C. Alcaraz, and J. Lopez, "Diagnosis Mechanism for Accurate Monitoring in Critical Infrastructure Protection", *Computer Standards & Interfaces*, vol. 36, pp. 501-512, 2014.
http://doi.org/10.1016/j.csi.2013.10.002
NICS Lab. Publications: https://www.nics.uma.es/publications

Diagnosis mechanism for accurate monitoring in critical infrastructure protection

Cristina Alcaraz, and Javier Lopez

Computer Science Department, University of Malaga,

Campus de Teatinos s/n, 29071, Malaga, Spain

{alcaraz,jlm}@lcc.uma.es

October 27, 2015

Abstract

Situational awareness for critical infrastructure protection, such as for energy control systems, has become a topic of interest in recent years. Despite attempts to address this area of research, more progress is still necessary to find attractive solutions that help bring about prevention and response at all times from anywhere and at any time. Given this need, we therefore propose in this paper, a smart mechanism able to offer a wide-area situational awareness with the ability to: (i) Control the real state of the observed infrastructure, (ii) respond to emergency situations and (iii) assess the degree of accuracy of the entire control system. To address these aspects, the mechanism is based on a hierarchical configuration of industrial sensors for control, the ISA100.11a standard for the prioritization and alarm management, and the F-Measure technique to study the level of accuracy of a sensor inside a neighbourhood. As proof of the functionality and feasibility of the mechanism for critical contexts, a software application implemented in nesC and Java is also presented in this paper.

Keywords: Critical Infrastructure Protection, Situational Awareness, Industrial Wireless Sensor Networks, the ISA100.11a standard, Accuracy

1 INTRODUCTION

Being aware of a situation in critical contexts is currently a matter of utmost importance within the research field of Critical Infrastructure Protection (CIP). International organisations and experts are combining efforts to tackle the topic of situational awareness [1]. This is the case at the National Institute of Standards and Technology (NIST), which not only classifies this need in [2] as one of the eight priority areas for protection, but also defines it as a new concept denominated as Wide-Area Situational Awareness (WASA). WASA consists of controlling and optimizing system resources deployed over large geographic areas, as well as delivering smart solutions in charge of prevention and response

before interruptions can arise within the system or between systems [2]. This means that it is necessary to address any type of instability, unforeseen event or potential fault caused by malicious actions [3] that may have a local, regional or national effect due to the existing interdependency relationships between Critical Infrastructures (CI) and their sectors [4]. According to the three latest reports of incidents in critical sectors published by the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) [5, 6, 7], these incidents (caused by failures or attacks) have become more and more prevalent in the last few years. Figure 1, based on statistical values taken from [5, 6, 7], illustrates this increase where one of the most affected critical sectors is precisely the energy sector and its control systems, also known as Supervisory Control and Data Acquisition (SCADA) systems.

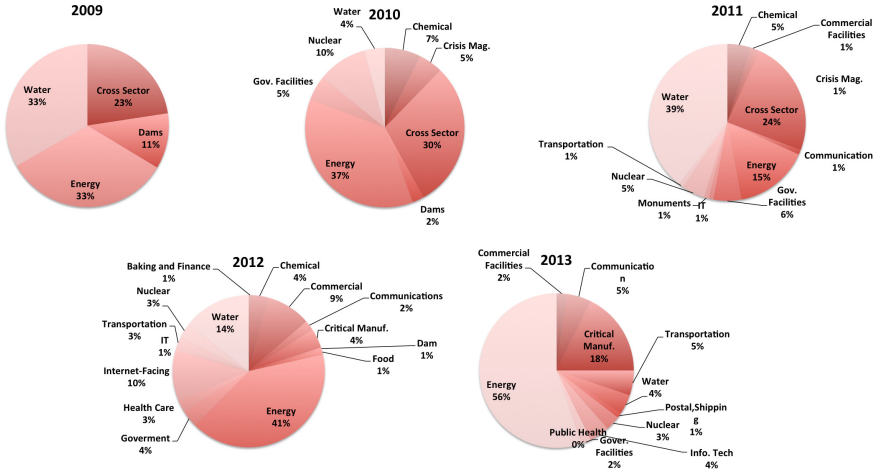

Figure 1: Incidents reported by the ICS-CERT in [5, 6, 7]

The consequences of this may be devastating with a high probability of triggering the famous cascading effect between critical systems. Therefore, Figure 1 clearly shows why operative agents (e.g., human operators) should be made aware of these situations so as to anticipate anomalies or deliver a rapid response. Moreover, this degree of protection should be provided by standalone solutions with proactive and reactive capabilities that help the underlying system work alone, especially, at distant locations, where the control may be reduced to a few human operators in the field. Taking into account the criticality of the application context and the goals of WASA, our main contribution in this paper therefore is to provide a smart mechanism able to offer interactive monitoring and protection of small control sub-domains, such as energy transmission/distribution substations. The mechanism is principally based on the technology

of Industrial Wireless Sensor Networks (IWSN) as part of an observation and protection system. Nonetheless, as this technology and its sensory devices can have a significant tendency towards generating operational errors [8] caused by hardware or software failures or some type of threat [8, 9, 10], the proposed mechanism also controls the behaviour so as to determine the degree of accuracy in observation and protection tasks.

To compute this accuracy, topics relative to the accurate detection of and response to anomalous behaviour are addressed, together with aspects related to alarm management offered by current industry standards such as the ISA100.11a standard [11]. Any information produced within the observation system, has to be locally monitored by human operators in the field and remotely supervised by the SCADA Center so as to be aware of the real state of both the underlying system and the protection system at all times. In order to validate the mechanism and show how it is able to offer an attractive way to deal with unforeseen situations and self-evaluate its functional capacities, a critical scenario is stressed (i.e., intentionally provoking emergency situations) to analyse the behaviour of the entire approach. The results show that this is a solution that could help the SCADA system know the real-state of its components, and even improve its governance, risk management, auditing and maintenance as stated in [12].

The paper is organised as follows. Section 2 introduces related work, and Section 3 presents the general architecture and the functional goals of the WASA mechanism together with the technologies described above. Then, we introduce the solution in Section 4, describing the prevention method applied to control the reliability in the observation tasks. Section 5 analyses a use case based on the results obtained from the simulation where a software application is also introduced. Section 6 concludes the paper and outlines future work.

2 Related Work

There are some experts in the CIP field that are currently developing attractive solutions [13, 14, 15] for CIP based on situational awareness. Most of these solutions follow similar architectural designs based on data recollection, analysis, verification, alerting and storage; a set of fundamental procedures that complies with the WASA methodological framework given in [16]. This framework is composed of the combination of two perspectives related to context-awareness and hybrid solutions which aim to contextualize and represent the environment considering human objections. This human interactivity and the degree of automation depend on a series of factors [17]; amongst them, the critical nature of the infrastructure, the application context and conditions.

The WASA methodological framework states the importance of the composition of primary and secondary actions. Primary actions consist of the normalization, understanding and representation of the context (e.g., through finer-scale analysis solutions) in order to offer an efficient and rapid alert and response to emergency situations. An example of this representation of the context is

given by Q. Hairong et al. through a conceptual framework in [18], which is able to understand complex context events by applying unsupervised event techniques. W. Xing et al in [15] present a comprehensive security monitoring and warning system with the capacity to constantly visualize emerging power system applications. In contrast, secondary actions are related to the additional capabilities of the system to ensure state recovery, stabilization, learning and updating, assessment and report. Orchestration of these actions and the use of complementary systems, such as technologies, protocols and applications [16, 19] through regulatory frameworks and standards [20], benefit the entire system and its interaction with the environment.

International organisations, such NIST or the European Network and Information Security Agency (ENISA), are also wholly motivated to introduce topics of situational awareness to provide protection benefits at different levels, either at cyber or physical level [2, 12]. For example, preparedness, response and recovery plans could be improved using context information and activity within such a context [12]. However, M. Endsley et al. state in [1] that despite this progress and motivations, investigation in the area of CI is still needed; and according to [2] this research should focus on control and resource optimization, prevention and response from '*anywhere*' and '*at any time*'. The reason for this lies in dimension issues of the infrastructure itself and its location within the system.

