When Wireless Sensor Networks Meet Robots

Giuseppe Amato*, Mathias Broxvall§, Stefano Chessa‡, Mauro Dragone†, Claudio Gennaro*,
Claudio Vairo*
**ISTI-CNR*
†*University College of Dublin*
‡*Università di Pisa*
§*Örebro Universitet*
Email:Giuseppe.Amato@isti.cnr.it,mb1@aass.oru.se,ste@di.unipi.it,Mauro.Dragone@ucd.ie,
claudio.gennaro@isti.cnr.it, claudio.vairo@isti.cnr.it

***Abstract*—Enabling integrated robots and Wireless Sensor Network (WSN) applications is an important and extended challenge for both robotics and WSN research & development and a key enabler for a range of advanced hybrid applications, such as environmental monitoring and Ambient Assisted Living (AAL). This paper describes a work-in-progress WSN/robots communication framework that is being purposefully built to facilitate the constructions of robotic ecologies, i.e. networks of heterogeneous computational nodes interfaced with sensors, effectors and mobile robot devices. This paper discusses a number of requirements characterizing this type of systems and illustrates how they are being addressed in the design of the new communication framework.**

***Keywords*-Robots; Wireless Sensor Networks.**

I. INTRODUCTION

A WSN is a wireless network composed by a number of sensors, each of which is an autonomous microsystem capable of sensing a number of environmental parameters (depending on the nature and number of transducers it embeds) and of locally processing and storing sensed data. While WSNs are perfect in monitoring the environment and detecting what is happening in it, they are very limited in reacting to what they detect. Robots, on the other hand, can act as interfaces to WSN solutions and also enhance them by providing important benefits such as sensor deployment, calibration, failure detection and power management.

On the other hand, developing integrated robots & WSN applications has the potential to solve many problems that hinder the spread of pure robotics solutions; in particular, the difficulty of understanding their environment with noisy and imprecise sensor capabilities. In contrast, integrated robot & WSN solutions advocate the augmentation of the robots' communication and interaction capabilities with those afforded by the sensors and services embedded within the environment.

The work described in this paper, which is a work in progress, is being conducted within the context of the EU FP7 project RUBICON, (Robotic UBIquitous COgnitive Network). The project will develop a self-sustaining, adaptive robotic ecology consisting of mobile robotic devices, sensors, effectors and appliances cooperating to perform complex tasks such as supporting an older person to live independently. These components will encourage and teach one another in order to achieve their goals more efficiently and to adapt to changing requirements and user's needs. This will reduce the need for pre-programming and human supervision, and so will make these systems much cheaper and simpler to deploy in a variety of applications, for different homes and users.

One of goals of the RUBICON project is the integration of robot and WSN technologies in order to test them in a real AAL environment. The system will use the network of wireless sensors and actuators, supplemented by a mobile robot, to learn to better recognize situations and activities in the AAL scenario.

Practical implementation of RUBICON's outcomes could be used for:

- Assisting the user in their daily living (e.g. closing the blinds when the user is sleeping).
- Gathering data for post process analysis.
- Monitoring of people at risk.
- Collecting information that will assist in clinical assessments.
- Identifying and alerting relevant stakeholders of potentially dangerous behavior and/or situations.

While the emphasis of our extended research is to provide learning solutions for this type of systems, in order to support our vision we need flexible communication capabilities able to connect components across different nodes and allow sharing of data and learning information, while changing communication path-ways in response to changing circumstances, due to mobility, network disruptions, failures, etc.

Supporting varying computational constraints is a primary priority, as target environments will contain devices such as computers with large processing and bandwidth capacities, as well as much simpler devices such as micro-controller-based actuators and sensor nodes, and even devices with no (customizable) computational capability at all, such as Radio Frequency Identifications (RFIDs).

While existing robot/WSN combined approaches investigate many related issues such as cooperative monitoring, localization and navigation, they often rely on ad-hoc communication mechanisms that usually lack a broader applicability and that do not fully embrace the robotic ecology concept. For instance, most of the existing solutions adopt a centralized integration approach, or apply a data-centric perspective, in which WSNs are treated as just another input to traditional control architectures.

