# Data fusion in observer networks

Lj. Bodrožić, D. Stipaničev, D. Krstinić
*University of Split, Faculty of electrical engineering, mechanical engineering and naval architecture, R. Boškovića bb, 21000 Split, Croatia*
*e-mail: ljiljana.bodrozic@fesb.hr*

## Abstract

*Sensor networks (SN) and wireless sensor networks (WSN) are one of key technologies in development for 21 century, challenging the imagination of scientists and engineer because when deployed directly in an area of interest a large volume of data can be obtained. Increasing the volume of data, we must create robust software support for data collecting and utilization. In those cases sensor network with more powerful data processing has to be used. We have introduced such a network using the term "observer network".*

*The papers main topic is data fusion in observer network performed on different levels of observer network architecture. As an example of observer network a part of iForestFire, the advanced forest fire monitoring system is discussed.*

## 1. Introduction

With rapid growth of sensor manufacturing technology, wireless communication technology and creating new algorithms for ad-hoc topologies and routing, wireless sensor networks are becoming the technology of the century [1]. Although the sensor network can provide user all sorts of information, not enough work on knowledge extraction form data was done so far, so majority of data remains unused.

Sensor network consists of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions [2]. Communication between sensing units in most cases is realized using wireless communication, so the name wireless sensor network (WSN) is also in use [3]. WSN can significantly improve the accuracy and density of scientific measurements of physical phenomenon because a large number of sensors can be deployed directly where experiments are taking place [4]. Also by correlating appropriate values, inconspicuous information can emerge from raw data.

1-4244-1455-5/07/$25.00© 2007 IEEE

In order to utilize sensor network data, one must provide software support for data collection, verification and data fusion. In our work we have introduced unified three layer architecture for sensor network data collection, analysis and fusion, inspired by human perception [5] and common sense reasoning [6]. A sensor network, together with software support based on this architecture is called the observer network.

The three layer observer network architecture consists of data, information and knowledge layer which are modified sensor network architecture layers introduced in [7]. The architecture is implemented as a multi agent system. When monitoring sensor network data, a human operator would reject some of data because they simply "don't make sense". One of tasks of observer network is automation of such process by performing artificial common sense reasoning. When done by human, common sense reasoning is transparent, but equipping computer with those skills can become complex and requires a great deal of knowledge about how to think about some particular area [6].

Although sensor network is decentralized, sometimes sensors are clustered on sensor platforms [8,9], or sensor nodes. Sensor node is a group of sensors and actuators clustered together, measuring the same scene from the same perspective. Sometimes, it can be characterized by one master node and numerous sensors that the master node controls, or just common geographic location. With this, overall sensor network is hierarchically divided.

Sensor and data fusion is done within data from one or several neighboring sensors. Although some authors [8] consider sensor fusion and data fusion as synonymous, we will use the term sensor fusion only for fusion of actual readings from many different sensors, and data fusion would mean fusion of actual and past data from one sensor.

In the next section, the three layer architecture is presented and described. Data fusion is done with different aims on each layer. Data fusion and detail explanation of the observer structure on information layer and knowledge layer are described in third section. In forth section multi agent architecture of iForestFire system and our experiences are given.

## 2. Sensor network and observer network architecture

Sensor network technology is one of the key technologies for 21st century [1]. Today, there are many potential sensor network applications like environmental monitoring, building and structures monitoring, military sensing, physical security, traffic surveillance, video surveillance, distributed robotics and similar. Different sensor network architectures have been proposed, and in context of our case study we will focus on three layer architecture shown in Figure 1.

