in an ad-hoc manner, the network topology has to be constructed in real time and updated periodically as sensors fail or new sensors are deployed. Moreover, because of this ad-hoc development nodes usually do not know their own position. Hence, localising sensor nodes by estimating its spatial coordinates is an important and popular area of research. However, the localisation problem does not naturally present itself to the MAS approach. Hence, contributions to this area typically stem from other communities. Thus, we address the interested reader to [27] [28] for good introductions and reviews to the techniques employed in localisation. Notice too that localisation is a very challenging issue when agents are not controlling sensors but instead *using* sensors. This is the case in settings where agents are autonomous robots exploring a sensor-instrumented environment [29].

4.2. Routing

Routing algorithms in SN have to be efficient in a network typically characterised as wireless (each node communicates using radio signals), ad-hoc (the set of nodes changes over time), and energy-constrained (nodes are battery-powered). These requirements differ from those for ad-hoc wireless routing protocols because the latter require to minimise energy consumption and extend network lifetime. Therefore, an active area of research is the development of energy-efficient ad-hoc wireless routing protocols that allow to route information in an energy-efficient way. [30, 31] survey routing algorithms for wireless SNs. Since most of these approaches stem from network research, hereafter we only review the most significant MAS contributions. These contributions provide examples of the performance and applicability of relevant MAS techniques such as computational

[3] Static environments are not considered because there would be no phenomena to monitor, and hence no need for a SN.

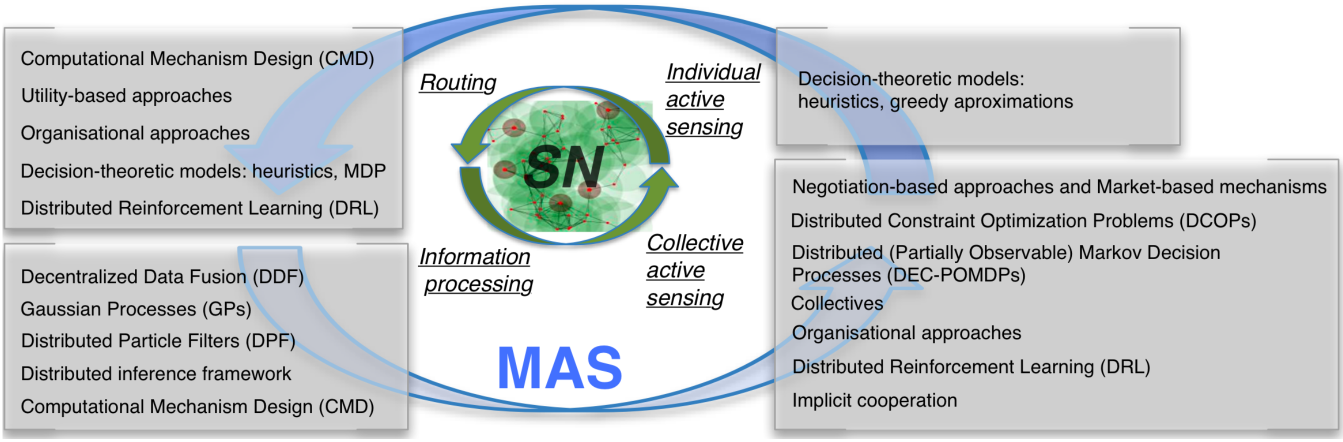
Computational Mechanism Design (CMD)
Utility-based approaches
Organisational approaches
Decision-theoretic models: heuristics, MDP
Distributed Reinforcement Learning (DRL)
Decentralized Data Fusion (DDF)
Gaussian Processes (GPs)
Distributed Particle Filters (DPF)
Distributed inference framework
Computational Mechanism Design (CMD)
Routing
Individual active sensing
SN
Information processing
Collective active sensing
MAS
Decision-theoretic models: heuristics, greedy aproximations
Negotiation-based approaches and Market-based mechanisms
Distributed Constraint Optimization Problems (DCOPs)
Distributed (Partially Observable) Markov Decision Processes (DEC-POMDPs)
Collectives
Organisational approaches
Distributed Reinforcement Learning (DRL)
Implicit cooperation

FIGURE 2. Main MAS techniques applied to SN research.

mechanism design, MAS organisations, or distributed reinforcement learning to the problem of routing in SN.

Some of the most significant MAS contributions result from applying Computational Mechanism Design (CMD) techniques [32]. CMD studies the design of interaction protocols for self-interested agents to achieve a specific outcome (maximise a global function or achieve some global properties). CMD focus on designing mechanisms that incentivise agents to behave as the designer intends (usually through a payment scheme that provides payments to agents in exchange of their services). Along this line, Rogers et al. [33] develop a new energy-aware, self-organised routing algorithm where sensors only transmit data to the sink. In this mechanism agents take their routing decisions using only local information (the expected lifetimes and their distances to the sink and to their neighbours). Sensors use a communication protocol to find and select a sensor that is willing to act as a mediator. A payment scheme ensures that sensors will only act as mediators whenever the overall performance of the SN is improved. Although this CMD schema is typically applied to SNs whose sensors are owned by different stake-holders, Rogers et al. also propose to apply it to the design of a single-owner SN based on the simplicity and well-studied properties of CMD.

In [34] Padhy et al. propose a new utility-based, energy-aware, self-organising routing protocol combined with adaptive sampling that finds the cheapest cost route from an agent to the sink. The so-called USAC protocol considers that a sensor might prefer to transmit its data via a more energy-consuming route if the least energy-consuming route contains some sensor(s) in a highly dynamic environment. Hence, the cost of a link from one agent to another is calculated considering that the energy that an agent spends transmitting data is at the expense of the energy used for its own sensing. The USAC protocol is specifically designed for its application to GLACSWEB, a battery-powered deployed SN that collects glaciers' environmental data. The use of in the GLACSWEB simulator shows that the USAC protocol outperforms the currently deployed, non-agent based, GLACSWEB protocol.

In [35] Zafar et al. propose to use organisational knowledge to improve routing performance in a wireless SN. MAS have typically used organisational design to guide communication policies (when agents communicate, how often, with whom, and with what priority) [36]. Due to this tight connection between organisation and communication, [35] proposes to integrate organisational knowledge into general-purpose wireless network routing algorithms that are generally employed to support agent communication. By knowing which agents are interested in communicating with whom and its priority, Zafar et al. can find optimal paths faster and save significant bandwidth for application use. This approach has been tested in a detailed network-level simulation of CNAS [10, 37], an agent-based, power-aware SN for atmospheric monitoring. The results confirm that incorporating organisational knowledge into routing policies significantly increases bandwidth availability at the CNAS application layer.