In contrast, we are building a communication framework on top of state of the art robotic and WSN middleware that is purposefully designed to address the characteristic requirements dictated by robotic ecology solutions.

Noticeably, our communication framework is general and it allows the building of several different applications all involving the integration of the moving and actuation capabilities of autonomous mobile robots and the sensing and interacting capabilities of WSN nodes.

The remainder of the paper is organized in the following manner: Section II provides an overview of the most significant robot/WSN integration approaches attempted in past research, focusing on how they have addressed the communication requirements. Section III discusses the requirements and the high-level design of our new communication framework. Section IV details the layered architecture of the framework. Finally, Section V summarizes the contributions of this paper and discusses our plans for future work.

II. RELATED WORK

There are many example of related work combining mobile robots with wireless sensor networks. The latter are usually used to report events that need further investigation and intervention by the robots in the environment whereas robots' mobility helps the WSN to monitor and operate in a larger area than is possible with fixed sensor deployments. For instance, a Mobile Robot is used in [1] to collect sensed data from a WSN in order to prolong the lifetime of the sensor nodes, and also to reduce the hop count cost, when the WSN is partitioned in islands.

Mohammad Rahimi et al. [2] studied the feasibility of extending the lifetime of a wireless sensor network by exploiting mobile robots that move in search of energy, recharge, and deliver energy to immobile, energy-depleted nodes.

With the PlantCare project, [3] have demonstrated how a robot can be used to deploy and calibrate sensors, detect and react to sensor failure, deliver power to sensors, and otherwise maintain the overall health of the wireless sensor network. Navigation strategies employing WSNs usually rely on the fact that the positions of all network nodes are well known or can be inferred. The solutions for the localization problem often employ RSSI readings, which are well documented as unreliable in dynamic environments, to determine node or robot positions [4] , often as part of Robot SLAM (Simultaneous Localization and Mapping) solutions [5].

Batalin [6] addresses the problem of monitoring spatiotemporal phenomena at high fidelity in an unknown, unstructured, dynamic environment. The robot explores the environment, and based on certain local criteria, drops a node into the environment, from time to time. Sensor nodes act as signposts for the robot to follow, thus obviating the need for a map or localization on the part of the robot.

All these solutions for the integration of WSN and mobile robotics usually are developed to solve specific problems in specific scenarios. However, a number of research initiatives have tackled the creation of generic communication frameworks to be used within the robot/WSN application domain.

Gil et al. [7] describes a data-centric middleware for wireless sensor networks in the scope of the European project AWARE. The middleware implements a high-level abstraction for integration of WSNs with mobile robots. This is achieved by providing data-centric access to the information gathered by the wireless sensor network, which includes mobile robotic nodes. Nodes in the network organize themselves to retrieve the information needed by the robots while minimizing the number of transmitted packets in order to save energy. Robots are connected via a high-bandwidth IEEE 802.11 WiFi network and interact with the low-bandwidth IEEE 802.15.4 WSN via a Gateway. The Gateway is in turn connected to both networks and used to collect the data gathered within the WSN.

The work of [8] focus in deploying mobile robots on environments already monitored by unstructured WSNs, for instance, in applications, such as search and rescue, where the robots must rely solely on this network for control and communication purposes. To this end, the mobile robots are equipped with sensor nodes and are capable of communicating with the WSN, which is used, not only to read WSN data, but also to access and control the robots. The resulting communication framework addresses bandwidth, message size and route restrictions by using adapter components to enable the communication of robot control messages through the WSN.

The RUBICON Communication Layer builds upon the PEIS middleware previously developed as part of the Ecologies of Physically Embedded Intelligent Systems project [PEIS] to provide a de-centralized mechanism for collaboration between separate processes running on separate devices. The PEIS middleware allows for automatic discovery, dynamic establishment of P2P networks, self-configuration and high-level collaboration in robotic ecologies through subscription–based connections and a tuplespace communication abstraction.