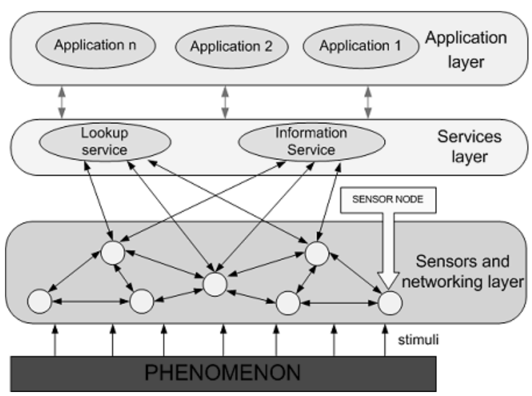


**Figure 1. Three layer architecture of a sensor network.**

Such three layer architecture was proposed by EU Eyes project [7], but we have modified it to better suit our approach and our ideas.

The first, bottom layer is the *sensors and networking layer* which contains the sensor nodes and the network protocols. The second layer is the *services layer*. It contains services for supporting sensors layer. Two major services have been identified. The *lookup service* supports sensors administration and the *information service* deals with aspects of collecting data. The third layer is the *application layer* where various applications could be built using services and sensor network.

Following the theory of knowledge engineering, and concept of intelligence spectrum [10], the sensor and network layer could be treated as **data level**, services layer as **information layer** and application layer as **knowledge layer**.

Our main area of interest are the information layer and the knowledge layer, because advanced sensor network features could be implemented on those levels, but first we will introduce the formal definition of the observer.

In 1987 Bennet, Hoffman and Prakash [5] introduced an approach to a study of perception that attempted to be both rigorous and general. They proposed a new formal foundation – the observer. The observer is defined as a six-tuple

$$O = (X,Y,E,S,\pi, \eta) \qquad (1)$$

where X and Y are measurable spaces, E and S are subsets of X and Y respectively, $\pi$ measurable surjective function and $\eta$ a conclusion kernel. Space X is a ***configuration space*** and E is a ***configuration event*** of the observer. Space X is a formal representation of those possible states of affair over which the configuration event E of the observer is defined. Y is a ***observation space,*** or ***premises space***, of the observer. Space Y is a formal representation of the premises available to the observer for making inferences about occurrences of E. S is the ***observation event***. All and only points in S are premises of observer inferences which conclude that an instance of the configuration event E has occurred. $\pi$ is a ***perspective map***, the measurable surjective function from X to Y ($\pi : X \rightarrow Y$) with $\pi(E) = S$. $\eta$ is a ***conclusion kernel*** of the observer. For each point in the observation event $s \in S$, $\eta(s,\cdot)$ is a probability measure on E supported on $(\pi^{-1}(s) \cap E)$. This means that kernel $\eta$ is a convenient way of assigning to every point of S a probability measure on E [5].

Let us now discuss on how the theory of perception and observer theory can be applied to sensor network application.

Perception, the process of acquiring, interpreting, selecting, and organizing sensory information, has a goal on identifying the occurrence of an event or scenario using only available particles of information and some a priori knowledge. Human observer is very reliable in identifying the scenario using its senses and heuristic knowledge.

In sensor network, each sensor maps some aspect of physical world surrounding it, usually using some of its side effects into a measurable space therefore, a sensor holds simple observer features.

In observer network architecture shown in Figure 2, on the data level, we have a group of simple sensors, utilizing this kind of observers, clustered together around master nodes which have more processing power. Usually master node does the transformation of incomprehensive data given by sensor to data readable by user. Recent data is held in memory and can be available on user request. Each sensor node $se_i$ on sensors and networking layer has its corresponding virtual nodes on services layer, and one virtual node at each application on application layer.

Information service has a set of virtual nodes arranged in triplets $i_i = (dc_i, ssv_i, sdv_i)$. Each triplet $i_i$ of information service virtual nodes corresponds to its sensor node $se_i$. Virtual node $dc_i$ is responsible for data collecting from sensor node, and $ssv_i$ and $sdv_i$ are responsible for sensor and data validation. These two virtual nodes are implementation of observer, and their observer structure will be discussed in the next chapter.

Each application on application layer also has virtual nodes. Virtual node $a_k$ at application layer could be connected with several virtual nodes at information service layer. Those information nodes form the set of information nodes $I_k = \{\ldots, i_i, i_j \ldots\}$ which correspond to application layer node $a_k$.