3 General Architecture of the WASA Solution

The solution proposed in this paper is illustrated in Figure 2. In this figure, one is able to see that the solution follows a hierarchical distribution based on clusters of nodes. Each cluster is composed of a small set of smart devices which are near to each other, known as industrial sensor nodes. These sensory devices have enough capability (microprocessor of 13-180MHz, 256-512KB of RAM, 4-32MB of ROM) to sense real states of an observed object or its surroundings, and provide useful services for its control, such as constant monitoring, diagnostics, tracking and reporting services. The control of each cluster is managed by certain trustworthy entities within each cluster, known as *Cluster Head* (denoted in this paper as CH), which have the inherent ability to manage, validate, aggregate and filter information [21].

Considering all this, it is possible to think that CHs may be single points of failure. However, our motivation by considering such a configuration is for several reasons. Firstly, if the local supervision was specified in a (flat) distributed configuration, the approach should then have to be integrated in each sensor device. This may increase the costs associated with computation (each node must compute information from their neighbours) and communication (any information must be sent to the controller). Secondly, current communication standards, such as the ISA100.11a, have diagnosis mechanisms available with the ability to check the lifetime and the existence of network devices [11]. Thirdly, we assume that the control system follows a rigorous security policy

to protect these critical points and their communication, using for example re-clustering techniques, frequent diagnosis mechanisms, lightweight intrusion detection mechanisms, redundant configuration, use of surveillance systems or strict and well-executed maintenance procedures [22, 9]. Fourthly, and lastly, it would be useful to be able to exploit the capabilities and intelligence of the CHs (normally they have greater capabilities than the rest of the sensors) to carry out a great part the approach at different points of the system.

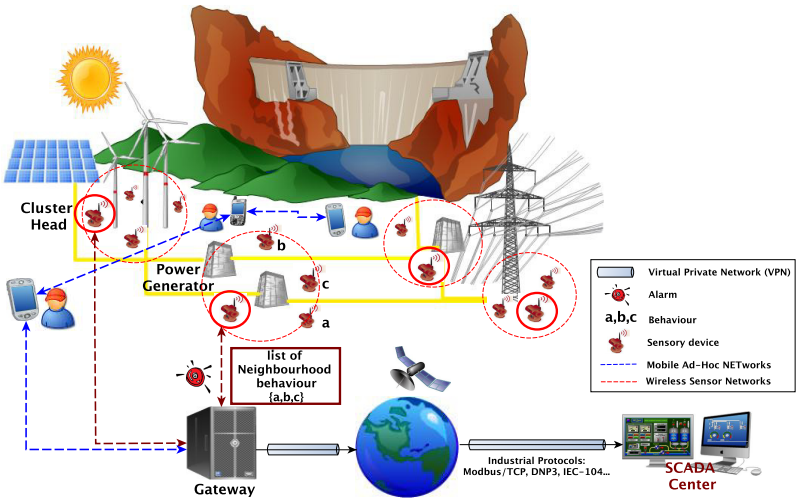

Figure 2: General Architecture of the WASA Solution

Any information received from these types of devices has to be sent to their respective CHs to manage, check, filter and aggregate the information received. When the information is aggregated, the CH also has to send this information to an intermediary interface between the observation system (i.e., the IWSN) and the SCADA Center. This interface is a powerful gateway that is able to process and manage a large amount of information received from the acquisition network, and interpret and translate different types of industrial communication protocols (e.g., ISA100.11a messages to Modbus/TCP messages, and vice versa). In order to link this concept of network configuration and its functionality to our mechanism, sensor nodes are responsible for diagnosing anomalous behaviour and alerting their CH to the situation. These in turn are responsible for filtering and aggregating valid information as well as analysing the recent past behaviour of each sensor. This also means that the gateway has to manage any information received from the CH (e.g., alarms or measurements), alert both the SCADA Center and the nearest human operator of any critical situation in the affected area, and evaluate the final behaviour of each sensor. However,

it is also necessary to understand the importance of the Internet and the use of heterogeneous communication technologies (e.g., Mobile Ad-Hoc NETworks (MANETs)) for control over large geographic areas. Moreover, redundancy aspects should also be considered in these types of solutions, given the criticality of the applications and their environments. The use of redundant components (e.g., two gateways) and protocols (e.g., store and forward) would help the control system maintain its supervision and monitoring at all times.

Regarding human operators, they can visualize the scenario and their surroundings using their hand-held devices (e.g., cellular or PDA devices).These interfaces facilitate local automation in the field, by managing: (i) Measurements associated with observation (e.g., voltage magnitude), (ii) alarms with relevant information on real states from the observed infrastructure, or (iii) commands that include a particular control action (e.g., stop/activate turbines). For communication from/to sensors, it is currently possible to apply wireless industrial communication protocols, such as ZigBee PRO [23], WirelessHART™ [24] or ISA100.11a. However, our mechanism is mainly based on the ISA100.11a standard, which is an extended version of WirelessHART™ and by providing useful high-level services, was specifically designed for industrial environments. These services include coexistence with other technologies, reliability of the communication through hopping and blacklisting methods for radio frequency change, security based on cryptographic services, and prioritization and warning services for control systems and industry. More specifically, ISA100.11a classifies the alarms into four types: Device diagnostics, communication diagnostics, security, and processes with five levels of priority: *Urgent*, *high*, *medium*, *low*, *journal*. These alarms are managed by each network device, but only one of them (generally, the gateway) is responsible for buffering them, using organized queues according to their given levels of priority. Regarding ZigBee PRO, its security is still weak, as an attacker could deduce the security credentials through an attack of differential power analysis (variations of power consumption of the microprocessor/memory whilst performing cryptographic operations) and by knowing a priori both the master key and the Symmetric-KeyKey-Exchange (SKKE) scheme [9].

For reasons of simplicity, we assume that the communication link 'sensor-sensor' and 'sensor-gateway' are protected by using the key management schemes of ISA100.11a [25]; whereas the communication links 'gateway-hand-held' and 'gateway-the SCADA Center' are protected through security services belonging to the TCP/IP standard such as Socket Secure Layer/Transport Layer Security (SSL/TLS) and Virtual Private Networks (VPN) using the Internet Protocol Security (IPSEC) tunnel mode. Note that some of these security services can also be found in the new version of the Internet Protocol IPv6 known as RFC-6272, Internet Protocols for the Smart Grid [26]. This RFC has been defined to allocate a considerable number of smart network devices, where it is expected that the vast majority of them will be connected with automated energy substations, such as smart meters or sensory devices. This is in accordance with the proposal of the American Recovery and Reinvestment Act (ARPA) of 2009 [27], which includes an investment of hundreds of automated substations with

thousands of sensor nodes to detect unforeseen changes and prevent local or regional power blackouts.

Considering the capabilities of the IWSN technology and the functional services of ISA100.11a, three chief functional characteristics are highlighted below, which represent the main benefits of this approach for critical contexts.

1. '*Control and diagnosis*' of disturbances or faults registered within a particular CI and/or in its industrial equipment, such as electricity pylons or generators. A disturbance could be, for example, abrupt changes of temperature registered in industrial engines or transformers, or peaks in voltage in electricity pylons. Part of this control is also focused on '*Identifying behaviour-based anomalies*' within a cluster of sensors. The idea is to increase the intelligent capabilities of CHs to detect local anomalous behaviours with respect to the general conduct of the rest of the nodes of a cluster. Existing local deficiencies caused by energy exhaustion, malfunction or an attack may also hamper the general supervision of the infrastructure and it is necessary to anticipate such situations since the protection of the entire system has been entrusted to these types of devices.

2. '*Respond to emergency situations*' to prevent any effect that may produce a cascading effect, and in this way comply with the conditions given in [16]. In addition, the hierarchical configuration helps the system locate problems by knowing, a priori, the network deployment.

3. '*Evaluate the level of accuracy of each sensory device*' with respect to a set of factors: Its activity noted in the recent past, the activity noted by each member of its neighbourhood and the feedback on the true situation received after a manual inspection. For this study, an accuracy statistical technique known as the F-Measure or F-Score [28] is applied. According to the Joint Committee for Guides in Metrology (JCGM) [29], accuracy can be defined as the "*closeness of agreement between a measured quantity value and a true quantity value*" and where "*the measurement accuracy is then said to be more accurate when it offers with a smaller measurement error*''. This concept is often confused with the concept of precision, which means the"*closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions*" [29]; i.e., obtain repeated sequences of samples under similar conditions with similar results. Given this and taking into account the focus of this paper, we need to determine the degree of correctness in the observation tasks through the concept of accuracy.