An alternative approach to the traditional multi-hop routing is the use of mobile sinks that periodically visit the vicinity of each sensor and collect their data in a one-hop routing style. Mobile sinks can significantly extend the lifetime of a SN by eliminating the need for expensive hop-by-hop routing. However, routing using mobile sinks requires that sensors decide when to send their data, and so solve the so-called transmission scheduling problem. In [38] Turgut and Boloni propose simple to help agents make their decisions using local and historical information: a human-inspired heuristic, a stochastic heuristic, and a constant-risk heuristic. In [39] the same authors propose a more sophisticated model using Markov Decision

Processes (MDPs). They present two variants of decision-theoretic algorithms based on MDPs using and disregarding history information. All these algorithms were empirically tested via simulation and compared in terms of performance, energy, and data loss ratio.

4.3. Information processing

Information processing in SNs includes tasks such as fusion, inference and prediction. In a SN, sensors usually need to exchange observations and data to: (1) process raw data to answer queries by the user (e.g report the position of a tracking object or temperature at some point); and (2) to build and maintain a common and consistent picture of the environment over which to perform control. Because of communication and energy restrictions, a centralised state estimation by a single node that receives all sensor data is, in general, infeasible. Hence, there is a need for decentralised (agent-based) algorithms, that allow sensors to locally estimate their states while minimising the amount of bandwidth used. Against this background, MAS researches have introduced the concept of *information agent* [11] for data acquisition and processing tasks. Information agents are in charge of locally acquiring and process real-time sensor data in a SN.

One active line of research in this area focuses on distributed fusion methods. These allow informative agents to locally fuse their observations with the information received from their neighbours. As an example consider the *Decentralized Data Fusion* (DDF) method [40]. DDF provides a robust, modular and scalable solution to the problem of obtaining common and consistent state estimates across a SN by allowing sensors to communicate information rather than states. The use of information measures allows agents to distinguish new information from prior knowledge. Moreover, this fusion process is straightforward since fusion of information is additive. DDF has been used in MAS to allow agents to share and fuse information about target positions in a SN [41, 42]. In these works DDF helps synthesise a global utility function for agents' actions, which combined with a distributed decision-making mechanism lets agents reconfigure sensors to maximise the information gathered. Although DDF is limited to Gaussian distributions, an extension to this method, the *Bayesian Decentralized Data Fusion* (BDDF) algorithm, overcomes this limitation [43].

Gaussian Processes (GPs) are another fusion method successfully applied to wireless SN to build a probabilistic model of the environment [44]. GPs enable agents to apply principled Bayesian inference to functions. As to environmental monitoring, these functions model how environmental parameters change over time. On the one hand, GPs have been used to model the correlation of different temporal measurements of a single sensor [45]. Thus, a GP builds a probabilistic model that allows agents to individually perform active sampling by only sensing at the most informative moments and thus avoiding redundant measurements over time. On the other hand, GPs have also been used to model spatial correlations among sensors [46, 47]. Incorporating spatial correlations into the probabilistic model allows agents to perform collective active sensing by avoiding redundant measurements among its neighbours and over time. GPs have been evaluated by performing: (1) environmental monitoring using data collected from a network of weather sensors located on the South coast of England[4] [46]; and (2) decentralised coordination of a team of mobile robots using real-world sensor data collected at the Berkeley Intel Research Lab[5] [47].

An alternative distributed fusion method also shown useful in SN is the method of Distributed Particle Filters (DPF) [6] [48, 49]. DPF stand for set of particles standing for candidate state descriptions that are adjusted based on sensor observations. They allow nodes to keep a belief over state histories instead of just single states. DPF can cope with highly dynamic environments and handle non-Gaussian distributions. For instance, in [49], Rosencrantz et al. use DPF to determine which measurements are worth sharing using a query-response system. Sensors keep a local particle filter and query one another for useful sensor measurements (a query is a small set of randomly selected particles). Sensors use query information to only transmit the most informative measures. To evaluate this approach, Rosencrantz et al. performed experiments using physical robots in a scenario where a team of robots was tasked to estimate a posterior distribution over an opponent's location from laser range data. Results show a great performance of the DPF in dynamic environments using a large number of robots.

Further information processing tasks are considered in [50], where Paskin and Guestrin introduced a general architecture for distributed inference in SN. This framework allows to solve information processing tasks that can be cast as inference problems, namely: (1) *linear regression* (or function fitting), a general data fusion technique to reduce communication; and (2) *distributed sensor calibration* [51], which involves automatic detection and removal of sensors' biases. Their architecture is based on distributing an asynchronous message-passing version of a typical centralised inference algorithm, the junction tree (JT) algorithm. Moreover, Paskin et al. also propose a novel distributed JT formation algorithm to compile the initial problem in a JT structure. Their approach also considers the quality of the communication links in the agent communication network as well as how

[4] `http://www.aladdinproject.org/alexdemos/sa_gp_flash/sa_gp_flash.html`

[5] `http://db.csail.mit.edu/labdata/labdata.html`

[6] Also known as Sequential Monte Carlo Methods.

to dynamically adapt to changes in a SN. Nonetheless, this approach is limited to small, tractable scenarios because the optimality of the JT algorithm can exceed the demand of resources in a bounded SN.

Another central issue in information fusion is the decide what data sensors should share. This task becomes even more complex when sensors are owned by different stake-holders with individual goals. As argued in [52], as far as SNs become ubiquitous, it is likely that individual sensors may benefit from interacting with sensors that are owned by different stake-holders. Rogers et al. illustrate such situation with the example of a farmer using a SN to measure local weather conditions to control an irrigation system. The farmer may benefit from using measurements from neighbouring landowners to track changing local weather patterns. In this context, there is no incentive for any sensor to transmit observations to its neighbours. In MAS these problems are typically addressed using Computational Mechanism Design (CMD) approaches. Within this setting CMD can provide mechanisms that incentivise the sharing of observations and ensure that communication bandwidth is fairly distributed. Rogers et al. [52] evaluate two incentive-compatible[7] allocation mechanisms in a simulated aerial surveillance scenario in which sensors with individual goals make efficient use of a limited bandwidth. In a related work, Dash et al. [53] extend a classical auction mechanism, the Vickrey-Clarke-Groves (VCG) mechanism, to deal with the interdependent valuations of data that arise in a distributed SN tracking scenario. The VCG mechanism is incentive-compatible and individually rational[8] and therefore provide a solution to the problem of data fusion in a multiple-owner SN. Notice that the use of CMD poses some computational problems that impedes coping resource-bounded SNs. On the one hand, the need for a single auctioneer introduces a single point of failure. On the other hand, the optimisation problems that an auctioneer must solve are very costly.

4.4. Active Sensing

According to the taxonomy in section 3, sensors can be configurable. Thus, sensors can change their configurations to vary the content, resolution and accuracy of their observations to maximise, in an energy-efficient manner, the information gathered over time. This implies that the way a SN senses is not passive, and hence it must be provided with *active sensing*: the capacity of autonomously reconfiguring and coordinating its sensors in order to maximise the amount of information perceived over time. Thus, agents in a SN must be provided with decentralised decision-making algorithms that allow them to perform an intelligent setting of sensors' parameters. We have classified contributions to active sensing into: *individual*, when each agent configures itself independently on other agents (individual decision making), and *collective*, when agents have to coordinate in order to determine their joint configuration (collaborative decision making).

