Changing control rules dynamically in embedded digital control systems

JAVIER VÁSQUEZ
University of Costa Rica, Computer Science Deparment
San José, Costa Rica
javier.vasquez@ucr.ac.cr
JOSE L. VÁSQUEZ
University of Costa Rica, Sede del Atlántico
Cartago, Costa Rica
jose.vasquez@ucr.ac.cr
CARLOS M. TRAVIESO
University of Las Palmas de Gran Canaria, Signals and Communications Department.
Las Palmas de Gran Canaria, España.
ctravieso@dsc.ulpgc.es, http://www.gpds.ulpgc.es

Abstract **:** This paper details a model developed at the University of Costa Rica to upgrade an Internet-based distributed measurement and control (DMC) system.
In this paper we present a model that enables us to separate the control rules from the embedded systems, besides assigning to each of it an IP address that allows interconnections and collaboration between them.
The model takes the NIST-DMC model and assigns over it a Business Layer, which is responsible to handle the interface with authorized users, who manage a set of control rules both dynamically as remotely. When a user modifies a rule, the Business Manager informs all the digital system involved that they must start to do some things in a different way.
By separating the control of the devices, the proposed model provides a broad longevity to the components of digital systems that implement it, provides greater flexibility to adapt the response of the devices with changing administrative interests, and promises a decrease in costs maintenance, since it reduces the need to modify the software embedded in the control device.

Key-words: dynamics rules, distributed control, embedded systems, NIST-DMC, greenhouse automation

1 Introduction

Advances in technology, reduced cost of microcontrollers and devices, such as sensors and actuators, and the proliferation of Internet connections had led to the appearance of digital system controllers (DSC) that are designed to monitoring and controlling equipment distributed by remote human intervention. These DSC consists of a series of sensors controlled commonly locally by a microcontroller, specifically a device is an entity capable of performing one or more tasks related to control (hardware - software) such as:

- View and control physical variables.
- Scheduling activities.
- Messaging service (e-mail, movil phone).

While devices (DSC) may work in a distributed manner, it is usual to have the controller intelligence permanently associated with one of the interconnected devices. For purposes of this paper we mean by *control* the attempt to impose predictability to unsafe entities that react to events. Also we understand *control policy* as the predefined set of events they are activated when so is determined by the control devices.

To carry out the distribution of control is proposed a distributed control model (DCM) that keeps separate control rules from control devices, hence any change in such policies do not necessarily alter the hardware or software used in such DSC. This method allows for incremental construction of more sophisticated control devices with increased useful life, whose flexible policies will be managed more in line with the future needs of customers. In the DCM the concept of control rule keeps control rules separated from actuators. The control rule approach uses a distributed applications platform based on client server architecture running on a TCP/IP network.

2 Background

In the late twentieth and early twenty-first has raised a great interest to the research on sensor networks, perhaps due to the confluence of three factors:

- Decreasing cost of technology needed to produce digital systems.
- Gradual increase in miniaturization and capabilities of such systems (which most often are wireless)
- Widespread use of embedded systems [2]

Some domains and potential uses are: everyday applications (heating, security, resource allocation, tracking cars and people), industrial manufacturing, transport applications (aviation, automobiles, power distribution, prevention and solution of traffic congestion), banking applications (management of debit and credit cards), health applications (medical monitors, medical diagnostics), monitoring (animals, tools, environmental conditions, events), environmental applications (diseases of crops, natural resources) and other [1, 2, 3, 4, 8, 11, 12].

Advances in technology determine that current limitations that have "motes"[1] will be irrelevant in a very short term, and also that we can avoid worrying too much by the interconnection of sensor networks reaching. In this regard, in [1] presents a "mote" design by Intel Company, called Stargate with tens of megabytes RAM, gigabytes of persistent memory and ability to interact with Internet. In [6] is introduced the use of chips of small size and large capacity for handling video and audio as well as its transmission in compact format, e.g., TC35273XB by Toshiba. Also is defined a processing architecture of three levels, processors near video cameras will be equipped with databases, filtering techniques, compression and pattern recognition. Devices that use these chips will have zoom function and will have turning capabilities of each camera and also shall be fitted with solar cells to become self sufficient. Similarly, in [13] is advocated because it gives local video processing in base stations to reduce the amount of data sent over the channel.