For the sake of clarity, all these goals, characteristics and their importance within the WASA mechanism are described in more detail in the remainder of this paper.

4 Protection through Diagnosis, Response and Assessment

Considering the architecture and the goals stated in Section 3, the next step is to provide each system element with a set of WASA functionalities.

4.1 Sensor Nodes: Dissemination and Warning

Each sensor node of a cluster has to be deployed near to the observed CI with the mission of monitoring a particular object (e.g., electrical generators) or its surroundings. For a constant monitoring, sensor nodes have to continuously sense, through their sensors, physical events (i.e., measurements) from the system under observation such as levels of temperature, voltage or pressure. Given that the proposed mechanism focuses on power scenarios, such physical events relate to values of voltage, denoted as v_i.

The left hand side of Figure 3 illustrates the software architecture of the observation system, which includes two chief modules: *Behaviour Pattern* (BP) module and *Alarm Manager* (AM) module. The BP module is in charge of analysing, through behaviour patterns, two types of states: *Normal states* and *anomalous states*. Normal states refer to those samples that are inside the normality thresholds defined by the organisation or its security policies. In contrast, anomalous states correspond to the behaviour of the object under observation, which deviates from the standard, expected, or usual.

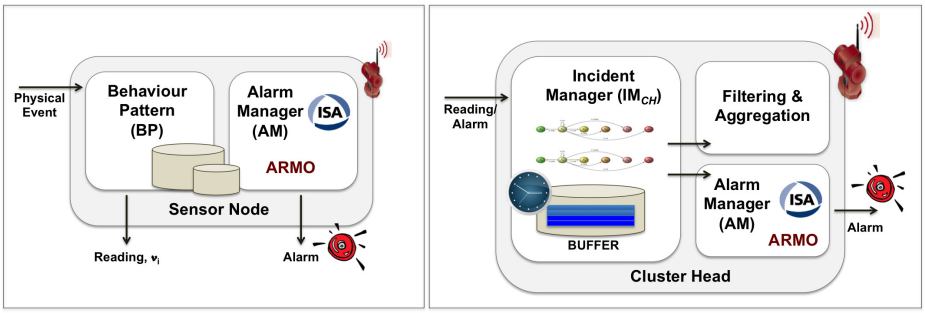

Figure 3: General Architecture of Sensor Nodes (Left) and the CH (Right)

In order to link normal and anomalous states with respect to the expected performance of the observed CI, our analysis focuses on *infrastructural states* to examine whether physical events (e.g., values of voltage, *vi*) deviate, or not, from the norm. Moreover, a normal state can likewise be subdivided into two further categories: *Satisfactory states* and *acceptable states*. A satisfactory state

refers to valid voltage readings that are within predefined thresholds, i.e., $v_i \in [V_{min}, V_{max}]$, where V_{min}, V_{max} represent the minimum and maximum values of voltage. An acceptable state corresponds to those infrastructural states that slightly deviate from the norm, but are acceptable according to security polices. This latter category is related to those alarms labelled with levels of minor priority.

To represent all these states, the BP module not only has to analyse each event received but it also has to tag it according to the five levels of criticality defined by ISA100.11a: *Journal* with value 1, *low* with 2, *medium* with 3, *high* with 4 and *urgent* with 5. Namely:

- Normal states are signalled with values (0-3), where the zero is reserved for satisfactory states and (1-3) for acceptable states. In other words:
  - Satisfactory states correspond to $v_i \in [V_{min}, V_{max}]$.
  - Acceptable states correspond to $v_i \notin [V_{min}, V_{max}]$, but they are within normal requirements. As it is possible to find three types of acceptable states (journal, low and medium), it is necessary to define the respective thresholds for each of them.
- Anomalous states are associated with values (4-5) to represent unstable states as well as an indicator of urgency (high and urgent, respectively).

Once the BP module has identified, through behaviour patterns, the different types of states and their levels of criticality, it has to warn the AM module of the situation. This module is responsible for generating a new ISA100.11a alarm and sending it to its CH through the ARMO (Alert Reporting Management Object) class. ARMO is a class of ISA100.11a, in charge of generating alerts with different priorities using the AlertReport service belonging to the DMAP (Device Management Application Process) class, which includes a set of objects for local or remote configuration and supervision of network parameters.

4.2 Cluster Head: Behaviour Analysis of the Neighbourhood and Warnings

Although sensor nodes are able to disseminate information to their respective CH, the main core of the mechanism is primordially distributed and located between the CH and the gateway. The goal at this point is to provide simple and straightforward diagnostic techniques to offer support for situational awareness. These techniques, described below, have to be implemented in each CH in order to diagnose any activity performed by their sensors. This diagnosis is fundamental because it could be assumed that each new node joining the observation system is a reliable and trustworthy node. However, their conduct can change over time due to internal faults (caused by hardware or software errors), external faults (produced by unplanned accidents or incidents) or simply because the node itself is a compromised device with malicious actions.

The module in charge of the diagnosis is illustrated in Figure 3 on the right hand side, labelled *Incident Manager* (IM_{CH}). However, its functionality does not end there. It has to forward the information received from its sensors to the gateway. If the information received is labelled with the value zero corresponding to a '*satisfactory state*' (i.e., $v_i \in [V_{min}, V_{max}]$), it has to be filtered and aggregated by the Aggregation module. Otherwise, it has to be sent to the Alarm Manager (AM) to produce an ISA100.11a alarm. In either of these two cases and before sending any information to the gateway, the IM_{CH} also has to store the aforementioned information in a temporal buffer so that it can periodically analyse anomalous behaviour shown by any sensor with respect to its neighbourhood.

Diagnostics basically concerns the definition of a discrete probability distribution between those states in which a node can remain or transit to. The transition of these states can be represented through a directed graph where the edges are labelled with probabilities of transiting from one state to another. Given this, and considering the five levels of criticality of ISA100.11a, we define $\mathcal{G}$ as the transition graph between states where each state belongs to $\mathcal{S} = \{s_0, s_1, s_2, s_3, s_4, s_5\}$. The probability of going from a s_β state to another s_α state takes the following transition distribution:

$$Pr(s_{k+1} = \alpha | s_k = \beta) = p_{\beta\alpha} \tag{1}$$

In order to represent the transition graph $\mathcal{G}$, we assume that the probability of moving from a state of greater criticality is lower than moving to a state of lower criticality, the criticality of which depends on the priority of the messages received from sensors with values (0-5) $\in \mathcal{S}$. We provide, in Equation 2, an initial approach to calculate the probability of transition of each state s_α assuming a priority order; i.e., $p_{s_0} > p_{s_1} > p_{s_2} > p_{s_3} > p_{s_4} > p_{s_5}$.

$$p_{s_\alpha} = \frac{1}{4 \times \alpha} \tag{2}$$

Note that this transition probability distribution is very general where we mainly consider the priorities given by ISA100.11a to facilitate the experimentation in the laboratory. In practice, we recommend that a set of factors to determine the real distribution of a context should be taken into account, such as the characteristics of the application, the frequencies given by a situation and its restrictions. A learning process based on classification and labelled can become a requirement to identify the different criticality thresholds of a context [30]. This procedure may involve (i) an initial training phase to detect several priority classes, in which a classifier learns of the situation using for example a labelled dataset; and (ii) a testing phase to classify situations using the classifier. Once the priority classes have been identified, it is then possible to determine the distribution degree and the priority order to be considered for the approach. We, for example, assume for the experimentation that less critical situations are more probable than critical situations; i.e., $p_{s_i} > p_{s_j}$ and $i < j$; however, this assumption is dependent on the application context.

As Equation 2 is only feasible when $\alpha > 0$, the calculation of moving from a s_β state to a s_α state with $\beta \geq 0$ is therefore defined in the following way:

$$Pr(S_{k+1} = \alpha | S_k = \beta) = \begin{cases} 1 - (\sum_{\alpha=1}^{5} p_{s_\alpha}), & \text{if } \alpha = 0; \\ p_{s_\alpha}, & \text{if } \alpha > 0; \end{cases} \tag{3}$$

where, $\sum_{\alpha=0}^{5}(Pr_{\beta\alpha}) = 1$. The result of calculating Equation 3 is represented in Table 1 and illustrated from Figure 4 to Figure 9.