4.4.1. Individual active sensing

In this section we discuss approaches to a type of active sensing well-studied in the literature, the so-called active sampling: nodes only use local information in order to reconfigure themselves with the aim of sensing at the most informative moments. Active sampling requires protocols that allow agents to individually adapt their sampling rate (how often to sample during some time interval) as well as their sampling schedule (when to sample). Padhy et al. [34] develop a novel mechanism for adaptive sampling that allows each sensor to adjust its sampling rate depending on the environment dynamics. Each sensor uses a regression model to forecast future data as a function of the last measurements. The optimal sampling rate is assessed as the one that keeps the confidence interval within a fixed limit. In this approach, an agent lowers its sensing frequency when it is capable of correctly predicting future values, and raises it otherwise.

Related to the above-mentioned work, Kho et al. [45] propose decentralised control algorithms for information-based adaptive sampling based on a novel generic information metric. In this approach agents infer the temporal variations of environmental parameters using GP-based regression. Then they use Fisher information as a novel measure of information for their sensor samples. Kho et al. introduce: (1) Their start proposing an optimal adaptive sampling algorithm that maximises their information metric by performing GP regression on the individual sensor nodes; (2) an algorithm that also uses GP regression but performs a greedy approximate optimisation to reduce computational costs; and (3) an algorithm that uses heuristics, rather than GP-based regression, to further reduce computation. These algorithms are evaluated in a simulator built upon high-fidelity models of a deployed wireless SN for real-time accurate flood forecasting (FloodNET [54]). To mimic the FloodNET scenario as realistically as possible simulation parameters are initially set to realistic data for batteries, tide readings, and cloud cover. In the FloodNet scenario sensors are solar-powered, have new battery levels each day, their available energy is distributed over the daily hours, and the phenomena of interest follow a daily pattern. Kho et al. empirically show that the heuristic approach, the most computationally-efficient out of their algorithms, increases the value of information gathered respect to some naïve non-adaptive methods and the protocol

[7] The dominant strategy is to truthfully reveal their private observations' values to the auctioneer.

[8] All agents in the system derive a non-negative utility from participating in the mechanism.

heuristic proposed in [34].

4.4.2. Collective active sensing

Collective active sensing strategies deal with SNs whose sensors need to coordinate to collectively perform some sensing task. Here agents face a collaborative decision-making problem: to cooperate to have sensors performing sensing tasks while maximising the SN global utility. At this aim, agents may need either: (i) to assess how to allocate sensors to tasks (or missions)[9]; or (ii) to assess their sensors' actions/configurations.

Next, we analyse MAS contributions to collective active sensing based on the technique that they use. We group techniques into the following categories: negotiation-based, market-based, distributed constraint optimisation, collectives, distributed Markov decision processes, organisation-oriented, distributed reinforcement learning, and implicit cooperation.

Negotation-based. In [56], Sims et al. propose a model to efficiently allocate sensors to sensing tasks. The approach allows to dynamically form coalitions of agents, each one responsible for a fixed sensing task in a sector. Thus, agents in each coalition work together to provide maximum coverage with the minimum number of agents (resources). Coalition managers are responsible for changing and negotiating over resources using an iterative distributed negotiation process aimed at maximising the global utility of the SN. The proposed negotiation protocol is based on the Contract Net protocol [57], but it can deal with interdependent tasks (tasks with positive utilities for the same resources). This approach also assumes that resources cannot be shared among coalitions and that the negotiation process evaluates each resource separately without considering sub-additive or super-additive relationships among resources. The protocol obtains either the optimal solution (when considering all the interdependences) or a sub-optimal one (when the chain of interdependences is cut at some length). In other words, considering interdependent tasks leads to higher computational costs.

Market-based mechanisms assume that each sensing task is aware of some utility function that it can employ to compose bids over sensor configurations. For instance, Abdallah et al. [58] formulate a generalisation of a classic resource allocation problem, the so-called setting-based resource allocation problem, to apply it to domains like SNs where a resource can be used (configured) to fulfil the needs of more than one task. They propose to solve this problem using a combinatorial auction: a central node (auctioneer) collects bids over sensor configurations from tasks to subsequently assess the optimal configuration of sensors. Notice that the role of the auctioneer is centralised and the winner determination problem for combinatorial auctions is very expensive, compromising the scalability of this approach.

Alternatively, the approaches in [59, 60] can better handle scalability issues. Both approaches provide solutions to the sensor-mission problem, namely the problem of agents assessing the best assignment of available sensors to missions. Firstly, Johnson et al. [59] formulate the Multi-round GAP (MRGAP) algorithm to solve a static version of the sensor-mission assignment problem. Following this algorithm each agent treats each mission as an individual (knapsack) problem.[10] An advantage of the MRGAP is that it can be implemented in a distributed fashion for missions that do not contend for the same sensors. In this case, the agents responsible of such missions can assess their assignment simultaneously. Moreover, Chalmers et al. also propose some heuristic-based schemes to solve the dynamics version of the sensor-mission problem (in a dynamic environment where missions may start at different times and last differently). Similarly, in [60], Le at al. propose an adaptation of the MRGAP algorithm, the so-called GAP-E, to allow agents to deal with dynamic environments by considering the competition for resources, the possibility of sharing resources, and multiple types of sensors. Notice though that both MRGAP and GAP-E have only been evaluated via simulation using randomly generated problem instances.

Distributed Constraint Optimisation. Some problems that arise in SNs have been modelled in MAS as *Distributed Constraint Optimisation Problems* (DCOPs). DCOPs are a formalism that captures the rewards and costs of local interactions in a team of agents, each of whom aims at choosing a set of individual actions [61]. Some examples of SN problems that have been cast as DCOPs are distributed scan schedule problems (e.g. how to detect and track mobile objects), or the coordination of sensors and actuators (e.g. to detect acoustic fairing and vibration damping on aircrafts [62, 63]).

MAS researchers have proposed several optimal algorithms to solve DCOPs [61, 64, 65]. Nonetheless, optimality requires significant communication and computation costs that may not be affordable. Therefore, optimal algorithms limit to small-scale SNs. More interesting to SNs are sub-optimal algorithms that can handle large-scale coordination via local interactions and distributed control. On the one hand, researchers have proposed stochastic algorithms such as the Distributed Stochastic Algorithm (DSA)[63], the Distributed Breakout Algorithm (DBA)[63], or the Maximal Gain Messaging (MGM)[66] algorithm, which have all been successfully applied to large SNs. Moreover, these algorithms have been recently characterised in the framework of k-optimality [66],

[9] Namely, to solve some multi-agent resource allocation problem[55].