PEIS communicates by publishing information as tuples consisting of a <key, data> pair, together with various pieces of meta-information such as time-stamps, creator etc. Through a search mechanism, any PEIS can find and consume the relevant tuples produced by any other PEIS, which allows for an expressive and flexible communication model. This tuple space thus constitutes a distributed database into which any PEIS-component can read and write information regarding any PEIS-component.

III. COMMUNICATION LAYER

The main objective of Communication Layer is to provide different type of communicating mechanisms for exchanging information between applications running on remote devices (robot, pc, motes, etc) of the Rubicon ecology. In particular, the Communication Layer will make available different paradigms of communication on the basis of the type of hardware involved in the communication, described by the following requirements.

A. Communication Requirements

- **Sensing**. In order to build and maintain an up-to-date picture of the state of the robotic ecology and its environment, and to enable collaboration between members of the robotic ecology (e.g. communication of localization data from the ceiling camera to the robot), the applications must be able to receive data and periodic status updates from every sensor and actuator it wishes for. In order to support reliability, the applications should also be able to specify the desired update rate and to be informed of the maximum latency to be expected by the resulting updates. The applications must tolerate the loss of some of these updates but all data must be time stamped in order to be able to ignore old updates.
- **Actuation**. The applications must be able to send control instructions (e.g. new set points, new output values) to every actuator it wishes for. For this type of transmission, the applications do not require the ability to communicate periodic updates of control instructions. However, in order to support reliability, transmission of control instructions should be reliable (acknowledged). In addition, the applications need to be informed of the maximum expected latency.
- **Data Sharing**. The applications must be able to (asynchronously) share it sensor data, actuator status and other information among distributed nodes (multiple robots, WSN nodes and other devices).
- **Messages**. In order to co-ordinate their operation across distributed nodes, the applications must be able to send reliable and synchronous control messages to all the nodes it wishes for.
- **Discovery**. The applications need an updated picture of all the components available in the system, including all the WSN nodes currently active. Every component should have a unique ID and the Control Layer should be informed whenever any robotic device or WSN nodes join (as they become operative and connect to the network), or leave the system (as they get disconnected, breaks, they battery get depleted or simply move out of network range

B. Architecture

In order to meet these requirements, the communication layer is mainly based on the two existing Software: the StreamSystem Middleware [9] and the PEIS Ecology middleware [10]. These two background technologies implement partly overlapping services but with important differences in hardware requirements and with services targeted towards applications for robotic devices and for distributed WSNs, respectively. The PEIS middleware provides automatic discovery, dynamic establishment of P2P networks, self-configuration and high-level collaboration in robotic ecologies through subscription–based connections and a *tuplespace* communication abstraction.

While the Tiny PEIS middleware kernel [11] is a specialization of the generic PEIS middleware to provide these high level robotic services also for WSN networks it requires significant WSN resources such as RAM memory and wireless bandwidth. Furthermore, due to the memory constraints and the single-hop only requirements of the Tiny PEIS on WSN nodes it is worthwhile to investigate a less restrictive approach for incorporating WSN nodes in the rich robotic middleware considered for the RUBICON project.

The Stream System framework provides a simple and effective access to the transducer and actuator hardware on wireless nodes, and a communication abstraction based on channels.

These two heterogeneous paradigms of communication are integrated by means of a star-topology (see Figure 1), in which a cluster of motes, called islands, will be interconnected by means of the PEIS-network. A special mote acting as sink node, connected via serial link with a *basestation* (a PC) will host a gateway component responsible for bridging the island with the tuplespace. In this way two basestations, which manage one island each, can communicate with each other through the PEIS middleware. By exploiting the P2P network of PEIS these devices can route messages originating in the islands to each other and forward such messages to the destination motes.