Regarding the management of control rules, in [3] is proposed the use of responsible control laws to ensure respect the preferences of users of sensor networks. The use of virtual machines to control management has been proposed in [10], those seeking to reduce the work of code conversion and thus the power consumption of devices. This virtual machine has 3 levels of abstraction *Handlers,* responding to system events; *Operations*, that are functional execution units and finally the *Code Capsules*, which are units of spreading code. The intercommunication between sensors has also arrived at structuring and delegating to agents, in [16] is proposed to handle sensor networks in equipment that could be handled by simple agents, regardless of the network could hypothetically be composed by thousands of sensors arranged hierarchically.

In [14] is proposed an architecture similar to that described in this article, e.g. use of Internet and "base stations", besides using Postgres to storage data. Similarly, in [5] is described a security application for a building and using a platform consisting of multiple motion sensors, sending messages through GSM and reception in a PDA, using the UDP protocol for sending data and the TCP protocol to send administrative information.

In order to adapt and update applications dynamically, in [11] is proposed the use of an intermediate layer (*middleware*) which releases the hardware from user needs. It is in this layer where they intend to be given the automatic adjustment of parameters that simplify distributed processing of data from mobile wireless sensor. It is evident that the middleware has been given a very poor use, and that is only used essentially for parameterization. In [7] is proposed the use of an intermediate layer, to address each sensor, based on the priority of it at the time of the event.

Like previous efforts, NIST-DMC model [15] proposed in 1999 by the U.S. Commerce Department to control distributed administration makes a clear difference between what the systems must perform and how to do so. This leads to rigid patterns of control in the area of monitoring, which once defined the policies, they are assembled and distributed to devices, so any changes to these policies, involves modifying the

[1] Devices that integrate sensors with computation and comunication.

software resident on the devices, hardware, or both, reducing their useful lives.

As shown in Figure 1 the NIST-DMC Model has two main layers:

Layer 1: The Open Networks Communication Layer defines a common object model to simplify the interface to connect transducers to the network, without the need to develop a new network standard. [12]

Layer 2: The Standardized Smart Transducer defines a Transducer Electronic Data Sheet to abstract the functionality a transducer can have. It allows designing new transducers that will be used transparently. [9]

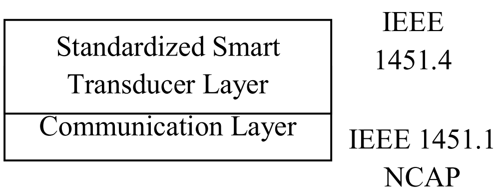

Fig. 1 NIST-DMC Model

In the field of software engeneering, the n-tier architecture is a term used to describe a methodology for designing and deploying applications. The methodology divides the functionality of an application into logical layers that interact to form a complete system.

The basic separation of an application proposes to divide the functionality of the system into three logical parts, one that abstracts the interface (user services), one which abstracts the rules of business services (business logic) and another that functions as a data repository.

In the case of distributed control systems, nowadays its development is based on Internet technologies, which includes not only Web applications, but the use of high-level languages, design and object-oriented programming, the use of open protocols, among others. Hence, the features and advantages of the methodology of n-layers, are appropriate to the design and implementation of these systems.

3 Distributed Control Model (DCM)

The DSCs in the market have control imbedded in themselves, thereby inhibiting another use, different from that for which they were conceived. In this paper we identify the benefits to be gained by changing this habit. We propose the use of a three-tier architecture that allows abstract control policies and that the DSC can dynamically modify its composition and behavior. It omits any consideration about the type of sensors used and the problems associated with the possible topologies of sensor network, focusing only on the application layer to propose a specific solution to the problem of handling dynamic control.

Figure 2 shows the 3-tier architecture used in the proposed model for distributed dynamic control (referred to as DCM). In the *client layer* are placed monitoring devices (Device Sensor), implemented through logic circuits. In the *business layer* are Web services that contain business rules. There is a web server, servlet, jsp and java beans, here are dynamic control rules, detected events are processed and it is sends messages to their local or remote actuators provided in the control rule; in the *server layer* is the database that can store data on players, devices, policies and events. It is also reflected in this figure that as the NIST-DMC-model lacks the support for control rules, it can benefits from these two layers. The following sections describe the main components of the DCM: control devices, dynamic management of control rules and distributed management of devices.