[10] Notice that all problems together form an instance of the Generalised Assignment Problem (GAP).

which provides quality guarantees on the algorithms' solutions, hence allowing agents to bound the error on their decisions. On the other hand, Farinelli et al. propose Max-Sum (MS) [?], a message-passing algorithm from the family of graphical models, for multiagent coordination. MS has been tested on real-hardware to solve generic DCOPs (graph-colouring) [?] and to solve the sensor-to-target assignment problem [42][11]. In the latter problem, MS is combined with an information processing technique, DDF (see section 4.3), which allows to build and maintain a distributed picture of the environment and to specify agents' rewards.

One common assumption made by DCOPs is that agents know the utility of their actions, namely the reward of their joint actions in the environment. Regarding SNs, this may be too strong an assumption. Hence, this issue has been recently tackled by MAS researchers. On the one hand, Jain et al. [67] develop DCOP algorithms for problems where agents must explore the environment to find their actions' rewards. At this aim, they propose different extensions to DCOP algorithms that employ different measures to trade-off exploitation versus exploration at run time. These measures were tested on robots, deploying DCOPs on a distributed mobile SN whose agents maximise the accumulated signal strength within a time limit. An alternative approach that goes one step further is presented by Stranders et al. [68] to coordinate a set of mobile sensors for information gathering. They propose the MSm-l algorithm, which extends MS: (1) by incorporating two general pruning techniques to speed up MS; and by (2) using a non-myopic approach that allows agents to optimise over a finite, adjustable look-ahead. Stranders et al. show that MSm-l enables sensors to coordinate their movements with their direct neighbours to maximise collective information gain while predicting measurements at unobserved locations using a GP (see section 4.3). Up to date, this approach has only been tested via simulation using a small number of sensors. In general, observe that incorporating uncertainty regarding actions' rewards calls for computationally costly solutions that compromise scalability.

Collectives. Probability collectives (PC) is a broad framework for analysing and controlling distributed systems, with deep theoretical connections to game theory, statistical physics, and optimisation. Unlike DCOPs, PCs can deal with problems with highly correlated utility functions that cannot be factored. Moreover, it is a general approach to continuous, discrete, time-extended variables. In [41] Waldock and Nicholson show how PC can be used for cooperative sensing to solve the sensor-to-target assignment problem. They exploit PC to sample the joint space of sensor actions to discover an optimal collective sensing strategy. Here PC are combined with the DDF method (see section 4.3) to share and fuse information and define agents' utility functions. Notice that PC are powerful to cope with uncertainty because optimisation is performed over the probabilities of actions rather than over the actions themselves.

Distributed Markov Decision Processes. Further challenges in SNs have also been modelled in MAS with the aid of Distributed Partially Observable Markov Decision Processes (Distributed POMDPs). DEC-POMDPs allow to model coordination problems where a team of agents must plan under uncertainty. Unfortunately, the problem of finding the optimal joint policy for a general DEC-POMDP is restricted to scenarios with very few agents and it usually exceeds the available resources of a SN due to the complexity of the model [69]. Against this background, and inspired by a realistic SN coordination problem, Nair et al. formulate in [69] a general model, the so-called Networked Distributed POMDPs (ND-POMDPs), which allows agents to exploit local interactions in SNs. A ND-POMDP algorithm combines the planning under uncertainty required by a POMDP with the exploitation of local agent interactions of DCOP algorithms. Recently, MAS researchers have developed efficient algorithms to solve ND-POMDPs such as LID-JESP [69], which features locally optimal policy search, SPIDER [70] and FANS [71], which use approximation schemes, and CBDP [72], based on a bottom-up dynamic programming approach. Although these contributions allow to significantly scale up the number of agents, the applicability of ND-POMDP algorithms is still restricted to small/medium scenarios and may exceed available resources in a SN.

Organisation-oriented. Organisational control has been identified as an effective way to approach the scaling of AI systems [36]. Organisation control is a multi-level approach that uses long-term organisational goals and roles as guidelines for agents' detailed operational decisions. In a multi-agent organisation, the communication and action spaces of agents are restricted by their roles. This reduces the use of non-local information and the number of reasoning agents, and then efficiency of the MAS improves through organisational assumptions. Moreover, the use of hierarchies allows to introduce supervisory levels whose agents deal with more abstract information. Supervision accelerates the propagation of information through the organisation, allowing to deal with larger MAS.

Given this background, it is not surprising that the MAS literature has proposed organisational approaches to control large and dynamic SNs. Thus, Horling et al. [73] adopt an organisation-oriented approach to deal with the issues of the *Distributed Sensor Network Challenge Problem* [3], a real-time tracking application where a set of doppler radars must track one or more targets that move along arbitrary paths

[11] The sensor-to-target assignment is the problem of assigning sensors to targets in order to maximise some measure of the system-wide performance.

in some area. To reduce the communication burden, the environment is partitioned into sectors. The organisation encompasses three agent roles: sensor (scan/track), sector manager, and track manager. Sensor and sector manager are static roles run by agents in charge of either a sensor or a sector respectively. Track manager is a dynamic role that can be taken on by different agents. This role is dynamically created by a sector manager whenever it detects a new target. This approach has been empirically evaluated in a realistic simulation environment called Radsim [3].

Organisation-oriented approaches have also been used in the context of SNs to organise agents into coalitions so that each coalition has enough resources to fulfil its tasks. Abdallah et al. [74] propose to approximately solve the coalition formation problem using an underlying organisation to guide the formation process. Agents in the organisation use reinforcement learning techniques to learn policies that speed up the search for future coalitions. The organisation used is a hierarchy whose lowest level represents resources controlled by a single agent and the rest of levels are composed of managers that control and assign tasks to their direct subordinates. To achieve scalability, managers are aware of an abstraction of the state of the organisation. When a manager receives a task, it asks his children for contributions to assign them subtasks. Managers use reinforcement learning to learn the optimal order to ask children for their contributions. If a manager cannot successfully allocate some task, it forwards it up the hierarchy.

The Dynamic Regions Theory (DRT) [75] appears very close to organisational approaches. DRT is based on a dynamic partition of a large-scale network into a set of none-overlapping regions, each one running an algorithm for a specific task. Agents consider where (which region) to coordinate with agents within the same region to cooperatively run algorithms. The DRT uses an organisation for coordination purposes: each agent locally runs a region identification process (at regular intervals) as well as a regional organisational policy to implement adaptive role swapping. Simulations (of a SN deployed to monitor a city's air quality) show that DRT helps empower agents with self-organisation capabilities. Although the identification process and role swapping policy must be specified by a SN designer, the results show that the application of DRT can lead to significant energy savings.

Although organisation-oriented approaches have proven to be valuable, there are still open issues for MAS research. On the one hand, the autonomous adaptation of the organisational structure. On the other hand, the adaptation of the topology of the underlying network to increase the ability of the organisation.