States (s_α)	s_0	s_1	s_2	s_3	s_4	s_5
Probabilities (p_{s_α})	0.4291	0.25	0.125	0,0833	0.0625	0.05

Table 1: Probabilities for Each State of $\mathcal{S} \in \mathcal{G}$

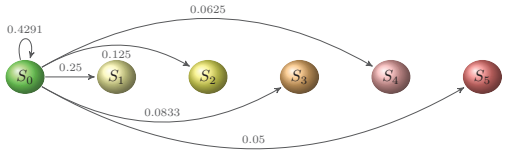

Figure 4: State 0 and Probabilities

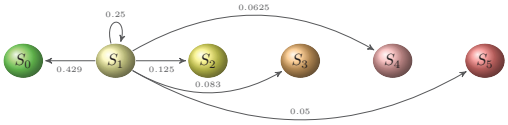

Figure 5: State 1 and Probabilities

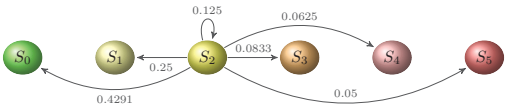

Figure 6: State 2 and Probabilities

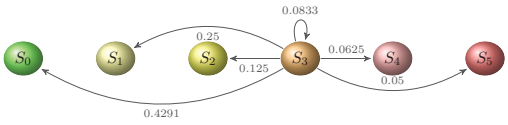

Figure 7: State 3 and Probabilities

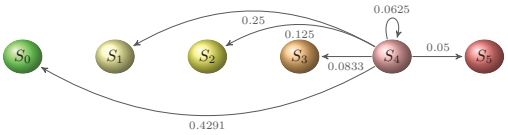

Figure 8: State 4 and Probabilities

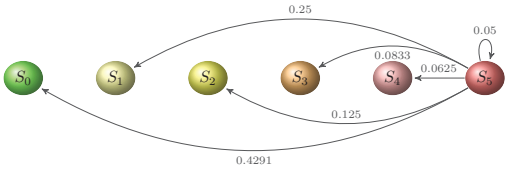

Figure 9: State 5 and Probabilities

To detect anomalous behaviours, each CH must periodically analyse the evidence streams registered in the temporal buffer. This buffer is divided into

several arrays with size Δ_{size} designated to each sensor ($sensor_i$ with ID_{sensor_i}) of their neighbourhood, where the cells of the arrays contain different values of criticality s_α, where $\alpha \in$ (0-5). To compute criticality decisions taken by sensors in the recent past, the system has to recover the criticality values in order to calculate the total sum of probabilities of transitions between states. The result is an average criticality value obtained from the sum of the criticality probabilities of moving to states. In other words, taking into consideration the notation represented in Equation 1, the behaviour of a $sensor_i$ in the past[1] is computed as follows:

$$\begin{aligned} prob &= \sum_{j=0}^{\Delta_{size}-1} Pr(s_{k+1} = array_i[j+1] | s_k = array_i[j]) \\ &= \sum_{j=0}^{\Delta_{size}-1} Pr_{array_i[j]array_i[j+1]} \end{aligned} \tag{4}$$

where, the $array_i[j]$ represents the state β and the $array_i[j+1]$ the state α. Note that in the case where an $array_i$ maintains '*anomalous states*' with values of high or urgent criticality, it is important to consider a further two situations:

1. Calculate the number of critical states (s_4-s_5) included within the $array_i$. We denote this type of computation with the variable $freq_{alarms}$.
2. Calculate the number of jumps generated from normal states (s_0-s_3) to anomalous states (s_4-s_5), and vice versa. The variable that contains this value is denoted as $changes_{states}$.

In the case of the first, it is necessary to count the frequency of critical alarms (4-5), and for the second an analysis of the abrupt changes between states is required. These changes can be analysed by pre-computing the sequence of alarms stored in $array_i$, generating another sequence based on binary values named here as $binarySeq_i$, with values 0 or 1 and an array size with Δ_{size_i}. Any entry in $array_i$ with high criticality (4-5) is assigned to the sequence $binarySeq_i$ with value one, otherwise with value zero. This can be better understood with an example. Let us consider an alarm sequence with $\Delta_{size_i} = 10$ and values 0 5 1 0 5 0 0 4 2 0. As this sequence stores critical values, its binary sequence would then be 0 1 0 0 1 0 0 1 0 0. With this new sequence, it is possible to calculate the difference (Equation 5) between consecutive values and the total number of abrupt jumps between states.

$$changes_{states} = \sum_{j=0}^{\Delta_{size-1}} |binarySeq_i[j] - binarySeq_i[j+1]| \tag{5}$$

Depending on the variables $freq_{alarms}$ and $changes_{states}$ given above, three situations can occur and one of them has to be computed to obtain the general behaviour of a sensor. They are as follows:

- Case A: $freq_{alarms} > 0$ and $changes_{states} > 0$; e.g., 0 5 0 5 0 5 0 5 0 5. This means that the sensor sends irregular values that can indicate a hardware or software problem, or even a threat.

[1] The time window for diagnosis could be established by the Δ_{size}.

- Case B: $freq_{alarms} > 0$ and $changes_{states} == 0$; e.g., 4 5 4 4 4 4 4 4 5 5. This situation indicates that the sensor considers that an emergency situation exists.
- Case C: $freq_{alarms} == 0$ and $changes_{states} == 0$; e.g., 1 2 3 3 3 2 2 2 2 1. This means that the sensor considers that the scenario does not present any critical situation.

According to these three scenarios, three further equations are defined.

$$bh_{sensor_i} = \begin{cases} \text{Eq. 6.1: } \dfrac{prob}{\Delta_{size_i} \times freq_{alarms} \times changes_{states}} & \text{if } Case\ A; \\ \text{Eq. 6.2: } \dfrac{prob}{\Delta_{size_i} \times freq_{alarms}}; & \text{if } Case\ B; \\ \text{Eq. 6.3: } \dfrac{prob}{\Delta_{size_i}}; & \text{if } Case\ C; \end{cases} \tag{6}$$

Once the individual behaviour of a set of sensors of a cluster has been calculated and temporally stored within a list of behaviours (denoted in this paper as *List_BehaviourSensor*), the CH has to determine their conduct using a particular threshold of normality. To define the threshold, we consider the concept of '*acceptable state*' with p_{s_3} defined above, the value of which would indicate that any behaviour with a probability of less than p_{s_3} would be considered a critical situation. In other words, through this threshold the CH can detect the behaviour of its neighbourhood by analysing discrepancies of criticality taken by its sensors in their recent past. For the analysis, the CH has to run through the entire *List_BehaviourSensor* to observe whether any bh_{sensor_i} on the list is equal to or different from the rest; inferring the existence of a discrepancy of perceived situations within a cluster or a consensus in the observation tasks. Namely, let us suppose that we have a list *List_BehaviourSensor* $= \{\text{bh}_{sensor_1} < p_{s_3},\ \text{bh}_{sensor_2} \geq p_{s_3},\ \text{bh}_{sensor_3} \geq p_{s_3}\}$, this means that the ID_{sensor_1} has perceived a critical situation, contrary to what has been detected by its neighbours. In contrast, a *List_BehaviourSensor* $= \{\text{bh}_{sensor_1} \geq p_{s_3},\ \text{bh}_{sensor_2} \geq p_{s_3},\ \text{bh}_{sensor_3} \geq p_{s_3}\}$ states that all nodes in a cluster have perceived the same situation. Only in the case where the CH deduces discrepancies between criticality values, does it have to generate a new ISA100.11a alarm with a high priority through the ARMO class and send it to the gateway in charge of evaluating the level of accuracy for each ID_{sensor_i}. This evaluation serves as an attractive way to assess how the detection was really done by each sensor belonging to a particular cluster. To this end, the ISA100.11a alarm should include, at least, the ID of the CH (ID_{CH_i}) the type of event that has occurred (i.e., '*event_instabilityCH*'), and the list of behaviours that has been previously generated. All of these steps and equations are summarized in Algorithm 4.1.