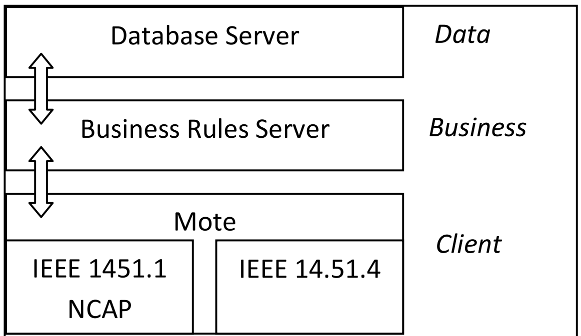

Fig. 2 Architecture for the control scheme with distributed intelligence

Control Device Characterization

Figure 3 shows that control devices can be designed by using the Model View Controller pattern, allowing abstract interface for managing devices, the distributed controller and the generic device model created by adding persistent components, grouped in 3 categories: the sensors that come to shape the set of artifacts that are permanently recording the occurrence of events, actuators that come to be the set of artifacts reacting to determined rules, instructed to do so by a driver, and finally are microprocessors, that can be microcontrollers or even personal computers.

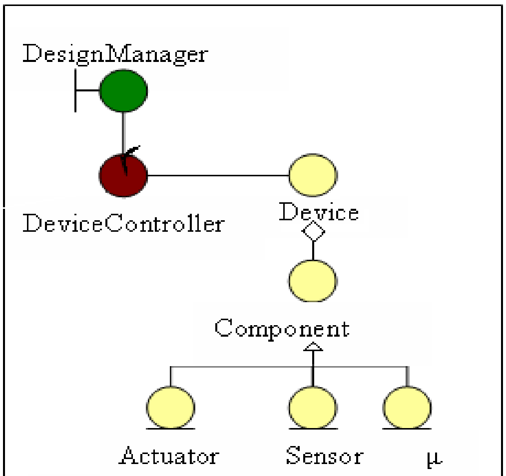

Fig. 3 Devices design with reusable components

From that abstract point of view, any device in the DCM is modeled as an object that has a number of attributes and that exhibit a behavior which can be implemented in hardware or in software as is stated in Figure 4. In this context, the components of a device are seen as the attributes of the same. Additionally, the behaviour of the object in the DCM is determined by its operations, and in the case of a device, the most common operations are related to the query and management of sensors and actuators, providing a service and setting its mode of operation.

In designing a device as an object, encapsulation property allows construction details from one device are hidden. To use this device only is required to know the set of operations or commands to interact with it. By using encapsulation in the device design, the control system obtains increased flexibility and a better way to adapt to the changes experienced by the devices, as well as changes in the user requirements. To integrate a device into the system, besides its activation, it requires to be recorded in the database along with the control policies in which it will collaborate. Once the device is operational, it can be accessed and managed remotely.

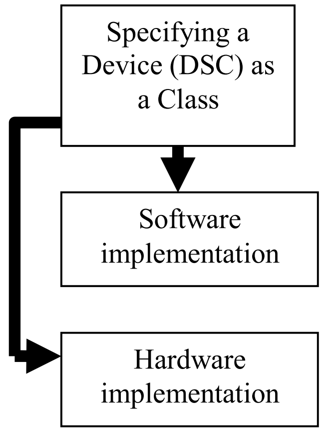

Fig. 4 Device implementation choise

3.2 Management of Control Rules

A control rule consists of one or more events and a series of actions that give them an answer. Any event generated by the sensors is recorded in a log, is evaluated against the control rules and if the necessary conditions are fulfilled, the controller executes the respective actions associated with a control rule capable of interpreting it and carry it out. These actions may in turn be events of different control rules. To configure the devices and to define the control rules are a duty of a control expert. These control rules can be edited by the DSC manager, which allows distinguishing two instances to manage them:

1. As part of the assembly of components, a default behavior is defined and stored in persistent memory in the form of policies. Thereafter the administrator can define a set of control rules, which modify its behavior. Rules should be evaluated and approved by the security administrator.