Distributed reinforcement learning (DRL). DRL addresses the problem faced by multiple agents that must learn to coordinate their actions such that they collectively solve a given task [76]. Notice that this problem is at the heart of MAS and SNs. DRL assumes that agents, likewise in reinforcement learning, obtain some reward (learning feedback) by performing trial-and-error actions. Tham and Renaud [77] evaluate several classical DRL algorithms for a lighting control application whose agents' objective is to cooperatively learn to completely illuminate a room in an energy-efficient manner. The algorithms are implemented on real Berkeley motes and evaluated in terms of communication, computation, energy costs, and speed of convergence to optimal policies. Moreover, the authors provide a discussion about the trade-offs required to employ DRL algoritms in resource-constrained SNs. Similarly, Sandhu [78] proposes a model for a WSN-based commercial lighting control system using DRL.

However, the approaches above do not conveniently exploit the structure of the problem. Kok and Vlassis [79] propose the Sparse Cooperative Q-learning techniques to help agents approximate a global payoff function based on the topology of a coordination graph. Coordination graphs are a very promising framework that exploits dependencies between agents to decompose a global payoff function into a sum of local terms [80]. The authors provide an evaluation of these techniques and a comparison with some traditional DRL algorithms on a simulated SN where agents learn to coordinate to perform a target-tracking task.

Implicit cooperation. All the techniques analysed so far for collective sensing employ *explicit* cooperation, namely agents explicitly coordinate their decisions. *Implicit* cooperation is defined as the process of cooperative inference through the exchange of measurements or estimates. Through implicit cooperation agents can reach a common picture of the environment, but decision-making is locally performed (without consulting other agents). The work in [47] exemplifies implicit cooperation. There a set of mobile sensors coordinate their movements to dynamically collect the most informative measurements. This approach requires that agents share all observations to create a common view of the environment and support local decision making.

5. AGENT-BASED DEVELOPMENT

In this section we review the contributions of MAS research to the development of SNs. The paradigm of agent-oriented software engineering (AOSE) is argued to be well suited for developing complex software systems in distributed and dynamic environments [13]. Following [13, 81], it allows to build inter-operable and reusable software components to enable a flexible, robust and extensible infrastructure. Therefore, AOSE appears as a promising approach to support the development of SNs. In what follows we review how MAS research is contributing to programming SNs.

In order to program SNs, different middleware architectures have been proposed [14, 15]. The term middleware refers to some software layer lying between the application layer and the (hardware) layer of sensors. Its main purpose is to support the development, deployment, execution, and maintenance of sensor-based applications [16]. We address the reader to [16] and [14] for interesting discussions on the differences between middleware for SNs and middleware for more traditional systems.

Therefore, SN programmers can implement their applications on top of some out-of-the-box middleware for SNs. However, agent technology has explored several approaches to develop agent-based sensor applications, namely: (i) the development of a multi-agent application on top of some middleware for sensor networks (e.g. [17]); (ii) the development of agent-based middlewares that support agent-based programming (e.g. [18, 19, 20, 21, 22]); and (iii) the extension of an existing multi-agent platform with middleware capabilities for sensors (e.g. [23] extends the existing agent framework EMAA [24], whereas [25] extends Agentscape [26]). At this point, we observe that there is no consensus on the use of a particular agent platform (for instance JADE [81]), to develop either an agent-based middleware or an agent-based sensor application. However, there is an interesting comparison in [18], where Cougaar [82] is preferred to JADE because of the availability of semantic technologies, its robustness, and its scalability, among other reasons.

Although MAS research has produced a significant number of agent-based middlewares, a fundamental issue still requires further work: to support self-* capabilities[12] so that a SN can autonomously achieve its system-wide goals despite environment changes and failures. Indeed, as early anticipated in [16], since sensors must operate unattended, a middleware for SNs (unlike middleware for more traditional systems) has to provide new levels of support that guarantee autonomous behaviour.

An important feature of an agent-based middleware is the capability of generating new (higher-level) information from pure sensor data [13]. Such capability is fundamental to found the advanced reasoning required by agents that coordinate and adapt. Along this line, semantic technologies are key. Although semantic technologies offer tools to reason with the aid of ontologies, as noticed in [18] there are no mature tools available to create a sensor model and no query languages exist. This is precisely the purpose of the Sensor Model Language (SensorML). SensorML provides standard models and an encoding (through meta-data) for describing sensors along with their measurement processes and instructions for deriving higher-level information from observations. Processes encoded in SensorML can be discovered and executed. Thus, modelling languages for sensors such as SensorML are key to enable the semantic treatment of sensor data.

Several authors have also built agent-based middleware that support mobile agents (e.g. Agilla [19], ActorNet [20], MAPS [22]). A mobile agent is a software agent capable of migrating from one computing node to another to fulfil its tasks. Mobile agents are of interest to implement core (data-intensive) services such as data aggregation/fusion/dissemination and query-based information retrieval. Moreover, they are also valuable to dynamically deploy new services through code dissemination as illustrated by the Ambiance platform [84] that we describe next.

The above-mentioned Actornet middleware [20] founds the development of Ambiance [84], a platform that supports the high-level programming of SNs in Ambient Intelligence domains. Ambiance allows on-the-fly creation and execution of high-level programs written by non-professional programmers. Such programs are transformed into a system of mobile agents (platform-specific code for ActorNet), which are dynamically deployed and executed. Hence, agent technology becomes here the enabling technology of an alternative programming paradigm for sensor networks.

Finally, notice that agent-based middlewares have contributed to the seamless integration of SNs with further technologies (technology convergence), particularly with services (e.g. with SOA-based services [23]) and Radio Frequency Identification (RFID) technology [18, 21, 85, 86]. This effort is important to abstract agents from all low-level technical details trough some unified access to heterogeneous sensors, services, and RFID tags.

6. CONCLUSIONS

Despite the many significant, promising contributions of MAS research to the different research topics identified in this survey, more research effort is required to allow the application of sensor networks to real-world problems. In what follows we analyse the lessons learned by MAS research so far. Moreover, we also identify some open issues that pose challenging problems that deserve further research.

There are several lessons we have learnt after analysing the contributions of MAS research. Firstly, most results can only cope with small and medium SNs. Secondly, MAS research has mainly focused on the coordination and control of sensors. Thus, adaptiveness in a SN has received less attention, though central to its autonomous behaviour. Thirdly, although MAS research has traditionally considered self-interested agents, there are few contributions focusing on SNs with

[12] Systems capable of managing themselves are known as self-* systems [83], where the * sign indicates a variety of properties: self-organisation, self-configuration, self-diagnosis, self-repair, etc.

[13] In fact, Henricksen et al.[14] identify this feature as important to any SN middleware.

multiple owners. Finally, although there is a significant number of agent-based middlewares to develop agent-based sensor applications, such development still remains highly challenging. Hereafter, we analyse how future research can help overcome the problems raised above.