Algorithm 4.1: ANALYSIS OF BEHAVIOUR PER NEIGHBOURHOOD(ID_{CH_i})

local $Anomalous\_BehaviourSensor, array_i, freq_{alarms}, changes_{states}$;
local $bh_{sensor_i}, List\_BehaviourSensor$;

$Anomalous\_BehaviourSensor \leftarrow$ **false**
$List\_BehaviourSensor \leftarrow$ INITIALIZATELIST();
for each $sensor_i \in CH$
do {
  $array_i \leftarrow$ EXPORTSTATES_FROMBUFFER($sensor_i$);
  $freq_{alarms} \leftarrow$ CRITICALALARM_FREQUENCY($array_i$);
  if $freq_{alarms} > 0$
  then {
    $changes_{states} \leftarrow$ STATESCHANGE($binarySeq_i$);
    if $changes_{states} > 0$
    then $bh_{sensor_i} \leftarrow$ CALCULATEEQUATION_6.1();
    else $bh_{sensor_i} \leftarrow$ CALCULATEEQUATION_6.2();
  }
  else $bh_{sensor_i} \leftarrow$ CALCULATEEQUATION_6.3();
  $List\_BehaviourSensor \leftarrow List\_BehaviourSensor \cup bh_{sensor_i}$
}
if DISCREPANCYOF_CRITICALITY($List\_BehaviourSensor$)
then {
  comment: Running through List_BehaviourSensor using the threshold p_{s_3}.
  GENERATEALARM_AM($List\_BehaviourSensor, ID_{CH_i}$, '$event\_instabilityCH$');
}

4.3 Gateway: Response, Assessment and Reporting

The main functionalities of the gateway focus on: *Location, warning and response* of the nearest human operator within the affected area, *assessment of the degree of accuracy* in the observation and protection tasks, and *reporting*. These three high-level services are described in detail in the following sections.

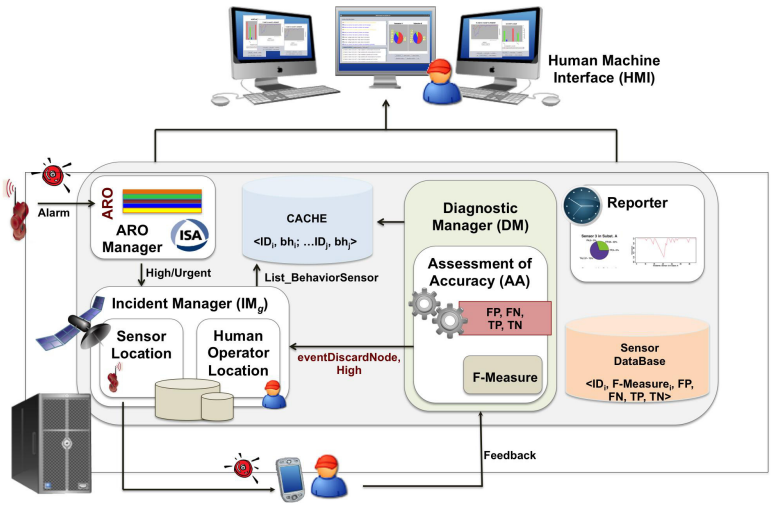

Figure 10: General Architecture of the Gateway

4.3.1 Location, Warning and Response

As mentioned, the gateway is in charge of managing any alarm received from the observation system and evaluating the real behaviour of its sensors. The architecture proposed in Figure 10 is based on three chief modules: *ARO*, *Incident Manager* (IM_g) and *Diagnostic Manager* (DM). The idea basically consists of receiving alerts through the ARO (Alert Receiving Object) class belonging to the DMAP class of ISA100.11a (see Section 4.2). This class is organized into five priority queues and each queue stores a type of alarm according to its level of criticality. In addition, the ARO module must perform three main tasks:

- Send any information received (either '*normal states*' or '*anomalous states*') from the observation system to the SCADA Centre.
- Send those '*anomalous states*' with criticality (4-5) to the IM_g so that it can locate the nearest human operator to the affected area. For the search for the nearest operator, the IM_g has to maintain a database with information related to the human operators (e.g., availability according to their contracts) and use global positioning technologies to identify their positions in the field. Once located, the system provides him/her with information about the situation, such as the location of the affected area, the type of event detected, the ID_{CH} or the ID_{sensor}. All the information has to be easily visualized through hand-held interfaces in a legible format to locally enable the mobility within the area to ensure a timely reaction.
- Notify the operator of the discrepancies of criticality detected by a CH, and help the WASA system compute the level of accuracy of each node belonging to such a cluster. In this case, the IM_g has to store, in a temporal memory cache, the list of the sensors' identifiers together with their anomalous behaviour to later assist in the assessment tasks managed by the *Assessment of Accuracy* (AA) module (introduced later).

4.3.2 Assessment of Accuracy

When the operator provides the system with the required feedback for evaluating the behaviour of each sensor of a neighbourhood after receiving the event '*event_instabilityCH*', he/she has to indicate the type of situation: '*Critical*' or '*non-critical*'. With this information, the system is then able to evaluate the level of accuracy attributing it to four different perspectives:

- A True Positive (TP): The sensor (sensors) observed a crisis scenario, and this coincides with the operator's feedback; i.e., a correct warning.
- A False Positive (FP): The sensor (sensors) observed a crisis scenario, but it is non-critical according to the operator's feedback; i.e., a false warning.
- A False Negative (FN): The sensor (sensors) did not observe a crisis scenario, when the system was threatened; i.e., a missed warning.

- A True Negative (TN): The sensor (sensors) did not observe any crisis scenario, and this coincides with the operator's feedback; i.e., a correct no warning.

Through the human operators' feedback, the DM can update the level of accuracy (see Section 3) of the observation system through the AA module. This module is in charge of maintaining four main counters for each sensor: (i) A $countTP_{sensor_i}$ to control the correct warnings, (ii) a $countFP_{sensor_i}$ to control the false warnings, (iii) a $countFN_{sensor_i}$ to control the missed warnings, and (iv) $countTN_{sensor_i}$ to control correct no warnings. Each of these counters has to be initialized to zero in the deployment and joining phases (i.e., during the commissioning phase), and are updated by one unit according to the conduct of the sensors in their detection and protection tasks. Note that the control of FNs is carried out when a CH observes an irregularity in its cluster. If all sensors except one (ID_{sensor_i}) warn of a critical situation and the operator's feedback verifies such a situation, then the rate of FNs associated to the node ID_{sensor_i} should be increased by one unit.

Once the counters involved have been updated, the system also has to compute the level of accuracy using the F-Measure technique [28]. This technique consists of computing the weighted harmonic mean (a mathematical concept related to the average) of *precision* and *recall*, the resulting value (probabilistic) of which, falls in the interval [0,1]. Zero indicates poor accuracy (i.e., a measurement with a significant error rate) and a value close to one represents good accuracy (i.e., a measurement with a small error rate). Observing Equation 7, the precision comprises the ratio of correct warnings ($countTP_{sensor_i}$) with respect to the rate of failure warnings (the sum of $countTP_{sensor_i}$ and $countFP_{sensor_i}$), whereas the recall specifies (see Equation 8) the ratio of correct warnings with respect to the rate of real failures (the sum of $countTP_{sensor_i}$ and $countFN_{sensor_i}$), both of which also fall in the interval [0,1].

$$precision = \frac{countTP_{sensor_i}}{countTP_{sensor_i} + countFP_{sensor_i}} \in [0,1] \tag{7}$$

$$recall = \frac{countTP_{sensor_i}}{countTP_{sensor_i} + countFN_{sensor_i}} \in [0,1] \tag{8}$$

According to [28], the F-Measure is computed as follows:

$$F - Measure = \frac{2 \times precision \times recall}{precision + recall} \in [0,1] \tag{9}$$

The F-Measure value is a probabilistic variable that will change according to the operator's feedback, who acts as a supervisor (see Section 4.3.2). To this end, the module AA needs to run through the whole *List_BehaviourSensor* stored in cache to compute the level of accuracy taken by a sensor in its recent past. To this end, the system proceeds as shown in Table 2, to increase by one unit one of the four counters mentioned above: $countTP_{sensor_i}$, $countFP_{sensor_i}$,

$countTN_{sensor_i}$ and $countFN_{sensor_i}$ $\forall$ $sensor_i$ of the cluster. For the computation of these counters, a threshold of criticality is required to delimit those critical and non-critical situations. This threshold will depend on the security policies and the owner organisation. We, for example, define as threshold, the value specified for p_{s_3} (defined in Section 4.2) given that it is the point of intersection between a critical situation and a non-critical situation; i.e., '*acceptable state*'.