2. Whenever the DSC comes into operation, as a first step it checks for updates to the set of rules. In this case, the new rules can be added

to the rule set or they can replace those predefined in the persistent memory and will be in effect until you import a new set of them. The predefined rules are necessary to ensure a minimum performance in case you can not communicate with the *business layer*.

3.3 Distributed Architecture for Dynamic Policy Management

The control rules management consists mainly of two phases, one of which allows dynamic modification of those rules of control and the other controls its activation.

Modification of control rules: the DCM allows remote users to edit a set of policies on devices if they have the security rights to do it. These devices have a longest life and can be reusable, because its control rules are persistent in a table of the database, allowing an adjustment in line with the interests and capabilities of the administrator.

Activation of the control rules: when a device controller detects an event, this event must be registered in the database. Also it check if exist a control rule associated with it. In this case the commands are extracted from the database and the corresponding devices will receive instructions to perform the indicated actions.

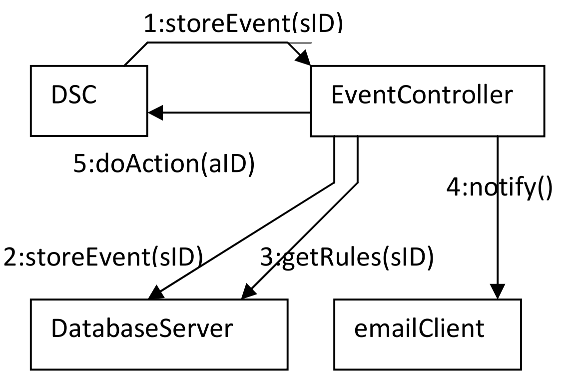

Fig.5 Control rules implementation

Figure 5 describes the protocol of activation of rules within DCM:

1. The DSC communicates to the Events Controller the fulfilling of any condition.
2. The Events Controller records in the database the received message.
3. The Events Controller searches into the database to determine if there is associated a control rule, if so, obtains from the database the actions to be performed.
4. The Events Controller could notify the user via the e-mail
5. The Events Controller orders the different actors to perform the corresponding actions that can be conceptualized as a chain of events.

A device only performs the actions associated with an event when these are indicated by the Events Controller. However sometimes the device can perform an operation on its own initiative, a case is when the device loses communication with the controller.

3.4 Rule selection

There are two different aspects involved in the rule fulfillment: the rule completion and the triggering mechanism.

a. The same event may trigger several rules at a time (see Fig. 6), to avoid trigger they always in the same order, it is necessary to use a non-deterministic policy.

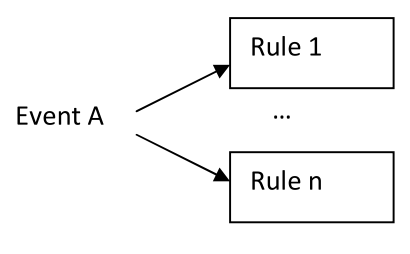

Fig. 6 Trouble at triggering rules

There are two options to do it:

1. Randomly choose the order in which the selected rules are triggered, which could be contradictory if priorities are handled.

2. Include priority and exclusivity within the rules as attributes, so that if a set of rules can be selected and their exclusivity allows it, choose the rule according to their priority. If still maintained a subset of eligible rules, you can use randomness in the activation sequence.

b. Perhaps the activation of a sensor complete in the same time several rules and hence must be decided whether those rules are triggered or not and in what order.

To solve this problem are included in each rule the following attributes

• PRIORITY which would have a value indicating the priority of the rule.

• EXCLUSIVITY, an attribute that indicates whether the conclusion of the rule can trigger it alone or if it permits to be triggered in association with the conclusion of other rules.

The rules can be of two types: simple or compound, whether they are simple only have a condition and an action, if composite may include other rules or use the following operators AND, OR, XOR, as shown in figures 7, 8 and 9.

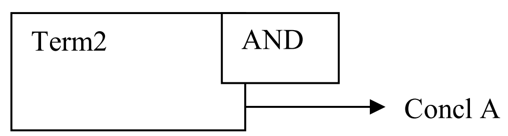

Fig. 7 Inclusive rule

Figure 7 indicates a conjunction between rules, where each rule is stored in a separate Tupel in the database.