Each application on the application level is responsible for identification of some kind of event, and therefore is an observer. Observing on application level has much more dimensions then observing on information level and depends on the nature of the phenomenon observed [18].

Figure 2. shows the new sensor network architecture which could noe be treated as observer network.

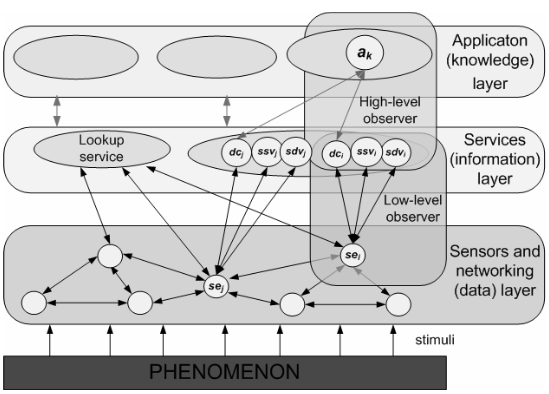


**Figure 2. Three layer architecture of the observer network.**

## 3. Observers and data fusion

Data fusion is done both on information and knowledge level, but different algorithms are applied. There is no need to explain the importance of having true, meaningful data in observer network. In the large volume of sensor data, collected from numerous sensors, a fault value sometimes occurs. It is important that such data is identified and rejected, so it is not taken into account during processing on higher levels. The main aim of data fusion on information level is identifying sensor network operation failure and triggering appropriate procedure for minimizing the damage of data loss.

There are two distinct types of failure that can be detected in observer network. Following the concept of information theory [10] we call those types of failure syntactic and semantic failure, by the level of consciousness that recognizes the error. Also syntactic validation can be addressed as sensor validation, and semantic as data validation.

### 3.1. Syntactic validation

Syntactic error can be spotted while perceiving syntactic aspects of information, which are: number of symbols, length of the signal, statistic attributes of the information source, and limitations in range of the value. Knowledge about the source of information is important here, because syntactic information is perceived in the aspect of specific sensor, which is why this observer must consult a knowledge base that holds information about the kind and limitations of the sensor. On this level of perception fusion of data given by the sensor and knowledge about the sensor, lead to the information if the data is valid.

When defining the specific observer, one must define the meaning of the six-tuple $(X, Y, E, S, \pi, \eta)$.

In syntactic observer X, the configuration space, is a continuous space with the same dimension as data expected from the sensor holding all values of the specific aspect of environment that is measured, while E holds values where syntactic failure can be identified. For example of temperature sensor, X is the space of real numbers, and E is the subset of possible temperature values, out of range -50 to 85. Perspective map is the function of the sensor, that is, the way sensor maps the environment into digital form, with $\pi(E)=S$.

Space Y is discrete space given with $\pi(X)$. Conclusion kernel assigns to each member of S a probability measure supported on E and depends on sensor kind and information available to observer. If environment is in state x, and $y \in Y$ is received from sensor, then the conclusion of the observer is that syntactic failure happened if y is in S, with probability given with $\eta$ or, if y is not in S, that data is syntactically valid.

### 3.2. Semantic validation

On the other hand, semantic failure is detected when value read from a sensor is not semantically valid. Semantic validation can be disseminated into two steps. The first step is discontinuity detection, and the other is common sense reasoning.

In the first step only data from a single sensor is taken into account, but temporally scattered, that is some historical data, usually last 10 readings from the same sensor is also taken into account. Usually, this set of data follow the same trend or pattern which can be identified from the data, learned in time, or be a part of the knowledge about the sensor.

The other step combines several sensors from the same observer node, and checks for irregularities. This is done with a help of knowledge base consisting of large number of simple rules, the same rules human observer uses when observing irregularities in sensor data.

Sensor fusion can be done between sensors of same kind, when only slight difference due to the different location is allowed, or between sensors that are of different kind that have some dependability.