Firstly, we consider the scalability scope of current MAS research. Because of resource-boundedness, the applicability of MAS techniques to SNs is restricted to low-overhead [63] algorithms. These support that agents use limited communication and computation to make their decisions at anytime with available local knowledge. Hence, MAS researchers have to sacrifice optimality and adopt bounded-optimality algorithms so that agents behave as well as possible given their resources. As MAS research has recently started to explore [66], such algorithms must be approximate algorithms that provide guarantees on the quality of their solutions. Nonetheless, more work is required to design algorithms whose quality guarantees are accurate enough.

Notice that although approaches that cope with information uncertainty or non-myopic agent behaviours employ approximate algorithms, the computational cost of the optimisations that they require is still an Achilles heel. That is the case of the decentralised MDP-based algorithms, whose efficient approximate algorithms suffer from restricted applicability to SNs because of the inherent complexity of their decision-theoretical models. Therefore, an open issue is how to extend existing, low-overhead algorithms with these aspects without (substantially) incrementing their complexity. Some MAS contributions have started to make headway along this direction by extending low-overhead DCOP algorithms to deal with uncertainty (caused by an unknown environment) [67] and with non-myopic agent behaviour [68].

Secondly, regarding control, it is important that algorithms can operate on *any* distribution of control and knowledge to abide by the existing, physical distribution of control and knowledge in a SN. For instance, current DCOP approaches make strong assumptions about the distribution of control and knowledge (relations in DCOP terms). Vinyals et al. [87] have started to make headway along this direction by proposing Action-GDL, a DCOP algorithm that can run on any distribution of knowledge.

Regarding adaptiveness, MAS researchers have designed many adaptive, robust, self-stabilising protocols that allow agents to adapt to the underlying changes in the network and the environment. Nonetheless, an open issue in SNs is the design of *efficient* adaptation methods *sensitive* to the magnitude of changes: the smaller the changes, the less the effort to adapt. For instance, consider a highly dynamic SN. Although the network and the environment may continuously change, changes are locally limited. Therefore, approaches that can reuse previous computations and already acquired knowledge to efficiently adapt are of interest. At this aim, MAS can benefit from transfer learning techniques (e.g. [88]) to allow agents to pass agents on what they learn in former problems.

Another open issue concerning adaptiveness, identified in the context of organisational approaches [36], considers how to adapt MAS control regimes to changes in some underlying SN. The organisational structure used by a MAS is crucial for its effectiveness and efficiency. Moreover, the optimal structure may vary depending on the underlying SN [89, 90]. This problem involves challenging issues such as learning to adapt the degree of distribution, the number of organisational levels, or the degrees of agent autonomy. Thus, although some works [91, 56, 92] have already started to explore this line, researchers should put more effort on the applicability of these approaches to SN.

Regarding the treatment of multiple owners, although economic-based approaches such as markets or computational mechanism design (CMD) have been employed with some success to deal with self-interested agents, they pose some computational problems that prevent them from coping with large scale SNs. On the one hand, the need for a single auctioneer introduces a single point of failure. On the other hand, the optimisation problems that an auctioneer must solve are very costly. We foresee the need for distributed mechanisms that overcome these problems. Distributed Algorithmic Mechanism Design (DAMD) [93] studies how to design incentive compatible mechanisms with tractable computation in a distributed setting. DAMD takes into account communication overhead in addition to the local computational burden on agents. Therefore, the design of distributed mechanisms is worth exploring. Another interesting strand of research would be to blend DCOP algorithms with CMD, along the lines of the M-DPOP algorithm presented in [94]. As a major benefit DCOP algorithms would be able to consider self-interested agents.

Finally, regarding the development of agent-based sensor applications, we agree with [84], which argues that programming SNs should ideally narrow down to specifying high-level programs. However, to realise this approach it is necessary that agent-based middlewares support self-* capabilities. Indeed, as early anticipated in [16], since sensors must operate unattended, a middleware for SNs (unlike middleware for more traditional systems) has to provide new levels of support that guarantee autonomous behaviour.

ACKNOWLEDGEMENTS

This work has been funded by projects IEA (TIN2006-15662-C02-01), Agreement Technologies (CONSOLIDER CSD2007-0022, INGENIO 2010) and EVE (TIN2009-14702-C02-01,TIN2009-14702-C02-02). Meritxell Vinyals is supported by the Spanish Ministry of Education (FPU grant AP2006-04636).

REFERENCES

[1] Chong, C. and Kumar, S. (2003) Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, **91**, 1247–1256.

[2] Culler, D., Estrin, D., and Srivastava, M. (2004) Guest Editors' Introduction: Overview of Sensor Networks. *Computer*, **37**, 41–49.

[3] Lesser, V., Ortiz, C., and Tambe, M. (2003) *Distributed Sensor Networks: A Multiagent Perspective.* Springer.

[4] Tynan, R., O'Hare, G. M. P., and Ruzzelli, A. G. (2006) Multi-agent system methodology for wireless sensor networks. *Multiagent and Grid Systems*, **2**, 491–503.

[5] De Wolf, T. and Holvoet, T. (2003) Towards autonomic computing: agent-based modelling, dynamical systems analysis, and decentralised control. *INDIN 2003*, Aug., pp. 470–479.

[6] Xu, N. *A survey of Sensor Network Applications.* `http://courses.cs.tamu.edu/rabi/cpsc617/resources/sensor%20nw-survey.pdf`.

[7] Nagel, D. J. *Wireless Sensor Systems and Networks: Technologies, Applications, Implications and Impacts.* `http://intranet.daiict.ac.in/~ranjan/isn2005/papers/APP/wireless.pdf`.

[8] Rogers, A., Corkill, D., and Jennings, N. R. (2009) Agent technologies for sensor networks. *IEEE Intelligent Systems*, **24**, 13–17.

[9] Farinelli, A., Rogers, A., Petcu, A., and Jennings, N. R. (2008) Decentralised coordination of low-power embedded devices using the max-sum algorithm. *AAMAS*, pp. 639–646.

[10] Corkill, D. D., Holzhauer, D., and Koziarz, W. (2007) Turn Off Your Radios! Environmental Monitoring Using Power-Constrained Sensor Agents. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 31–38.

[11] Rogers, A., Osborne, M., Ramchurn, S. D., Roberts, S., and Jennings, N. R. (2008) Information agents for pervasive sensor networks. *PerCom*, pp. 294–299. IEEE Computer Society.

[12] ERCIM (2009). The sensor web. ERCIM News 76.

[13] Jennings, N. R. (2000) On agent-based software engineering. *Artificial Intelligence*, **177**, 277–296.

[14] Henricksen, K. and Robinson, R. (2006) A survey of middleware for sensor networks: state-of-the-art and future directions. *MidSens*, pp. 60–65.

[15] Molla, M. M. and Ahamed, S. I. (2006) A survey of middleware for sensor network and challenges. *ICPP Workshops*, pp. 223–228.