Figure 8 represents an event (Term1) that completes more than one rule in the same time.

The Figure 9 represents an Event that can complete only one from many Rules that use the same term.

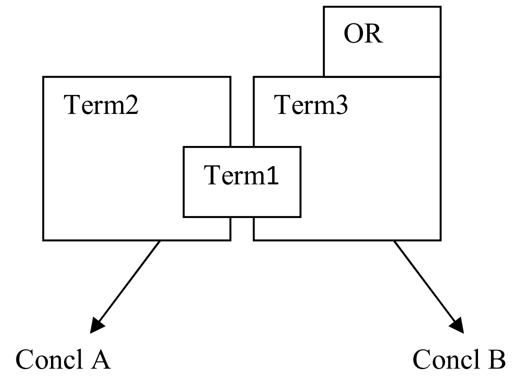

Fig. 8 Election Rule

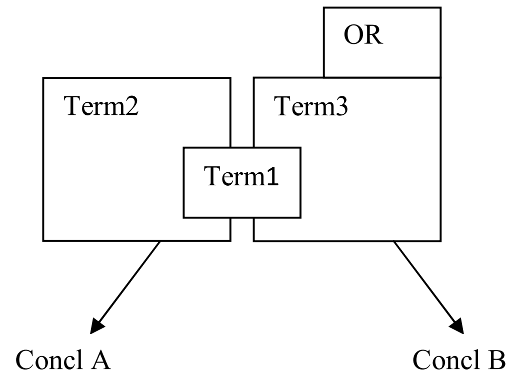

Fig. 9 Exclusive Rule

For the selection of rules in the DCM it was created a mechanism that allows nondeterministic selection of rules considering the priority and exclusivity of them.

3.5 Example: Designing Control for a Greenhouse

A greenhouse is used to grow plants in a better environment than on the field. However to maintain such conditions it is required continuous monitoring and control, because crops may be affected by numerous factors.

For example the objective of maintaining constant temperature inside a greenhouse, is threatened by climate change and solar radiation produced throughout the day. Additionally, the life cycle of a crop may require different environmental conditions, fertilization and irrigation, are applied according to their stages of development.

Typically the algorithms for managing a greenhouse temperature considered parts of the plant (root, stem, leaves) and variables that shape humidity, light, acidity (pH) of the substrate, amount of carbon dioxide and other. To achieve and maintain these stable conditions is common to use rule-based expert systems or fuzzy logic, and requires that such systems have ventilation systems, irrigation, heating and sometimes include air conditioning.

An automated greenhouse has sensors to monitor different variables such as humidity, temperature, carbon dioxide and light. It also has actuators as conditioned air equipment, irrigation systems and auxiliary lightning system. All of these sensors and actuators are configured in the DSC. As first stage to configure it, define control rules, exemplified partially in the following samples:

Variable: Temperature
Purpose: To maintain temperatures over 17 Celsius degrees
Rule 1: if the Temperature sinks under 17 Celsius degrees, turn the lights on

Variable: Temperature
Purpose: To maintain temperatures under 25 Celsius degrees
Rule 2: if the Temperature raises over 25 Celsius degrees, ventilate

Variable: Humidity
Purpose: To maintain humidity level under 70%
Rule 3: if the humidity increases over 70%, open the windows and ventilate

Variable: Humidity
Purpose: To maintain the humidity level in an interval
Rule 4: if the humidity increases over 70% or decreases under 60%, turn the bell on

If this rule set becomes rigid or if each rule is unchangeable, the system will be inappropriate and possible that system will be in use only for a short period of time.

By using the DCM the specialist can change the system behaviour at any time, updating the list of actions associated with a rule, deleting an undesirable rule or adding new rules to the set. This property allows a better use of the sensors and actuators in a greenhouse.