In the sense of theory of perception and observer definition, the meaning of the six-tuple $(X,Y,E,S,\pi, \eta)$ is following:

X is an n-dimensional space, where n is the number of values the observer node holds plus one, with added time dimension. Y is also n–dimensional space, but while X is continuous, Y is discrete in respect to the sensor resolution or any other restrictions.

E and S are subsets of X and Y respectively, holding only combinations of values that implies semantic failure.

Perspective function is function that maps state of the world into a sensor input set, and in ideal conditions it should be identity function, but usually, it has components of discretization and fitting to allowed range. Conclusion kernel assigns to each value in S probability supported on E.

Let us notice that an important part of the perspective function in semantic observer is syntactic validation observer, because data rejected by the syntactic observer will not be taken into account by semantic observer. The term *nested observer* can be introduced.

Semantic observer uses values taken from several neighboring sensors, which are sensors from one observer node and observes semantic failures in readings from a sensor node. Semantic observer performs data and sensor fusion because not only current readings, but also past reading from the sensors are taken into account.

### 3.3. Application observer

On knowledge level, verified meaningful information is transformed to knowledge about phenomenon of interest. One or more observer node information is used for identifying the presence of the phenomenon if interest, neighboring nodes can be used for verification of phenomenon and detecting the exact location of the phenomenon.

Depending on the type of phenomenon, configuration space is n-dimensional space, where n depends on the number of values the phenomenon identification depends on.

In the spirit of observer theory X is a space of scenarios, and each scenario holds information about event occurrence, volume or strength of the phenomenon and location where the phenomenon occurs. Premises space Y has the same or less dimension as X and usually less points, loosing the continuity because of resolution and limitations of sensors. Space E is subset of configuration space holding only scenarios where the phenomenon of interest occurs, that is scenarios of phenomenon, and S is subset of premises spaces holding values that will be sensed when a scenario from E takes place. A perspective map is used to map space X into a space of measures, Y. The map describes sensor sensing functions and functionalities from the information level. S is an image of E in respect to perspective map.

When a scenario $x \in X$ occurs, a set of configurations y is sensed by observer network lower levels. Application observer checks if y is in S, or if the image corresponds to the premises indicating the event of interest. If an error is detected by information observers, y will not be in S. If $y \in S$, then, observer makes decision only that the phenomenon of interest happen. The exact scenario identification is task of the conclusion kernel, which is supported only on S. Even if y is in S it is still possible that x is not in E, due to the Y having smaller dimension and resolution than X. We call this situation false alarm. Conclusion kernel consults rule base in identifying such situations. When making conclusion on exact scenario, observer can also consult an observer assigned to neighboring node to verify and locate the event. In the next chapter we will present and describe a forest fire observer, application observer responsible for forest fire detection and identification.

## 4. Forest Fire Observer

Our particular interest focuses on forest fire protection, as an important part of overall environment protection. Forest fire is a phenomenon with severe impacts on vegetation, animals, landscape, soil, air and the human community living in those areas. The forest fire monitoring system is responsible for protecting large areas of land from devastation caused by forest fire by:

a) Real-time predicting forest fire danger index,

b) Early spotting and detecting forest fire in initial phase and alarming responsible fire fighting units,
c) Predicting forest fire behavior to support fire fighting management activities and
d) Enabling distant video and meteorological presence at forest fire field also to support fire fighting management.

iForestFire is an Integral Forest Fire Monitoring System, and it provides these functionalities via web interface [15,16,17,18].

A typical user interface of the system can be seen in Figure 3.

**Figure 3. Monitoring unit and user interface of experimental intelligent forest fire monitoring system iForestFire.**

*iForestFire*[13] is a TCP/IP-based system conceived of sensor network and central server units.

Central server unit used in case study is linux server with dual core Xeon™ running on 3.0 GHz processor, and was used for collecting, processing and storing all data.

Each sensor node consists of pan/tilt/zoom-controlled video camera connected to the network-embedded video Web server, mini meteorological stations connected to network-embedded data web servers, and wireless (IEEE 802.11 a/b/g) communication unit.