[16] Römer, K., Kasten, O., and Mattern, F. (2002) Middleware challenges for wireless sensor networks. *Mobile Computing and Communications Review*, **6**, 2002.

[17] Thoelen, K., Michiels, S., Joosen, W., Vangheluwe, K., and Verbeeck, K. (2009) A sensor middleware and agent-based communication platform for supply-chain management. *AAMAS*, pp. 1409–1410.

[18] Seitz, C., Schöler, T., and Neidig, J. (2009) An agent-based sensor middleware for generating and interpreting digital product memories. *AAMAS*, pp. 61–68.

[19] Fok, C.-L., Roman, G.-C., and Lu, C. (2005) Mobile agent middleware for sensor networks: an application case study. *IPSN*, pp. 382–387.

[20] Kwon, Y., Sundresh, S., Mechitov, K., and Agha, G. (2006) Actornet: an actor platform for wireless sensor networks. *AAMAS*, pp. 1297–1300.

[21] Bade, D. (2009) Towards an extensible agent-based middleware for sensor networks and RFID systems. *Proceedings of the Third International Workshop on Agent Technology for Sensor Networks (ATSN-09).*

[22] Aiello, F., Fortino, G., Guerrieri, A., and Gravina, R. (2009) Maps: A mobile agent platform for wsns based on java sun spots. *Proceedings of the Third International Workshop on Agent Technology for Sensor Networks (ATSN-09).*

[23] yuan Sheu, R., Czajkowski, M., and Hofmann, M. (2007) Adaptive peer-topeer agent sensor networks. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 39–46.

[24] Hofmann, M., Chacon, D., Mayer, G., Whitebread, K., and Hendler, J. Cast agents: Network-centric fires unleashed. *Proceedings of National Fire Control Symposium*, Lihue, Kauai, Hawaii, pp. 12–30.

[25] Harman, T., Padget, J., and Warnier, M. (2009) A multi-layered semantics-ready sensor architecture. *Proceedings of the Third International Workshop on Agent Technology for Sensor Networks (ATSN-09).*

[26] Agentscape operating system. Available on `http://www.iids.org/research/aos/index_html`.

[27] Langendoen, K. and Reijers, N. (2003) Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, **43**, 499–518.

[28] Niculescu, D. (2004) Positioning in ad hoc sensor networks. *Network, IEEE*, **18**, 24–29.

[29] Ferranti, E. and Trigoni, N. (2009) Practical issues in deploying mobile agents to explore a sensor-instrumented environment. Technical Report RR-09-02. Oxford University Computing Laboratory.

[30] Akkaya, K. and Younis, M. (2005) A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, **3**, 325–349.

[31] Jiang, Q. and Manivannan, D. (2004) Routing protocols for sensor networks. *CCNC*, pp. 93–98.

[32] Dash, R., Jennings, N., and Parkes, D. (2003) Computational-mechanism design: A call to arms. *IEEE Intelligent Systems*, **18**, 40–47.

[33] Rogers, A., David, E., and Jennings, N. R. (2005) Self-organized routing for wireless microsensor networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, **35**, 349–359.

[34] Padhy, P., Dash, R., Martinez, K., and Jennings, N. (2006) A utility-based sensing and communication model for a glacial sensor network. *AAMAS*, pp. 1353–1360.

[35] Zafar, H., Lesser, V. R., Corkill, D. D., and Ganesan, D. (2008) Using organization knowledge to improve routing performance in wireless multi-agent networks. *AAMAS*, pp. 821–828.

[36] Lesser, V. Scaling AI through multi-agent organizations. Invited talk. IJCAI-09 Award for Research Excellence.

[37] Corkill, D. D. (2008) Reporting Down Under. A CNAS (Collaborative Network for Atmospheric Sensing) Update. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 25–32.

[38] Damla Turgut, L. B. (2008) Three heuristics for transmission scheduling in sensor networks with multiple mobile sinks. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 1–8.

[39] Bölöni, L. and Turgut, D. (2008) Should i send now or send later? a decision-theoretic approach to transmission scheduling in sensor networks with mobile sinks. *Wireless Communications and Mobile Computing*, **8**, 385–403.

[40] Manyika, J. and Durrant-Whyte, H. (1995) *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Prentice Hall PTR Upper Saddle River, NJ, USA.

[41] Waldock, A. and Nicholson, D. (2007) Cooperative Decentralised Data Fusion Using Probability Collectives. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 47–53.

[42] Waldock, A., Nicholson, D., and Rogers, A. (2008) Cooperative Control using the Max-Sum Algorithm. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 65–70.

[43] Makarenko, A. and Durrant-Whyte, H. (2006) Decentralized Bayesian algorithms for active sensor networks. *Information Fusion*, **7**, 418–433.

[44] Rogers, A., Kho, J., Dash, R. K., and N.R.Jennings (2007) Gaussian Process Information Measures for Wireless Sensor Networks. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 1–2.

[45] Kho, J., Rogers, A., and Jennings, N. (2009) Decentralised control of adaptive sampling in wireless sensor networks. *ACM Transactions on Sensor Networks*, **5**, article 19–(35 pages).

[46] Osborne, M. A., Roberts, S. J., Rogers, A., Ramchurn, S. D., and Jennings, N. R. (2008) Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes. *IPSN*, pp. 109–120.

[47] Ruben Stranders, N. J., Alex Rogers (2008) A Decentralized, On-line Coordination Mechanism for Monitoring Spatial Phenomena with Mobile Sensors. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 9–15.

[48] Coates, M. (2004) Distributed particle filters for sensor networks. *IPSN*, pp. 99–107.

[49] Rosencrantz, M., Gordon, G., and Thrun, S. (2003) Decentralized sensor fusion with distributed particle filters. *UAI*, pp. 493–500.

[50] Paskin, M., Guestrin, C., and McFadden, J. (2005) A robust architecture for distributed inference in sensor networks. *IPSN*, pp. 55–62.

[51] Bychkovskiy, V., Megerian, S., Estrin, D., and Potkonjak, M. (2003) A collaborative approach to in-place sensor calibration. *IPSN*, pp. 301–316.

[52] Rogers, A., Dash, R. K., Jennings, N. R., Reece, S., and Roberts, S. (2006) Computational mechanism design for information fusion within sensor networks. *9th Int. Conf. on Information Fusion*, pp. 1–7.

[53] Dash, R., Rogers, A., Reece, S., Roberts, S., and Jennings, N. R. (2005) Constrained bandwidth allocation in multi-sensor information fusion: A mechanism design approach. *The Eighth International Conference on Information Fusion (Fusion 2005)*.

[54] De Roure, D. (2005) Floodnet: A new flood warning system. *Ingenia*, pp. 49–51.

[55] Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J. A., Phelps, S., Rodríguez-Aguilar, J. A., and Sousa, P. (2006) Issues in multiagent resource allocation. *Informatica (Slovenia)*, **30**, 3–31.