When a mobile robot comes within range of an island of motes, it can communicate with it through a local WSN mote carried by the robot capable of participating with the island as a peer and capable of forwarding messages and RSSI data to the robot. The sink node thus must be able to handle the dynamic connection and disconnection of a mobile robot mote. Furthermore the WSN must accommodate the possibility of more than one entry point of messages into / from the island. Note however that this does not pose any additional demands on the routing of outgoing message since any such messages can be propagated through the robot-based P2P network regardless of entry point.

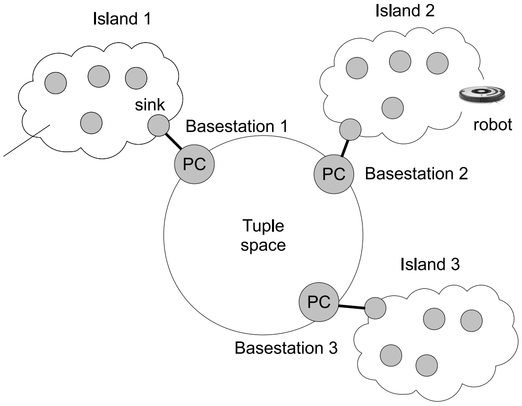

Figure 1. The Topology of the RUBICON Ecology

The communication layer provides transparent transfer of data between end users, providing reliable data transfer services to the upper layers. It controls the reliability of a given link through flow control, segmentation/desegmentation, and error control. Some protocols are state- and connection-oriented. This means that the communication layer can keep track of the segments and retransmit those that fail. It also provides the acknowledgement of the successful data transmission and sends the next data if no errors occurred.

IV. INTEGRATING ROBOTS AND WSN

A. Architecture

The system is divided into two conceptual entities: the group Robot-Sink and the WSN deployed in the environment. The sink mote is connected to the Robot by means of a serial interface and acts as gateway between the Robot and the WSN. In particular, it forwards to the WSN the commands coming from the Robot and it notifies to the Robot the information coming from the WSN, for example the RSSI of the sensors or the detection of a particular event interesting for the Robot.

The Robot can move around in the environment according to the information coming from the WSN.

The part of the system running on each of the sensors of the WSN is composed of three layers (see Figure 4): Application, Transport and Network. The Application layer can be programmed according to the application requirements. In general, it receives the requests coming from the sink and reacts as consequence. For example it can respond to a request of reading one or more transducers value, or to a request to activate an actuator on the mote.

```
command error_t send(dest, data, nbytes,
    seq, comp_id);
event void receive(src, data, nbytes,
    seq, comp_id);
```

Figure 2. Network Layer Interface.

```
command error_t send(dest, data, nbytes,
    reliable, comp_id);
event void receive(src, data, nbytes,
    comp_id);
event void ack(seq, comp_id);
```

Figure 3. Transport Layer Interface.

B. Network layer

The Transport layer provides an interface for receiving messages to be sent over the radio and to signal the reception of incoming messages (see Figure 2). The textttsend command takes in input the destination address to send the message, the pointer to the data buffer to be sent, the number of bytes of the message, the sequence number of the message, and the identifier of the transport component that is sending the message. The `comp_id` parameter is used to perform the multiplexing/demultiplexing of the messages. In particular, it is used to deliver the message to the right component upon receiving a new message.

The address of mote in the Ecology is formed by the address of the island and the address of the mote in the island. The address of the island is a unique number that corresponds to the peis-id of the basestation. It is codified with a byte, that is interpreted as un unsigned integer. The value 255 is used to indicate “this-island”. The address of the mote is a two bytes unsigned integer and is a unique number that corresponds to the TinyOS-ID. The value 65535 is used to indicate a broadcast inside the island, and the value 65534 is used to indicate “myself”.

A robots that wants to join an island of the WSN, is equipped with a mote connected via USB to the Robot computing unit. The motes allows the robot to communicate either directly to any mote of the island or to the sync of the island.

C. The PEIS Proxy

To simplify the integration of application dependent objects into already deployed RUBICON ecologies, the communication layer will also exploit the concept of *PEIS proxy* [11]. This allows one to customize the available hardware devices without any need for re-implementation of any of the robotic software already running in the ecology.