Multi agent system, incorporating the three layer architecture is responsible for senor network data collection, verification, and forest fire detection. Intelligent agents are implemented using JADE [11], and knowledge base is written in JESS [12] language. Agents run Rete algorithm in knowledge based reasoning.

An agent of type CollectorAgent is responsible for data collection and archiving data into a database. SyntacticObserverAgent is triggered when data is stored into database with purpose of checking syntactic validity. SemanticObserverAgent uses data from the same observer node and performs semantic validation of data from a sensor node.

The multi agent architecture of the system is shown in Figure 4.

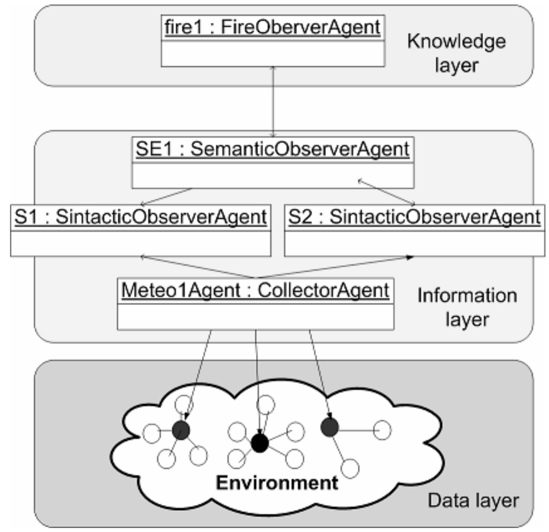


**Figure 4. Multi agent system of the observer network**

Only verified data is allowed to be used by application observer. While services layer observers give conclusions about the network functionality, the application observer gives information about presence of phenomenon of interest. Fire observer is designed for forest fire detection and it uses complex image processing algorithms in forest fire and smoke detection and alarm post processing procedures. Image processing algorithms will be a subject of our future paper, while here we will focus on multi sensor data fusion in alarm generation.

Although these algorithms are adaptive, parameters ranges should be known before the algorithms start. Parameters control sensitivity of detector, by calculating forest fire danger index, probability of clouds, illumination level and so forth. This can be done by defining several typical conditions, namely: rainy day, cloudy day, sunny day and very hot day, and creating the set of parameters for each of these configurations. Fusing sensor data, the actual conditions are selected and parameters can be adjusted.

If image processing detects possibility of forest fire, alarm needs to be verified using post processing data fusion. Sometimes, cloudy weather looks exactly the same as smoke, but human observer knows that if there is rain, no fire can exist. That is exactly what fusion with other sensor can provide us. Using knowledge base, an agent can reject alarm, when a drop of rain is recognized as a smoke, using information from leaf wetness sensor, as illustrated in Figure 5.

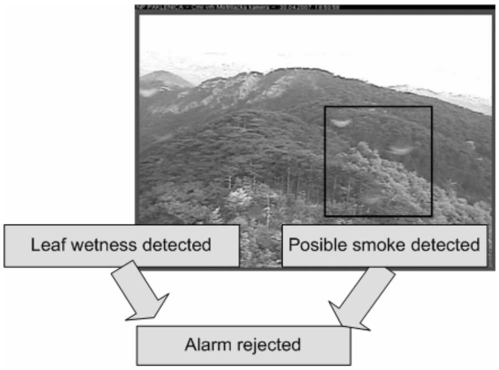


**Figure 5. Alarm verification procedure**

Alarm, once generated by image processing algorithm, can be rejected by post processing procedures, or by human operator.

## 5. Conclusion

In this paper we presented a unified architecture for data collection, verification and application based data fusion in an environment monitoring observer network. Formal theory of perception and definition of observer are used to describe data analysis and fusion procedures on each level of the architecture. The architecture can be used over sensor network consisting of large number of sensors distributed in space.

This architecture can be used in monitoring of any kind with sensor network of any kind, forming an intelligent environment.