Once a time the system is running, new rules can be added, for example:

Variable: Temperature
Purpose: To keep the minimal temperatures in a day
Rule k: if the Temperature sinks under minTempD make it the new minTempD and notify it

Variable: Temperature
Purpose: To keep the variation of the day and night temperatures in a specific range
Rule k: if the Temperature at night is less than mintempD minus 10 and the outer temperature isn't, open the windows and turns the lights on

To implement this system was necessary to provide the DSC with remote interfaces to authenticate users, modify the behaviour of the devices in a dynamic environment and coordinate with the mail server associated.

4 Implementation

We developed a prototype of an automated greenhouse that includes all the elements described in the proposed model. In this implementation free software tools were used. One of the prototype monitoring devices is composed by a general purpose microcontroller with an ethernet interface built into the chip, making easier their connection to the network. The other components of monitoring device are a temperature sensor, a sensor of humidity and two electrical switches for lights and air conditioning. The controller is a Java application, which is primarily responsible for verifying and implementing the current control at the moment. The database, implemented with MySQL, contains the entities and relationships needed to show the performance characteristics of the model. The database was populated with data from users, devices, log of events and control rules. Finally, it was used Tomcat as web application container, consisting of XHTML pages, EJB, JSP, and Servlets. This application provides the user a platform to query and specify control rules in the database, and to communicate directly with the monitoring device.

Once developed and implemented the DCM, there have been three important results:

1. Ability to alter the runtime behaviour of a DSC
2. Integration of open source technologies in the design and implementation of embedded systems
3. DCM_NIST model was increased to enable distribution of dynamic control.

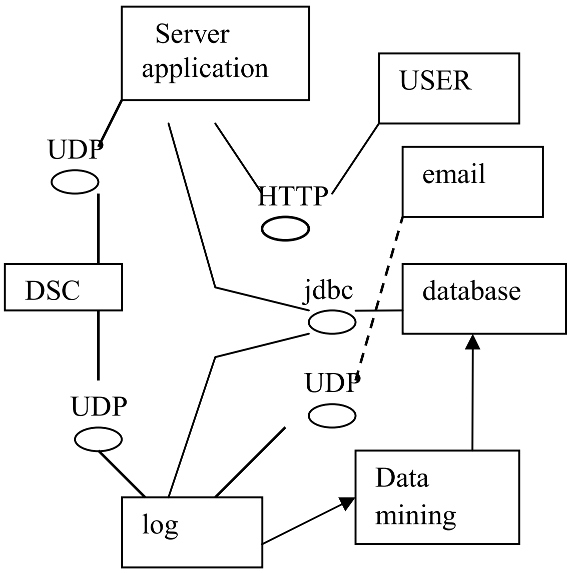

Fig. 10 Using DCM in a greenhouse

As stated in Figure 10, using the DCM in a greenhouse procures to obtain following advantages:

Adequacy, each manager can change the response that actuators must take after an event. This response is so flexible that in the future the manager can decide another process after the same event.

Automation, with this model it is possible to decide what environmental variables and crop properties are needed to control in an automatic way.

Scalability, the type and quantity of sensor and actuators can change in the future without to thrown the old ones.

Simplicity, the system has a GUI to update the database.

Learning capacity, with the data stored in the log and data about the crop (production, diseases, costs, technologies used, varieties, etc) the system can search for patterns and suggest what changes the control rules needs.

Remote access, to control the greenhouse and to program the response actuator must have is just needed to have an internet access point (mobile or not). Messages can be obtained in a PAD, intelligent phone, or email account.

A greenhouse which implements the DCM can easily be managed as a lab to research production procedures and crop handling techniques.

5 Conclusions

The modelling of a monitoring device through the concept of object increases its level of abstraction, which facilitates the maintenance of their hardware and software. Also, the definition of an interface that hides the implementation details contributes to the interoperability and scalability of the system.

The physical separation between policy and control devices enables the reuse of devices and policies.

The characteristics of the n-layers methodology are appropriate to the design and implementation of control systems.

The incorporation of the business layer in the DCM as a component of the control rule mechanism, extracts from the monitoring devices the task to process events and to communicate with the database server.

The DCM gives system administrators a tool for remote management of their policies and devices, making it ideal for remote monitoring when they are very critical conditions due to cost or time.

The DCM enhances collaboration between microcontrollers, because through control rules and the IP address associated with each device, the controller simplifies the delegation of tasks regardless of whether it is a homogeneous or mixed architecture.