For each island node, the basestation creates a PEIS proxy component that emulates the behavior of the WSN mote. This proxy component is characterized by (a) appearing as a unique entity with a unique ID number in the PEIS ecology

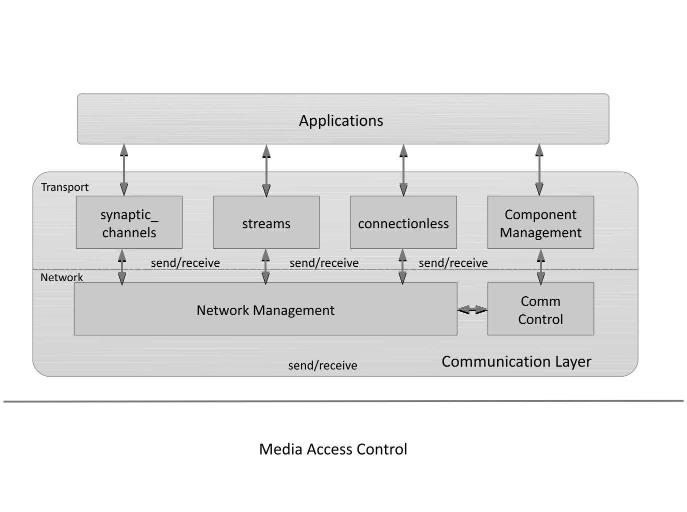

Figure 4. The architecture of the Communication Layer

and (b) it publishes and receives tuple data corresponding to the operations available to all WSN motes.

Initially this PEIS proxy node is populated by the basestation with all statistic information about the given node, relying on a static translation between the mote type and the functionalities that the gateway associated with mote type.

Whenever the WSN basestation receives updated sensor data from a mote, it translates these sensor data to the format of the PEIS tuplespace (most notably, in ascii format) and publishes it to the corresponding WSN proxy node. Furthermore, it subscribes to changes to any actuation tuples in the proxy node, and translates and sends actuation commands to the corresponding mote.

If multiple basestations are in range of the same mote, a failure is noted when they attempt to create the proxy for each node. This failure to create an additional node is regulated in the basic PEIS ecology framework and will lead to only one of the two basestation to be able to proxy the same WSN node.

D. Transport layer

The Transport layer provides a first abstraction for sending the messages (see Figure 3). In particular, it keeps track of the pending messages (by means of a sequence number that it adds to each message to be sent), it provides the feature of a reliable communication and it manages the ack of the reliable messages. It may offer additional services, such as connection-oriented service. In this case it is the responsible for allocating the data structures needed to the communication and to maintain them.

Here is a list of features that the communication layer provides to the WSN developers:

- *Synaptic Communication*: This type of communication is used exclusively by the Learning Layer and enables two ESNs (Echo State Network), running on different nodes, to exchange data. In particular it enables the transmission of the output of a set of neurons from a source ESN to a destination ESN.
- *Data Stream Communication*: This type of communication is used exclusively by the Control Layer and enables the point-to-point communication between two specific devices, typically for reading data from remote mote transducers.
- *Connectionless Message Passing*: This type of communication is used exclusively by the Control Layer and enables the point-to-point communication between two specific devices, typically for sending commands to remote mote-actuators.

Data Stream Communication is the paradigm of communication more relevant for the purposes of integration between robots and WSN. The stream represents a generic unidirectional data channel that is able to carry data records. In particular, we have three types of streams:

1) **Local Streams** represent local data channels where read and write operations must occur on the hosting sensor.
2) **Sensor Streams** are the basic abstraction for collecting readings from transducers. They can only be read by operators since the writing is carried out by the associated transducers (these can be thought of as virtual operators writing to sensor streams).
3) **Remote streams** require cooperation between two nodes since they intend to provide a data channel between two different nodes. Write operations can be carried out on one of them (the stream write-end) and read operations can take place on the other (the read-end).

Figure 5 illustrates these concepts.