	Normal State	Anomalous State
Operator's Feedback	$\text{bh}_{sensor_i} \geqslant p_{s_3}$	$\text{bh}_{sensor_i} < p_{s_3}$
Critical	$countFN_{sensor_i}$	$countTP_{sensor_i}$
Non-Critical	$countTN_{sensor_i}$	$countFP_{sensor_i}$

Table 2: Evaluating Behaviour both per Node and per Neighbourhood

As these counters have been updated, the system then has to calculate their new value F-Measure considering Equation 9. If the new value of F-Measure is close to zero or it is less than or equal to the minimum threshold (defined by the organisation), the AA module has to warn the system of the inefficacy or unreliability of the ID_{sensor_i}. The new warning should contain, at least, the identifier of the sensor, its location and the type of event *event_discardSensor*. The output should be managed by the IM_g to search for the nearest human operator to the affected node (see Section 4.3.1). For the sake of clarity, Algorithm 4.2 summarises this proposal.

Algorithm 4.2: ACCURACY PER NEIGHBOURHOOD($List\_BehaviourSensor, OpFeedback$)

```
local bh_sensor_i, fmeasure, recall, precision;
for each sensor_i ∈ CH
do {
    bh_sensor_i ← OBTAINBEHAVIOUR(sensor_i);
    if ISEQUAL(OpFeedback, "critical")
      then {
        if bh_sensor_i ⩾ p_s3
          then
            INCREASE(countFN_sensor_i);
          else INCREASE(countTP_sensor_i);
      }
      else {
        if bh_sensor_i < p_s3
          then
            INCREASE(countFP_sensor_i);
          else INCREASE(countTN_sensor_i);
      }
    precision ← CALCULATEEQUATION_6();
    recall ← CALCULATEEQUATION_7();
    fmeasure ← CALCULATEEQUATION_8();
    UPDATEFMEASURE(fmeasure, sensor_i);
}
```

4.3.3 Reporting

As the system manages information for each sensor, the *Reporter* module should periodically and/or on-demand generate and send frequent reports to the SCADA Centre. These reports help the system maintain a clearer vision of the functionality of the observation network by showing, through certain formats, (e.g.,

statistical graphics) the level of accuracy and functioning of its control elements. Likewise, it should be noted that although the rate of TN is not considered by the F-Measure technique in Equation 9 (see Section 4.3.2), it is considered in our mechanism in order to deliver a high degree of information in the reports to the SCADA Center.

5 Software Application for WASA: Examples and Discussions

In order to validate the first part of the mechanism defined in Section 4.2, we have implemented it in nesC and simulated it using the Avrora simulator under the de-facto standard operating system for sensor nodes TinyOS 2.x [31]. Avrora is able to interpret conventional sensor nodes such as the Mica2, which belong to category II defined in [32] with typically 4-8MHz, 4-10KB RAM, 48-128KB ROM with 2-8mA of energy. Table 3 illustrates the results of the simulation, which indicate that a cluster working as a Mica2, requires less than 7 MHz to execute the software, consuming around 0,67 Joule for CPU and 1.69 Joule for radio, approximately reaching a maximum of 2.8% for reading (r) and 3% for writing (w) in memory. Therefore, if traditional sensors are able to work as CHs, then ISA100.11a sensors belonging to category III with higher capabilities (13-180MHz, 256-512KB RAM, 4-32MB ROM and 40mA of energy) are also able to serve as CHs.

Overhead	**CH with 0 sensors**	**CH with 1 sensor**	**CH with 2 sensors**
CPU	6,55 MHz	6,46 MHz	6,63 MHz
Memory (r-w)	2,75% - 3,01 %	2,74% - 3,01%	2,73% - 3,00%
Energy - CPU	0,67 J	0,67 J	0,67 J
Energy - General Radio	1,69 J	1,69 J	1,69 J
Energy - Reading	2,75 J	2,74 J	2,73 J
Energy - Writing	3,01 J	3,01 J	3,00 J

Table 3: Computational and Communication Costs Invested in Algorithm 4.1

The second part has been implemented in Java, in which we have designed an emergency scenario based on two substations (substation A and substation B) under the control of two CHs (ID_{CH_1}, ID_{CH_2}) placed in each substation. Each CH has been configured with two sensors each (ID_{sensor_3}, ID_{sensor_4}, ID_{sensor_5} and ID_{sensor_6}) with all their counters $countTP_i$, $countFP_i$, $countFN_i$ and $countTN_i$ set to value zero. Each sensor node synchronously produces events with values that can range from valid readings, labelled with priority 0, to alarms, labelled with priority (1-5) (see Section 4.1). These events correspond to (either satisfactory and acceptable) normal and anomalous states, which are, respectively, linked to (p_{s_0} - p_{s_3}) and (p_{s_4} - p_{s_5}) (see Section 4.2). This also means that any alarm that exceeds the limits of normality (which should be

established by the organisation and its security policies) should be treated correctly. For the simulation, we have considered as the limit of normality the threshold (0-3) (non-critical alarms with normal states), and any event with criticality (4-5) (critical alarms that are considered as anomalous states) must be sent to one of the six virtual human operators that have been implemented for the simulation. These agents are chosen according to their availability (work time) defined through their virtual 'contracts' where their feedback is determined, probabilistically. Note that the scenario has been objectively stressed so as to intentionally generate the four different situations associated with TP, FP, FN and TN. This enables a more extensive study of the mechanism including each of the four cases possible.

Any event generated has to be analysed by CHs, as specified in Section 4.2. To understand the functionalities of these devices, a small example extracted from the simulation is further analysed below, where event sequences are received from sensors, such as ID_{sensor_3} and ID_{sensor_4} belonging to the ID_{CH_1} of substation A. Considering these notions together with graph $\mathcal{G}$ and Equation 6, Equation 7 and Equation 8 defined in Section 4.2, the study carried out by each CH with size of buffer $\Delta_{size} = 15$ is as follows:

- Sensor ID_{sensor_3} has received the sequence 1 0 0 1 0 0 1 0 1 3 0 0 3 0 1. As the $freq_{alarms}$ is equal to zero, the ID_{CH_1} computes the bh_{sensor_3} using Equation 6.3. As a result, the behaviour of the sensor, bh_{sensor_3}, is $0.3066 \in (p_{s_0} - p_{s_1})$. Given that $bh_{sensor_3} \geq p_{s_3}$, ID_{CH_1} infers that the sensor has not observed a critical situation in the last few minutes.

- Sensor ID_{sensor_4} has received the sequence 3 0 1 4 0 1 4 5 5 5 5 5 5 5 0. As the $freq_{alarms}$ is equal to nine and there are four changes in the state ($changes_{states}$), the cluster head calculates its behaviour through Equation 6.1. The result is bh_{sensor_4} of $0.0055 \in (p_{s_4} - p_{s_5})$. As $bh_{sensor_4} < p_{s_3}$, ID_{CH_1} concludes that the sensor detected a critical situation in the past.

As one of the sensors of the cluster has detected an emergency situation and the other hasn't, ID_{CH_1} has to send a new critical alert with a high priority to the gateway in order to observe the behaviour of the entire neighbourhood. The new alert should contain, at least, ID_{CH_1} and the list *List_BehaviourSensor* with $\{bh_{sensor_3}, bh_{sensor_4}\}$. Likewise, a human operator has to be warned of this situation to obtain from him/her certain feedback which would correspond to the true nature of the situation. In this way, it is also possible to determine whether ID_{sensor_3} and ID_{sensor_4} were right in their observation tasks; i.e., Has there really been an emergency situation in the last few minutes or not? If so, this means that the ID_{sensor_4} was right, but: Why didn't ID_{sensor_3} detect this situation? Moreover, exactly the opposite can happen.

In these circumstances, the system has to penalize/compensate the operation of both nodes in some way (the node in question and its neighbour). To this end and depending on the human operator's feedback, two types of situation can appear.