Our case study was forest fire protection, because Croatian coast and islands have quite enhanced summer forest fire risk. This part of the iForestFire system helps human operators the most, giving them alarms only if a malfunction in sensor network or environmental hazard is detected, so there is no need for their constant attention.

## 6. Acknowledgements

This work was supported by the Ministry of Science, Education and Sport of Republic Croatia under Grant 023-0232005-2003 "AgISEco-Agent-based intelligent environmental monitoring and protection systems", TP-03/0023-09 "System for early forest fire detection based on cameras in visible spectra" [17]and Split and Dalmatia County authorities through a study [16].

## 7. References

[1] Chong C., Kumar S.P., "Senor Networks: Evolution, Opportunities and Challenges", Proceedings of the IEEE, VOL. 91, No. 8 Aug 2003

[2]K.Römer, F.Mattern "The Design Space of Wireless Sensor Networks", IEEE Wireless Communications, Vol.11, No.6, Dec. 2004, pp. 54 – 61

[3] Wikipedia -www.wikipedia.org

[4] Johannes Gehrke and Ling Liu, "Sensor-Network Applications," IEEE Internet Computing, vol. 10, no. 2, 2006, pp. 16-17.

[5] B.M.Benett, D.D.Hoffman and C.Prakashi, "Observer Mechanics – A Formal Theory of Perception", Academic Press Inc., Public Domain, 1989.

[6] Push Singh, Marvin Minsky, and Ian Eslick (2004). Computing commonsense. BT Technology Journal, 22(4):201-210.

[7] "EYES – Energy Efficient Sensor Network", EU Research Project, 2002 – 2005, http://www.eyes.eu.org

[8] T. Smestad, Data Fusion – for Humans, Computers or Both ?, Translated article from Mikroskopet, Norwegian Defence Research Establishment, Feb. 2001.

[9] Llinas, J., Hall, D.L., An introduction to multi-sensor data fusion, Circuits and Systems, 1998. ISCAS apos;98. Proceedings of the 1998 IEEE International Symposium on Volume 6, Issue , 31 May-3 Jun 1998 Page(s):537 - 540 vol.6

[10] N.Rožić: Informacije i komunikacije, kodiranje s primjenama, Zagreb, Alineja, 1992.

[11] JADE – Java Agent Development Framework - http://jade.tilab.com/

[12] JESS – Java expert system shell - http://herzberg.ca.sandia.gov/jess/

[13] D.Stipaničev, T.Vuko, Lj.Bodrožić, Location Determination of Automatic Forest Fire Monitoring Stations based on AHP and GIS data, TIEMS (The International Emergency Management Society) 2007 Int.Conference, Trogir, June 2007.

[14] Lj.Bodrozic, D.Stipanicev, M.Stula, "Agent Based Data Collecting in Forest Fire Monitoring System", Proc. of 2006. Int. Conf. SoftCOM 2006 , Split, Dubrovnik, Sept. 29.-Oct.1. 2006

[15] D.Stipaničev, B.Hrastnik, Holistic approach to forest fire protection in Split and Dalmatia County, study for Split-Dalmatia County, 2004. (220 pages in Cratian)

[16] D.Stipaničev, T.Vuko, D.Krstinić, M.Štula, Lj.Bodrožić, Forest Fire Protection by Advanced Video Detection System - Croatian Experiences ,Third TIEMS Workshop - Improvement of Disaster Management System, Trogir, Sept. 26 - 27, 2006., CD proceeding, 10 pages

[17] D.Stipaničev, D.Krstinić, M.Štula, Lj.Bodrožić, Technological project TP-03/0023-09 "System for early forest fire detection based on cameras in visible spectra", Ministry of science, education and sport of Republic Croatia, 2006.

[18] D.Stipaničev, Lj.Bodrožić, M.Štula, , Environmental Intelligence based on Advanced Sensor Networks, 14th Int. Conference on Systems, Signals and Image Processing, Maribor June 2007.