Another important concept in command/event-based systems is that of split-phase operations. The call requesting to start the operation returns immediately, without waiting for the operation to complete.

V. CONCLUSION AND FUTURE WORK

The purpose of the RUBICON project is to create a robotic ecology comprising robots, a large number of heterogeneous environmental sensors and actuators, and learning and cognitive capabilities. In this ecology, the communication layer should provide almost homogeneous services to devices with very different communication and processing capacities. The approach chosen in RUBICON is to build communication services on top of established robotic and WSN middlewares.

This paper draws the status of the development of the communication layer, which is still in the early stage of development. In particular, based on the RUBICON communication requirements, it describes the overall architecture of

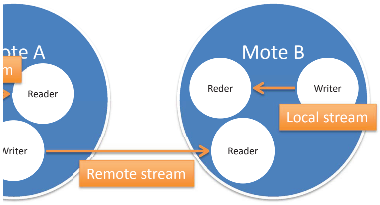

Figure 5. The paradigm of the data stream communication

the communication layer and its current status of development, and it discusses how the integration ROBOTS-WSN is achieved.

Next steps of the RUBICON communication layer developments, which will span the next 18 months, are the integration of specific communication mechanisms for connecting the RUBICON distributed learning and cognitive components, and its validation and performance evaluation.

One of the scenarios that we are going to test will exploit a data acquisition application that can be used by mobile robots to acquire sensor and radio signal strength index (RSSI) information to support their localization in indoor environments.

ACKNOWLEDGMENT

This work has been partially supported by the EU FP7 RUBICON project (contract n. 269914).

REFERENCES

[1] T.-C. Chen, T.-S. Chen, and P.-W. Wu, “On data collection using mobile robot in wireless sensor networks,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 41, no. 6, pp. 1213 –1224, nov. 2011.

[2] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, “Studying the feasibility of energy harvesting in a mobile sensor network,” in *Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on*, vol. 1, sept. 2003, pp. 19–24.

[3] A. LaMarca, W. Brunette, D. Koizumi, M. Lease, S. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello, “Making sensor networks practical with robots,” in *Pervasive Computing*, ser. Lecture Notes in Computer Science, F. Mattern and M. Naghshineh, Eds. Springer Berlin / Heidelberg, 2002, vol. 2414, pp. 615–622.

[4] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, “Range-only slam with a mobile robot and a wireless sensor networks,” in *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, may 2009, pp. 8 –14.

[5] G. Tuna, K. Gulez, and V. Gungor, “Communication related design considerations of wsn-aided multi-robot slam,” in *Mechatronics (ICM), 2011 IEEE International Conference on*, april 2011, pp. 493 –498.

[6] M. A. Batalin, “Symbiosis: cooperative algorithms for mobile robots and a sensor network,” Ph.D. dissertation, Los Angeles, CA, USA, 2005, aAI3180329.

[7] P. Gil, I. Maza, A. Ollero, and P. J. Marron, “Data centric middleware for the integration of wireless sensor networks and mobile robots,” in *Proceedings of the 7th Conference On Mobile Robots And Competitions*, Paderne, Portugal, 2007.

[8] L. de Souza, R. Padilha, and C. Decker, “Neural fault isolator for wireless sensor networks,” in *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on*, june 2008, pp. 47 –50.

[9] G. Amato, S. Chessa, and C. Vairo, “Mad-wise: A distributed stream management system for wireless sensor networks,” in *Software Practice & Experience, 40 (5): 431 - 451*, 2010.

[10] M. Broxvall, “A middleware for ecologies of robotic devices,” in *Proceedings of the 1st international conference on Robot communication and coordination*, ser. RoboComm ’07. Piscataway, NJ, USA: IEEE Press, 2007, pp. 30:1–30:8.

[11] M. Bordignon, J. Rashid, M. Broxvall, and A. Saffiotti, “Seamless integration of robots and tiny embedded devices in a peis-ecology,” in *Proc of the IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, San Diego, CA, 2007, pp. 3101–3106.