- Case A: The operator's feedback denotes a *critical* situation, so the AA module integrated inside the DM has to compute two specific cases:
  - A FN: ID_{sensor_3} did not detect the situation properly (see Table 2 with $\text{bh}_{sensor_i} \geqslant p_{s_3}$). The AA module therefore increases its counter $countFN_{sensor_3}$.
  - A TP: ID_{sensor_4} did detect the emergency situation correctly. Hence, the AA increases its $countTP_{sensor_4}$ (see Table 2 with $\text{bh}_{sensor_i} < p_{s_3}$).
- Case B: The operator's feedback indicates a *non-critical* situation, therefore:
  - A TN: ID_{sensor_3} did not detect an emergency scenario properly (see Table 2 with $\text{bh}_{sensor_i} \geqslant p_{s_3}$). The AA module therefore increases its counter $countTN_{sensor_3}$.
  - A FP: ID_{sensor_4} the sensor made a mistake in the observation and the AA increases its $countFP_{sensor_4}$ (see Table 2 with $\text{bh}_{sensor_i} < p_{s_3}$).

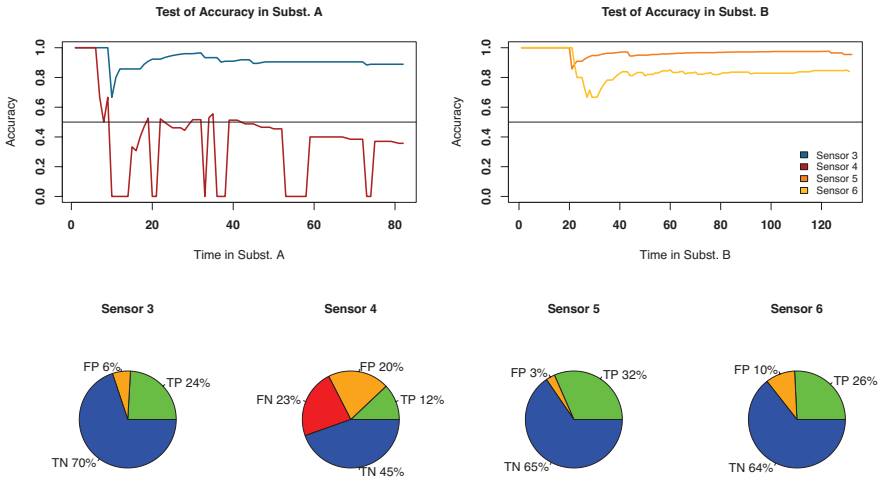

Figure 11: Addressing Behaviour to Measure Reliability

Once these two situations and the updating of the counters have been taken into account, the AA module then calculates the new F-Measure value using Equation 9 defined in Section 4.3.2. As a result, Figure 11 depicts, on the one hand, the relevance of maintaining the rates of TP, TN, FP and FN, and on the other hand, the importance of the F-Measure for each sensor involved (also including the sensors ID_{sensor_5} and ID_{sensor_6}). In order to understand

the significance of this illustration, a brief analysis for each substation is given below.

- Substation A: The cluster head ID_{CH_1} together with ID_{sensor_3} and ID_{sensor_4} have produced 82 events, of which 49 indicated a real critical situation (the sum of rates of both TP and FN). Moreover, 52 warnings (between TP and FP) were managed: 25 for ID_{sensor_3} and 27 for ID_{sensor_4}.

  Looking at Figure 11, we observe that the worst scenario is caused by ID_{sensor_4}, receiving 19 FN and 17 FP according to the operators' virtual feedback. This fact means that its F-Measure value significantly decreases in Figure 11 until it falls to the minimum value permitted (we have considered the value zero). Moreover, the node ID_{sensor_4} presents an irregular behaviour in its control tasks the whole time. This not only may obstruct the protection tasks of its neighbourhood but it may also put the welfare of the CI at risk. To the contrary, ID_{sensor_3} decreases its F-Measure ($\simeq 0.9$) but maintains its F- Measure value within the acceptable permitted threshold (between 1 and 0.5). This also means that this node has behaved appropriately for the protection during the simulation.

- Substation B: The cluster head ID_{CH_2} together with ID_{sensor_5} and ID_{sensor_6} generated 131 events, of which 76 indicated a real emergency situation (TP). In addition, 93 warnings (between TP and FP) were managed: 46 for ID_{sensor_5} and 47 for the ID_{sensor_6}.

  The worst scenario is to be found in ID_{sensor_6}, receiving 13 FP whereas ID_{sensor_5} demonstrates a correct conduct at all times (see Figure 11). Note that although the F-Measure of ID_{sensor_6} is significantly reduced ($\simeq 0.8$), its value falls between the values of acceptable threshold of the F-Measure [1,0.5]. This means that although its behaviour is suitable for the control, the frequent semi-abrupt changes taken by the F-Measure should put those human operators on alert so as to anticipate a response when this is needed. Nonetheless, both sensors have shown suitable behaviour during the simulation.

Figure 12: Principal Interface of the WASA Solution

As mentioned, the WASA mechanism has been implemented to offer an interactive solution applicable for critical contexts. The software application is based on three main interfaces, which are depicted in Figure 12 and Figure 13. Figure 12 corresponds to the principal interface of the operator in which he/she can visualize:

- The alarms received from the observation system, which are categorized by the five levels of priority given by the ISA100.11a standard [11]. Each

alarm is associated with: The sensor identifier, the location of the sensor within the cluster, the criticality of the alarm, a brief explication of the problem (e.g., type of event) and the time when such an alarm was received. Despite the fact that this WASA solution has mainly been based on the ISA.100.11a standard, both the mechanism and the software application can be configured to define other ways of classifying, managing and representing other formats of alarm. In a nutshell, both the proposed mechanism and its application can be customized for application in different types of application contexts (e.g., transportation systems, water treatment systems, Smart Grids, etc.).

- The list of available operators according to their virtual 'contracts' and the operator attending (or has attended) to the alarm.
- The rate of TP, FP, TN and FN carried out in each substation at all times.

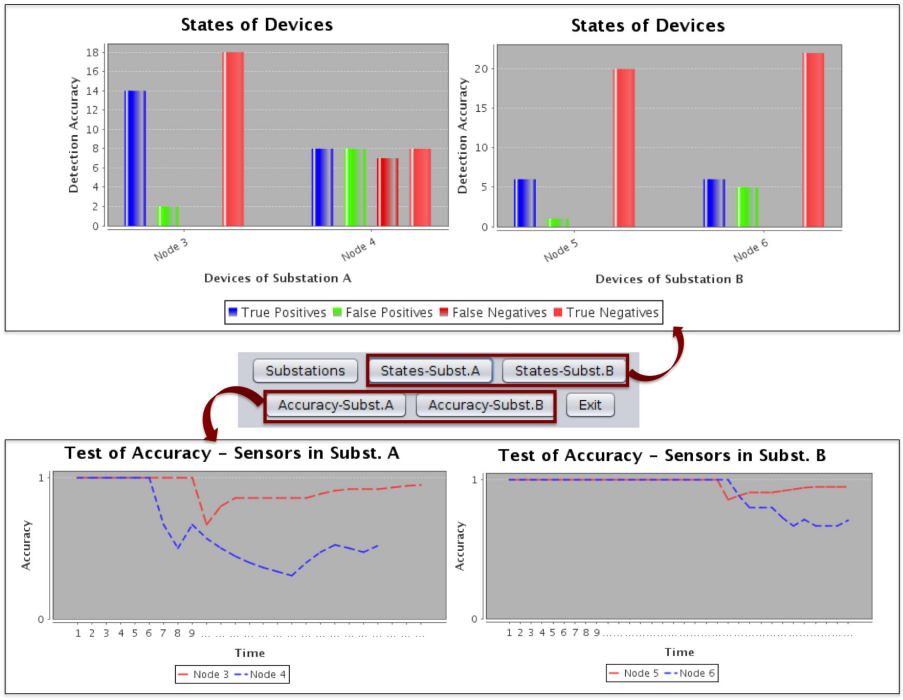

Figure 13: Accuracy in the Observation Tasks

Figure 13 shows, on the one hand, the rate of TP, FP, TN and FN carried out by each sensor in each substation (at the top of Figure 13); and on the

other hand, the test of accuracy of such sensors with respect to the time line (at the bottom of Figure 13). As a result, human operators and the SCADA Centre are not only made aware of the real nature of its observation system through these interfaces (running at the time), but they can also understand the degree of severity of a situation by using easy and legible graphical interfaces. For example, they can control the values taken by the rates of FN and FP at all times. In fact, a significant increase of their values could trigger a serious operational problem not only for the control infrastructure itself, but also for those controlled infrastructures [8].