[56] Sims, M., Goldman, C., and Lesser, V. (2003) Self-organization through bottom-up coalition formation. *AAMAS*, pp. 867–874.

[57] Smith, R. G. (1981) The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, **C-29**, 1104–1113.

[58] Ostwald, J., Lesser, V., and Abdallah, S. (2005) Combinatorial auction for resource allocation in a distributed sensor network. *RTSS*, pp. 266–274.

[59] Johnson, M. P., Rowaihy, H., Pizzocaro, D., Bar-Noy, A., Chalmers, S., Porta, T. F. L., and Preece, A. D. (2008) Frugal sensor assignment. *DCOSS*, pp. 219–236.

[60] Le, T., J.Norman, T., and Vasconcelos, W. (2009) Agent-based Sensor-Mission Assignment for Tasks Sharing Assets. *In Proceedings of the Third International Workshop on Agent Technology for Sensor Networks (ATSN-09)*.

[61] Modi, P. J., Shen, W.-M., Tambe, M., and Yokoo, M. (2005) Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.*, **161**, 149–180.

[62] Béjar, R., Domshlak, C., Fernández, C., Gomes, C. P., Krishnamachari, B., Selman, B., and Valls, M. (2005) Sensor networks and distributed csp: communication, computation and complexity. *Artif. Intell.*, **161**, 117–147.

[63] Zhang, W., Wang, G., Xing, Z., and Wittenburg, L. (2005) Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artif. Intell.*, **161**, 55–87.

[64] Petcu, A. and Faltings, B. (2005) A scalable method for multiagent constraint optimization. *IJCAI*, pp. 266–271.

[65] Mailler, R. and Lesser, V. R. (2004) Solving distributed constraint optimization problems using cooperative mediation. *AAMAS*, pp. 438–445.

[66] Pearce, J. P. and Tambe, M. (2007) Quality Guarantees on k-Optimal Solutions for Distributed Constraint Optimization Problems. *IJCAI*, pp. 1446–1451.

[67] Jain, M., Taylor, M., Tambe, M., and Yokoo, M. (2009) Dcops meet the real world: Exploring unknown reward matrices with applications to mobile sensor networks. *IJCAI*, pp. 181–186.

[68] Stranders, R., Farinelli, A., Rogers, A., and Jennings, N. (2009) Decentralised coordination of mobile sensors using the max-sum algorithm. *IJCAI*.

[69] Nair, R., Varakantham, P., Tambe, M., and Yokoo, M. (2005) Networked Distributed POMDPs: A Synthesis

of Distributed Constraint Optimization and POMDPs. *AAAI*, pp. 133–139.

[70] Varakantham, P., Marecki, J., Yabu, Y., Tambe, M., and Yokoo, M. (2007) Letting loose a spider on a network of pomdps: generating quality guaranteed policies. *AAMAS*, pp. 1–8.

[71] Marecki, J., Gupta, T., Varakantham, P., Tambe, M., and Yokoo, M. (2008) Not all agents are equal: scaling up distributed pomdps for agent networks. *AAMAS*, pp. 485–492.

[72] Kumar, A. and Zilberstein, S. (2009) Constraint-based dynamic programming for decentralized pomdps with structured interactions. *AAMAS*, pp. 561–568.

[73] Horling, B. and Lesser, V. (2005) Analyzing, modeling and predicting organizational effects in a distributed sensor network. *Journal of the Brazilian Computer Society*, **11**, 9–30.

[74] Abdallah, S. and Lesser, V. (2004) Organization-based coalition formation. *AAMAS*, pp. 1296–1297.

[75] Ruairí, M. and Keane, M. (2007) The Dynamics Regions Theory: Role Based Partitioning for Sensor Network Optimization. *In AAMAS Workshop on Agent Technology for Sensor Networks*, pp. 25–30.

[76] Weiss, G. (1995) Distributed reinforcement learning. *Robotics and Autonomous Systems*, **15**, 135–142.

[77] Tham, C.-K. and Renaud, J. (2005) Multi-agent systems on sensor networks: A distributed reinforcement learning approach. *ISSNIP*, pp. 423–429.

[78] Sandhu, J. S. (2004) Wireless sensor networks for commercial lighting control: Decision making with multi-agent systems. *In AAAI Workshop on Sensor Networks*, pp. 131–140.

[79] Kok, J. R. and Vlassis, N. A. (2006) Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, **7**, 1789–1828.

[80] Guestrin, C., Koller, D., and Parr, R. (2001) Multiagent planning with factored MDPs. *NIPS*, pp. 1523–1530.

[81] Bellifemine, F. L., Caire, G., and Greenwood, D. (2007) *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. Wiley & Sons.

[82] Force, C. Cougar. Available on `http://cougaar.org/`.

[83] Luck, M., McBurney, P., Shehory, O., and Willmott, S. (2005) *Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing.* Agentlink III.

[84] Razavi, R., Mechitov, K., Sundresh, S., Agha, G., and Perrot, J.-F. (2006) Ambiance: adaptive object model-based platform for macroprogramming sensor networks. *OOPSLA Companion*, pp. 663–664.

[85] Vrba, P., Macrek, F., and Mařík, V. (2008) Using radio frequency identification in agent-based control systems for industrial applications. *Eng. Appl. Artif. Intell.*, **21**, 331–342.

[86] Cui, J. F. and Chae, H. S. (2007) Developing Load Balancing System for RFID Middlewares Using Mobile Agent Technology. *KES-AMSTA*, pp. 757–764.

[87] Vinyals, M., Rodriguez-Aguilar, J. A., and Cerquides, J. (2009) Generalizing DPOP: Action-GDL, a new complete algorithm for DCOPs. *In proceedings of AAMAS Workshop on Optimisation in Multi-Agent Systems.*

[88] Abdallah, S. and Lesser, V. (2007) Multiagent reinforcement learning and self-organization in a network of agents. *AAMAS*, pp. 172–179.

[89] Horling, B. and Lesser, V. (2005) Analyzing, modeling and predicting organizational effects in a distributed sensor network. *Journal of the Brazilian Computer Society, Special Issue on Agents Organizations*, **11**, 9–30.

[90] pa So, Y. and Durfee, E. H. (1996) Designing tree-structured organizations for computational agents. *Computational and Mathematical Organization Theory*, **2**, 219–246.

[91] Sims, M., Corkill, D., and Lesser, V. (2008) Automated Organization Design for Multi-agent Systems. *Autonomous Agents and Multi-Agent Systems*, **16**, 151–185.

[92] Kota, R., Gibbins, N., and Jennings, N. R. (2009) Self-organising agent organisations. *AAMAS*, pp. 797–804.

[93] Sami, R. (2003) Distributed Algorithmic Mechanism Design. PhD thesis Yale University.

[94] Petcu, A. (2009) *A Class of Algorithms for Distributed Constraint Optimization*, Frontiers in Artificial Intelligence and Applications, **194**. IOS Press.