By defining control rules based on events and actions, there is a better use of data generated by events, because the data can be leveraged both within the DSC as in other DSC and in applications connected via the network..

6. Acknowledgment

This work was supported by the Laboratory in Automate and Intelligent Systems in Biodiversity of the University of Costa Rica and the AECID from the Spanish Government in 2010/01. Besides, this work was supported in part by the Spanish Government under Grants from "Programa José Castillejo" in 2008/09.
.

References:

[1] Conner, W. S.; Heidemann, J.; Krishnamurthy, L.; Wang , X. & Yarvis, M. "Workplace Applications of Sensor Networks", Intel Research and Development University of Southern California, Information Sciences Institute. USC/ISI Technical Report ISI-TR-2004-591 http://www.isi.edu/div7/publication_files/Conner04a.pdf

[2] Culler, D.; Berkeley, D. & Srivastava, E. M. "Overview of Sensor Networks", IEEE Computer , vol:37 , no:8 , pp:41-49. Agosto 2004

[3] Deshpande, A.; Guestrin, C. & Madden, S. R. "Resource - AwareWireless Sensor - Actuator Networks", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 2005

[4] Dunkels, A.; Alonso, J.; Voigt, T.; Ritter, H. & Schiller, J. "Connecting Wireless Sensornets with TCP/IP Networks", Proceedings of the Second International Conference on Wired/Wireless Internet Communications (WWIC2004), Frankfurt (Oder), Germany, February 2004

[5] Dunkels, A.; Feeney, L.M.; Grönvall, B. & Voigt, T. "An integrated approach to developing sensor network solutions", Proceedings of the Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA, antes IEEE SNPA) Boston, Massachusetts, USA, August 2004 http://www.sics.se/~adam/sanpa2004.pdf

[6] Feng, W.; Walpole, J.; Feng, W. & Pu, C.. "Moving Towards Massively Scalable Video-Based Sensor Networks", Proceedings of the 9th annual international conference on Mobile computing and networking. ACM Press New York, NY, USA. 2003

[7] Heinzelman, B.; Murphy, L.; Carvalho, H. & Perillo, M. "Middleware to Support Sensor Network Applications", IEEE Network Magazine Special Issue. Jan. 2004.

[8] Kumar, S.; Shepherd, D. & Zhao, F. Guest Editors. "Collaborative Signal and Information Processing in Micro-Sensor Networks", IEEE Signal Processing Magazine, March 2002

[9] Lee, Kang; A Synopsis of the IEEE P1451-Standards for Smart Transducer Communication. http://ieee1451.nist.gov/1451synosis-599F.pdf

[10] Levis, P.; Gay, D. & Culler, D. "Active Sensor Networks", Proceedings of the 2nd USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI), May 2005

[11] Liu, T. & Martonosi, M. "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems", ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPoPP), June 2003

[12] Mark, John & Paul Hufnagel, The IEEE 1451.4 Standard for Smart Transducers. JJM/PJH 06-09-04 http://standards.ieee.org/regauth/1451/IEEE_1451d4_Standard_Genl_Tutorial_090104.pdf

[13] Obraczka, K.; Manduchi, R. & Garcia–Luna, J.J. "Managing the Information Flow in Visual Sensor Networks", 5th International Symposium on Wireless Personal Multimedia Communications, Honolulu, HI, October 2002.

http://www.soe.ucsc.edu/~manduchi/Papers/178-obraczka.pdf

[14] Szewczyk, R.; Polastre, J.; Mainwaring, A. & Culler, D. "Lessons From A Sensor Network Expedition", European Workshop on Wireless Sensor Networks 2004 http://www.eecs.harvard.edu/~mdw/course/cs263/papers/gdi-ewsn04.pdf

[15] Schneeman, Richard D. "Implementing a Standards-based Distributed Measurement and Control Application on the Internet", U.S. Department of Commerce, 1999.

[16] Yadgar, O.; Kraus, S. & Ortiz, C.L. "Scaling-up distributed sensor networks: cooperative large-scale mobile-agent organizations", http://www.umiacs.umd.edu/~sarit/ Articles/bar-ilan-chapter.pdf