6 Conclusions and Future Work

Given that situational awareness is a priority topic for the protection of energy control systems, we have presented, in this paper, a diagnosis mechanism based on industrial wireless sensor networks, on the ISA100.11a standard and on the F-Measure technique. Combining these three components, together with other communication technologies, such as mobile ad-hoc networks or the Internet, the system is able to: (i) Know natural conditions of both the critical infrastructure observed and the monitoring system itself, (ii) respond to (critical or anomalous) situations, and (iii) assess the degree of accuracy reached in the control tasks. In order to validate and verify the feasibility of the mechanism for critical contexts, a software application has been implemented together with a critical scenario consisting of two substations based on a hierarchical configuration. Each cluster head receives a set of (critical and non-critical) events from the sensors so as to evaluate their behaviour according to their levels of accuracy in detection and protection tasks.

As for future work, it would be useful to extend the mechanism to include forecasting models that help the system anticipate irregular behaviour and respond in advance. In addition, and taking advantage of the accuracy concept, the system could even assess these forecasting models and the reliability of the assessment modules to deliver a much more complete tool able to evaluate, by itself, the entire mechanism. Moreover, it would be interesting to explore the benefits of other accuracy techniques (e.g., reputation) and other existing technologies for wide-area situational awareness, such as cloud computing or the Internet of Things, where aspects relative to privacy of critical information and location of devices have to be carefully considered.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation through the research project ARES (CSD2007-00004), by the Andalusian government through the PISCIS project (P10-TIC-06334), and by the EU FP7 through the FACIES project (HOME/2011/CIPS/AG/4000002115). Additionally, in the particular case of the first author, the research leading to

these results has received funding from the Marie Curie COFUND programme "U-Mobility" co-financed by University of Malaga and the European Community Seventh Framework Programme under Grant Agreement No. 246550.

References

[1] M. Endsley, and E. Connors (2008), Situation Awareness: State of the Art, IEEE Power and Energy Society General Meeting Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1-4.

[2] NIST (2012), NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0, NIST Special Publication 1108R2.

[3] B. Miller and B. Young (2012), A Survey of SCADA and Critical Infrastructure Incidents, Conference on Information Technology Education, pp. 1-6.

[4] J. Peerenboom and R. Fisher (2007), Analysing Cross-Sector Interdependencies, IEEE Computer Society, HICSS, IEEE Computer Society, pp. 112–119.

[5] ICS-CERT (2011), ICS-CERT Incident Response Summary Report, pp. 1-17, 2001-2009, `http://www.us-cert.gov`, Retrieved on January 2013.

[6] ICS-CERT (2012), ICS-Monitor Malware Infections in the Control Environment, pp. 1-15, October/November/December 2012, `http://www.us-cert.gov`, Retrieved on August 2013.

[7] ICS-CERT (2013), ICS-Monitor Brute Force Attacks on Internet-Facing Control Systems, pp. 1-15, June 2013, `http://www.us-cert.gov`, Retrieved on August 2013.

[8] C. Alcaraz and J. Lopez (2012), Analysis of Requirements for Critical Control Systems, International Journal of Critical Infrastructure Protection, Elsevier, 2, 3-4, pp. 137-145.

[9] C. Alcaraz and J. Lopez (2010), A Security Analysis for Wireless Sensor Mesh Networks in Highly Critical Systems, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40, 4, pp. 419-428.

[10] B. Reaves, and T. Morris, An Open Virtual Testbed for Industrial Control System Security Research, International Journal of Information Security, 11, 4, pp. 215-229, 2012.

[11] ISA (2009-2013), ISA100.11.a-2009: Wireless Systems for Industrial Automation - Process Control and Related Applications, `http://www.isa.org/`, Retrieved on August 2013.

[12] ENISA (2012), National Cyber Security Strategies, Setting the Course for National Efforts to Strengthen Security in Cyberspace, pp. 1-15.

[13] J. Butts H. Kleinhans, R. Chandia, M. Papa and S. Shenoi (2009), Providing Situational Awareness for Pipeline Control Operations, IFIP Advances in Information and Communication Technology, Critical Infrastructure Protection III, 311, pp. 97-111.

[14] A. Mavridou and M. Papa (2012), A Situational Awareness Architecture for the Smart Grid, Global Security, Safety and Sustainability & e-Democracy, Social Informatics and Telecommunications Engineering, Springer Berlin Heidelberg, LNCS 99, pp. 229-236.

[15] W. Xing, G. Castelli, C. But-Chung, X. Jinghai, and D. Sun (2012), Comprehensive Situation Awareness in a very Large Power Grid Control Center, Transmission and Distribution Conference and Exposition, IEEE PES, pp. 1-6.

[16] C. Alcaraz, and J. Lopez (2013), Wide-Area Situational Awareness for Critical Infrastructure Protection, IEEE Computer, vol. 46, no. 4, pp. 30-37.

[17] R. Parasuraman, and V. Riley (1997), Humans and Automation: Use, Misuse, Disuse, Abuse Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol. 39, No. 2, pp. 230-253.

[18] Q. Hairong, L. Yilu Liu, F. Li, L. Jiajia, H. Li, K. Tomsovic, L. Tolbert, and C. Qing (2011), Increasing the Resolution of Wide-Area Situational Awareness of the Power Grid through Event Unmixing, Hawaii International Conference on System Sciences (HICSS), pp. 1-8.

[19] F. Fujikawa (2012), Evaluation of Applicability to WAMPAC (Wide Area Monitoring Protection and Control) of IEEE 1588, IEEE Third International Conference on Smart Grid Communications (SmartGridComm), pp. 593 - 598.

[20] S. Matsumoto, Y. Serizawa, F. Fujikawa, and T. Shioyama (2012), Wide-Area Situational Awareness (WASA) system based upon international standards, International Conference on Developments in Power Systems Protection (DPSP), pp. 1-6.

[21] V. Gungor, B. Lu and G. Hancke (2010), Opportunities and Challenges of Wireless Sensor Networks in Smart Grid, IEEE Transactions on Industrial Electronics, 57, 10, pp. 3557-3564.

[22] G. Gupta and M. Younis (2003), Fault-tolerant Clustering of Wireless Sensor Networks, In IEEE Wireless Communications and Networking, 3, pp 15791584.

[23] ZigBee Alliance (2011), ZigBee-08006r03: ZigBee-2007 Layer PICS and Stack Profiles (ZigBee-PRO), Rev. 3, http://www.zigbee.org/, Retrieved on August 2013.

[24] HART Communication Foundation (2009), WirelessHART. http://wirelesshart.hartcomm.org/, Retrieved on August 2013.

[25] C. Alcaraz, J. Lopez, R. Roman, and H. Chen (2012), Selecting Key Management Schemes for WSN Applications, In Computers & Security, Elsevier, 38, 8, pp. 956966.

[26] F. Baker and D. Meyer (2011), Internet Protocols for the Smart Grid, Internet Engineering Task Force (IETF), RFC-6272.

[27] The White House, Office of the Press Secretary (2009), President Obama Announces $3.4 Billion Investment to Spur Transition to Smart Energy Grid, News.

[28] F. Salfner (2008), Event-based Failure Prediction An Extended Hidden Markov Model Approach, PhD Thesis, Humboldt-Universittzu Berlin, pp. 1-345.

[29] JCGM 200:2008 (2008), International Vocabulary of Metrology Basic and General Concepts and Associated Terms (VIM), pp. 1-89.

[30] V. Chandola, A. Banerjee, and V. Kumar (2009), Anomaly Detection: A Survey, ACM Computer Survey, vol. 41, no 3, article 15, pp. 1-58.

[31] TinyOS Working Group (2013), http://www.tinyos.net/, Retrieved on August 2013.

[32] J. Lopez, R. Roman, and C. Alcaraz (2009), Analysis of Security Threats, Requirements, Technologies and Standards in Wireless Sensor Networks, In Foundations of Security Analysis and Design 2009, LNCS 5705, pp. 289-338.